Local-Density-Based Optimal Granulation and Manifold Information Granule Description

Ji Xu, Guoyin Wang, *Senior Member, IEEE*, Tianrui Li, *Senior Member, IEEE*, and Witold Pedrycz, *Fellow, IEEE*

***Abstract*—Constructing information granules (IGs) has been of significant interest to the discipline of granular computing. The principle of justifiable granularity has been proposed to guide the design of IGs, opening an avenue of pursuits of building IGs carried out on a basis of well-defined and intuitively appealing principles. However, how to improve the efficiency and accuracy of the resulting constructs is an open issue. In this paper, we present a local-density-based optimal granulation model (LoDOG), exhibiting evident advantages: 1) it can detect arbitrarily-shaped IGs and 2) it finds the optimal granulation solutions with $O(N)$ complexity, once the leading tree structure has been constructed. We describe IGs of arbitrary shapes using a small collection of landmark points positioned on the skeleton of the underlying manifold, which contribute to approximate reconstruction capabilities of the original dataset. A dissimilarity metric is developed to evaluate the quality of the obtained reconstruction. The interpretability of LoDOG IGs is discussed. Theoretical analysis and empirical evaluations are covered to demonstrate the effectiveness of LoDOG and the manifold description.**

***Index Terms*—Granular computing (GrC), local density, manifold descriptor, optimal granulation, principle of justifiable granularity.**

I. Introduction

INFORMATION granulation, defined as a process of building information granules (IGs) from the original data, has been recognized as a core step in granular computing (GrC). Building IGs is helpful when dealing with the 5Vs of big data, namely volume, velocity, variety, value, and veracity. By putting the similar or neighboring data together, granulation can considerably reduce the size of data and the volume problem is partly solved. In DP-Stream [1], we have used the fat node leading tree (LT) to timely deliver the clustering result of newly arrived items in a high speed data stream, which serves as a good example that IGs help to conquer the difficulty of velocity. To construct IGs from heterogeneous data sources, one has to define a uniform representation for the information collected, thus the variety is removed. Chen and Zhang [2] have pointed out that different granular levels of IGs exhibit distinct knowledge, where some features are neglected and others of interest are highlighted. In this way, the values of big data are revealed to meet different cognitive requirements. At last, the process of IG construction usually includes detecting and removing outliers, and coping with incomplete data. Therefore, the veracity of big data is guaranteed.

There have been a number of formalisms and methods used to construct granules, e.g., rough sets, fuzzy sets, neighborhood systems, clustering, and others. These methods answer the question of how to granulate the data, but the problem of what requirements the granules should meet were not fully addressed until recent years. Pedrycz and Homenda [3] explored this issue by proposing the principle of justifiable granularity. The basic idea of justifiable granulation is that a good IG should contain as many data points as possible (coverage) while keeping the contained data lying in a compact closure (specificity). These two key elements are named as *experimental evidence* (or coverage) and *semantics* (or specificity) [3], [4].

A number of models have been developed [such as ellipsoidal IGs (ElliGra) [5] and fuzzy sets IGs [4]] to construct granules that follows the principle of justifiable granularity.

However, the existing granulation methods following the justifiable granularity exhibit several limitations.

1) They are unable to detect some concave shapes of granules.
2) The processes of granulation are computationally expensive because they usually involve an iteration procedure to find the optimal number of granules, as well as to determine the shape of each IG when considering the reconstruction of the IG's original data (e.g., particle swarm optimization is used in [6] and differential evolution is used in [5]). But in some applications such as on-line community detection in social networks, granulation models of high complexity may miss a critical change of the data, thus might not be affordable. Therefore, highly efficient granulation models are desired in such situations.

Manuscript received May 30, 2017; revised August 27, 2017; accepted September 4, 2017. Date of publication September 19, 2017; date of current version September 14, 2018. This work was supported in part by the National Key Research and Development Program of China under Grant 2016QY01W0200 and Grant 2016YFB1000905, and in part by the National Natural Science Foundation of China under Grant 61572091. This paper was recommended by Associate Editor Y. Yang. *(Corresponding author: Guoyin Wang.)*

J. Xu and T. Li are with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China (e-mail: alanxuch@hotmail.com; trli@swjtu.edu.cn).

G. Wang is with the Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: wanggy@ieee.org).

W. Pedrycz is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6R 2V4, Canada (e-mail: wpedrycz@ualberta.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

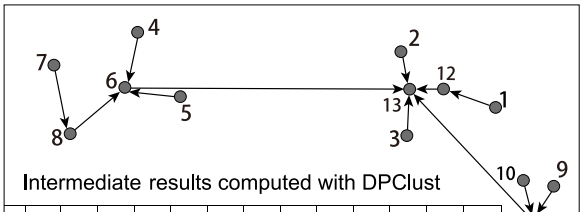
Digital Object Identifier 10.1109/TCYB.2017.2750481

3) It is frequently encountered in real-world applications, if not always, that is, when the granules lie on or approximately near a manifold that has a lower intrinsic dimensionality (e.g., an S-shaped curve line or a curve surface like a bowl), the existing granular descriptor, such as ellipsoids and boxes, would be of lower coverage and specificity and thus being unable to reconstruct a dataset similar to the original one. In other words, the descriptors would have a relatively large volume (viz., low specificity) yet contain only a few data points (viz., low coverage) on the manifold with a lower intrinsic dimensionality.

The contribution of this paper is twofold. We first propose a local-density-based optimal granulation (LoDOG) approach to address the first two issues. Using the local density of each data point, we first build a tree structure called an LT [7]. In the LT, each noncenter datum is led by its father to join the same cluster (or micro-cluster). The procedure of assigning the noncenter data points to their center is simplified as disconnecting the root (the selected center) of a subtree from the father of the root. The potential of each data point x_i to be assigned as a center (denoted as γ_i) is calculated as the product of local density (denoted as ρ_i) and the nearest distance to a neighbor with higher density (denoted as δ_i). That is, we use the parameter $\gamma_i = \rho_i * \delta_i$ to indicate the possibility of x_i being chosen as a center. With the definition of LT, one can easily find that the relation between any pair of nodes in the LT is a relation of partial order [1]. By leveraging the properties of the LT, finding the optimal granules by LoDOG is linear with respect to the number of data. This makes LoDOG highly efficient and applicable in the context of big data.

Then, for the third problem, our method is based on locality preserving manifold dimensionality reduction [8]–[10] and landmark points sampling. Since it is difficult to capture the geometry of the data points without knowing its underlying distribution, we turn to sample the landmark points on the embedding of lower dimensionality, then by leveraging the properties of locality preservation and the order (or the indices) unchanging, the sampled image points are mapped back to their corresponding inverse images. In this way, the skeleton of the original dataset is captured by the landmarks points on the manifold of high dimensionality. We call the set of these landmark points (or the linear patches formed by the adjacent landmark points) a *manifold descriptor* of the data subset as an IG. Based on this manifold descriptor, we present an algorithm to reconstruct the artificial data points to imitate the original ones. Also proposed is a metric named as *sketch error*, defined as the summation of all mean block-wise Earth mover's distance (EMD) [11] assigning the cost on each edge as 1, to evaluate the dissimilarity between the reconstruction and the original dataset. The coverage and specificity of the manifold descriptors are discussed subsequently.

This paper is organized as follows. Section II briefly reviews existing research closely related to our methods. Section III presents the method of LoDOG for efficient and accurate information granulation. In Section IV, we describe the manifold descriptor, generation of artificial data (reconstruction for short), and the metric sketch error to evaluate the quality of reconstructions. The interpretability of LoDOG IGs is quantified in Section V. Section VI discusses the time complexity and the relationship to other researches. Section VII illustrates and validates our methods with artificial datasets and real-world datasets. A scientific collaboration network is included to demonstrate the potential of LoDOG in the problem of community detection. At last, we reach the conclusion in Section VIII.



I	1	2	3	4	5	6	7	8	9	10	11	12	13
ρ	0.32	0.72	0.93	0.52	0.27	0.95	0.22	0.48	1.52	1.48	1.63	1.62	1.70
δ	1.17	0.78	0.83	0.93	1.13	5.93	1.38	0.97	0.38	0.45	3.97	0.44	5.93
γ	0.38	0.56	0.77	0.48	0.30	**5.63**	0.30	0.47	0.59	0.67	**6.45**	0.71	**10.07**
P	12	13	13	6	6	13	8	6	11	11	13	13	-1

Fig. 1. LT example of 13 nodes. Note that the γ values of $\boldsymbol{x}_6$, $\boldsymbol{x}_{11}$, and $\boldsymbol{x}_{13}$ are large compared with those reported for others.

II. Related Studies

A. Local Density and Leading Tree Structure

Rodriguez and Laio [12] proposed a novel clustering method based on density peaks (referred to as DPClust). The array indicating the nearest neighbor with higher local density, as one intermediate result of DPClust, forms an LT [7]. In an LT, each node (except the root of the whole tree and those that are selected as centers) is led by its parent to join the same cluster.

In this paper, we use the structure LT and the principle of justifiable granularity to efficiently construct optimal granules, with the linear search on the sorted $\boldsymbol{\gamma}$ vector. To construct the LT for a given dataset, three steps are performed.

1) Compute the distance matrix for all the data points $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$. The output is $\text{Dist} = \{d_{ij}\}$, d_{ij} is any dissimilarity metric computed for $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. In this paper, l_2-norm (Euclidean distance) is used if $\boldsymbol{x}_i \in \mathcal{R}^D$. For social network mining task, we use the distance based on *overlap* [13] in (25).
2) Compute the local density ρ_i for each $\boldsymbol{x}_i$. The method to compute ρ_i is in the form of cut off kernel or Gaussian kernel. Here we use the latter that is defined as

$$\rho_i = \sum_{j \in I \setminus \{i\}} e^{-\left(\frac{d_{ij}}{d_c}\right)^2}. \tag{1}$$

3) Construct the LT by linking every data point (except the one with the highest ρ_i) to its parent. The index of each parent node is specified as a vector $\boldsymbol{p} = (p_1, p_2, \ldots, p_N)$, in which

$$p_i = \begin{cases} -1, & \text{if } \rho_i = \max(\boldsymbol{\rho}) \\ \arg\min\limits_j \left\{d_{ij} | \rho_j > \rho_i\right\}, & \text{otherwise.} \end{cases} \tag{2}$$

Fig. 1 illustrates the construction of an LT of 13 nodes.

Once the LT has been constructed, another parameter γ_i needs to be associated with every node $\boldsymbol{x}_i$ in the LT to indicate the possibility that $\boldsymbol{x}_i$ is selected as a center. That is, for $i = 1, 2, \ldots, N$

$$\gamma_i = \rho_i * \delta_i \tag{3}$$

in which, δ_i is the distance between x_i and its parent if x_i is not the root of the whole LT, and δ_i equals to the maximum among other δ values if x_i is the root. Formally

$$\delta_i = \begin{cases} \max\limits_{j \in I \backslash i} \{\delta_j\}, & \text{if } \rho_i = \max(\boldsymbol{\rho}) \\ \text{Dist}(i, p_i), & \text{otherwise.} \end{cases} \tag{4}$$

B. Principle of Justifiable Granulation

The principle of justifiable granulation [3] is a milestone in the progress and evolution of GrC. It offers macro guidance on quantitatively evaluation of the granulation result of an original dataset.

One can find out that coverage and specificity are in conflict by nature because a granule containing more objects usually has a closure of "bigger volume." Higher coverage implies lower specificity and vice versa. Therefore, as a trade-off between experimental evidence and semantic meaning, the performance index of an IG and the overall performance of a granulation solution are defined as follows.

Definition 1 [5]: Suppose a dataset X has been granulated as a set of granules $\boldsymbol{\Omega} = \{\Omega_i | i = 1, 2, \ldots, c\}$, then the performance index for an IG Ω_i is

$$V(\Omega_i) = \text{Coverage}(\Omega_i) * \text{Specificity}(\Omega_i), i = 1, 2, \ldots, c \tag{5}$$

and the overall performance index of $\boldsymbol{\Omega}$ is

$$V(\boldsymbol{\Omega}) = \sum_i V(\Omega_i). \tag{6}$$

Here Coverage(Ω_i) is the cardinality of the data point set covered by granule Ω_i and Specificity(Ω_i) can be defined as any strictly monotonically decreasing function of the volume of Ω_i.

C. Local Linear Embedding

The construction of our manifold descriptor can be based on any method for locality preserving manifold dimensionality reduction. For easier understanding and efficient implementation, we choose local linear embedding (LLE) proposed by Saul and Roweis [8] and Roweis and Saul [9]. The idea of LLE is simple and ingenious. It works under the following assumption: if the data points in D-dimensional space are sampled intensively enough then the neighboring points would approximately lie on d-dimensional linear patch with $d < D$. The core idea of LLE is that every data point can be expressed as the weighted summation of its k-nearest neighbors. LLE finds the coordinates of the data points in a lower dimensional space through three steps completed in the following sequence.

Step 1: Select k nearest neighbors for each data point under any distance metric.

Step 2: Determine the optimal W_{ij} for each data point and its k-nearest neighbors, such that the reconstruction error $E(W)$ becomes minimized

$$E(W) = \sum_i \left| \boldsymbol{x}_i - \sum_j W_{ij} \boldsymbol{x}_j \right|^2 \quad \text{s.t.} \sum_j W_{ij} = 1 \text{ for all } i. \tag{7}$$

Step 3: Compute the optimal coordinates for each original data point $\boldsymbol{x}_i$ by using only the W_{ij} derived in step 2 to minimize the embedding cost function

$$\Phi(Y) = \sum_i \left| \boldsymbol{y}_i - \sum_j W_{ij} \boldsymbol{y}_j \right|^2. \tag{8}$$

After formulating the problem in the simple, elegant and symmetric form, the optimal solutions for W and Y can be found by the standard linear algebra methods. The reader can refer to [8, Sec. 4] for details on how to solve (7) and (8).

To use LLE for the manifold descriptor construction, we should keep three points in mind. First, LLE transformation preserves the neighborhood and co-location of the data points, because the $\{W_{ij}\}$ that are optimized in step 2 remain constant in step 3. Second, LLE does not change the order of the points, that is, $\boldsymbol{y}_i$ is definitely the image of $\boldsymbol{x}_i$ for all $1 \le i \le N$. The last one is that there is no explicit transforming matrix can be found between X and Y. To address this issue, Saul and Roweis [8] discussed two possible approaches to construct mappings between X_i and Y_i. But the approaches are not suitable for our fast descriptor formation because they require considerable computations.

III. Local Density-Based Optimal Granulation

Rodriguez and Laio [12] stated that the product of ρ_i and δ_i is an effective index showing how likely x_i is chosen to be a center. With this feature, LT has been used in efficient hierarchical clustering [7], and the partial order relationship between any pair of nodes in LT has been used for data stream clustering [1]. Here, we show it is feasible to obtain the optimal granulation with linear search on the possible centers by leveraging the semantics of parameter γ and the partial order present in the LT.

For LoDOG, because it is inconvenient and unnecessary to calculate the geometrical volume of each granule, we modify the formula in [5] that evaluates the performance of a granulation solution, yet the spirit of the justifiable granularity keeps unchanged. We formulate the objective function to be minimized as

$$J(N_g | \alpha) = \alpha * H(N_g) + (1 - \alpha) \sum_{i=1}^{N_g} \text{DistCost}(\Omega_i) \tag{9}$$

$$\text{DistCost}(\Omega_i) = \sum_{j=1}^{|\Omega_i|-1} \left\{ \delta_j | \boldsymbol{x}_j \in \Omega_i \backslash R(\Omega_i) \right\} \tag{10}$$

where N_g is the number of granules (which can be regarded as micro-clusters in a real cluster); α is the parameter striking a sound balance between the coverage and specificity; Ω_i is the set of points included in ith granule as defined in Section II-B; $|\bullet|$ is the cardinality operator; $H(\bullet)$ is a strictly monotonically increasing function used to adjust the magnitude of N_g to well match that of $\sum_{i=1}^{N_g} \text{DistCost}(\Omega_i)$, and it can be automatically selected from a collection of commonly used functions such as logarithm, linear, power functions, and exponential ones; and $R(\Omega_i)$ is the root of the granule Ω_i as a subtree.

The objective function $J(N_g|\alpha)$ is to be minimized to get the optimal granulation. Intuitively, we want the population of the points covered in a granule to be great (coverage item) and the data within an IG to be densely distributed (specificity item). So we alternatively want the number of granules to be small, and the summation of the distance between every element and its parent to be small. The $\delta_{R(\Omega_i)}$ of each subtree, as an IG, are excluded from the $\text{DistCost}(\Omega_i)$ computation because it is the external distance to another granule. Note that the two objectives are in conflict by nature, so we use a simple weighted summation to strike a sound compromise between them.

In the LT structure, the partial order and the semantic of γ parameter imply that if a data point $\boldsymbol{x}_i$ with center possibility indicator γ_i is not selected as a center, then all other $\boldsymbol{x}_j$ with $\gamma_j < \gamma_i$ would have no chance to make a center. Therefore, LoDOG can find the optimal solution of N_g to minimize $J(N_g|\alpha)$ by only one pass of a linear search. The search is performed on the sorted $\boldsymbol{\gamma}$. In addition, $\sum_{i=1}^{N_g} \text{DistCost}(\Omega_i)$ can be computed in an incremental fashion, because we have

$$\sum_{i=1}^{N_g-1} \text{DistCost}(\Omega_i) = \delta_M + \sum_{i=1}^{N_g} \text{DistCost}(\Omega_i) \tag{11}$$

where δ_M denotes the distance between the center $\boldsymbol{x}_M$ of the merged cluster and the parent of $\boldsymbol{x}_M$ in the LT. M is index of the object that has Ngth greatest γ value. That is, if $\tilde{\boldsymbol{I}}$ denotes the index array manipulated by sorting $\boldsymbol{\gamma}$ in a descending order, then we have

$$M = \tilde{\boldsymbol{I}}_{N_g}. \tag{12}$$

So LoDOG is of high efficiency when compared to other competing methods. The details of this procedure are described in Algorithm 1, and the readers can refer to Section VII-A1 for an illustrative example.

Algorithm 1: LoDOG

Input: Dataset X, parameter α, d_c
Output: The optimal granules $\boldsymbol{\Omega}$ in the form of N_g^{opt} microclusters

Compute $\boldsymbol{\rho}$, $\boldsymbol{\delta}$ and $\boldsymbol{\gamma}$ and $\boldsymbol{P}$;
Construct the LT;
//construct and evaluate the finest granules;
$DistCost$=0;
$\boldsymbol{SubT}$= split the LT into $MaxN_g$ subtrees;
$[\boldsymbol{\gamma}_s, \boldsymbol{I}_\gamma]$=sort($\boldsymbol{\gamma}$, "descending");
for i=*1 to* $MaxN_g$ **do**
  for *each non-root* $\boldsymbol{x}_j$ *in* $SubT_i$ **do**
    $DistCost$= $DistCost$+δ_j;
  end
end
$J(MaxN_g|\alpha)$=$\alpha * H(MaxN_g) + (1-\alpha) * DistCost$;
//evaluate the coarser granules incrementally;
for $i = MaxN_g - 1$ *to* 2 **do**
  M= $I_\gamma[i]$;
  $DistCost$= $DistCost$+δ_M;
  $J(i|\alpha)$=$\alpha * H(i) + (1-\alpha) * DistCost$;
end
$N_g^{opt} = \arg\min_i(J(i|\alpha))$;
$\boldsymbol{\Omega}$=split the LT into N_g^{opt} subtrees;

IV. Manifold Description of the IGs

In many applications, the data points sufficiently sampled from high dimensionality actually lie on an embedding of much lower dimensionality. The artificial dataset of S-shaped surface and the faces dataset in [9], and the handwriting digits dataset in [14], all can serve as good examples in this regard.

In this situation, traditional granule descriptors using ellipsoids [5] or hyper-boxes [15] to enclose the data points may be confronted with two essential problems. First, they fail to capture the intrinsic topology, and lead to a considerable drop in both coverage and specificity (Fig. 2 offers some explanation). Second, if one needs to reconstruct data points from the IG descriptor, then the data points randomly sampled in the closure (defined by the descriptor) will completely lose the geometric feature of the original IG. Therefore, we develop an embedding-landmarks-based approach to describe the optimal granules that are obtained with LoDOG in the previous step.

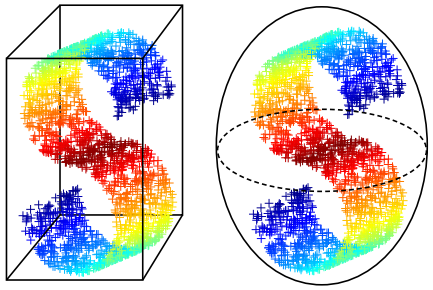


Fig. 2. Cube and ellipsoidal descriptor for the S-surface IG.

A manifold descriptor is a set of landmark points selected from an IG, which are decided by first sampling the images with equal population (or distance) interval on the IG's low-dimensional embedding, and then tracking back the images to the original representation space. In the next section and Section VII, Fig. 3(c) and the first panel of Fig. 7 illustrate the images sampling on the low-dimensional embedding; Fig. 3(d) and the second panel of Fig. 7 depict the sampled landmark points among the original data as a manifold descriptor. This proposed description can avoid low coverage and specificity, as well as eliminate incorrect reconstruction.

The idea of landmark points has been widely used in computer vision society, e.g., in image representation [16],

diffeomorphic point set registration [17], and so on. We borrow their ideas to describe the concave-shaped IGs here.

A. Manifold Description of the IGs

We use a sketch approach to derive a concise description of the formed IGs. By "sketch," we mean that the local neighboring data points lying on nonlinear space are described with a corresponding linear patch. As previously mentioned, LLE can find the intrinsic dimensionality of a manifold while keeping the order of the appearance of the data points unchanged. Therefore, the process of constructing the manifold descriptor includes three steps as follows.

Step 1: Apply LLE on the dataset to find the underlying embedding of lower dimensionality. LLE algorithm needs to specify only two parameters, one is the number of neighbors (denoted as k) and the other is the dimensionality d of the target embedding. k is manually set by users, and d can be either manually set or automatically determined. When the data points are uniformly sampled from a manifold, the number of neighbors k_ϵ and a small ϵ should satisfy the relationship $k_\epsilon \propto \epsilon^d$, where ϵ is the distance to determine two points as neighbors and d is the intrinsic dimensionality [8]. Using this idea, some methods to estimate the intrinsic dimensionality have been available (e.g., [18]).

Step 2: Sample the representative landmark points on the embedding. We first divide the range of the first dimension of the embedding into S_1 line segments of equal length, and then the dataset is split into S_1 small blocks. For each subset derived with respect to the first dimension, the range of the second dimension is equally divided into S_2 segments, and so on. The dimensionality-dividing process is performed for each dimension of the embedding, thus a collection of "ideal landmarks" (virtual data points) is derived as a result. But these ideal landmarks are not directly useful for the manifold descriptor, because there is no explicit coordinate transformation from the embedding to the original manifold (see the end of Section II-C). Therefore, we search a nearest real data point for each ideal landmark in the embedding, within one of the $N^{lp} = \prod_{i=1}^{d} S_i$ small subsets derived by splitting across each dimension. The method of sampling landmark points from the embedding is described in Algorithm 2.

Step 3: Track back the original data points on the manifold by using their indices. Thus, the linear patches defined by the sampled landmark points on the original manifold are the required descriptor.

An example of constructing the manifold descriptor is shown in Fig. 3.

Note that in Algorithm 2 we have $\prod_{i=1}^{d} S_i$ patches and $\prod_{i=1}^{d} (S_i + 1)$ landmark points. That is, the landmark points and the patches are not in one-to-one correspondence. So, we let the last two landmark points along each dimension share the same patch.

B. Reconstruction From the Manifold Descriptors

There has been some research on the issue of generating artificial data points from the existing real data points (e.g., [19] and [20]). However, these purposes are different.

Algorithm 2: Sample Landmark Points From Y

Input: Embedding $Y(Y \subset \mathcal{R}^d)$, $d < D$; number of segmentations S_i on each dimension, $1 \le i \le d$.

Output: The landmark points L^y from Y, where $N^{lm} = |L^y| = \prod_{i=1}^{d} (S_i + 1)$; the index collection in each linear patch $\boldsymbol{I}_i^{lp}$, where $\sum_i \left| I_i^{lp} \right| = N,\ 1 \le i \le N^{lp}$.

```
CurSet = {Y};
// Split the embedding into subsets;
for i = 1 to d do
    SetCard = |CurSet|;
    for j = 1 to SetCard do
        CurSet_j=split CurSet_j along the i^th dimension into S_i subsets;
    end
end
I^lp=Record the indices in CurSet;
//Decide the splitting length on each dimension;
for i = 1 to d do
    L_i^dim = (max(Y(:, i)) − min(Y(:, i)))/S_i;
end
//Find the landmark in each linear patch;
for i = 1 to N^lp do
    for j = 1 to d do
        SegInd=Decide the index of segmentation on j^th dimension for i^th linear patch with modulus and quotient w.r.t. {S_i};
        vp_j = (min(Y(:, j)) + (SegInd − 1) ∗ L_j^dim;
    end
    vp = (vp_1, vp_2, ..., vp_d);
    L_i^y= find the nearest point to vp in CurSet_i;
end
```

They use the synthetic examples to conduct semisupervised learning to improve the learning performance when the labeled data are not adequate. We reconstruct the data points here for the purpose of recovering the original data points, which is more similar to the imprecise decompression. Following this description-reconstruction approach, one may head for a unified compression-decompression framework for generic type of data. And later, we will evaluate the manifold descriptor based on the reconstructions.

Once the linear patches (the manifold descriptor) are formed, one can randomly generate data points on each patch with the population equals to that of X falling in the corresponding *CurSet* in Algorithm 2. However, because it is computationally expensive to randomly sample from a curved surface when the surface equation is unknown, we instead use a linear generation strategy illustrated in Fig. 4. Here, we start by uniformly arranging 8 data points between the two ends on both sides along the first dimension, and each pair of the points (including the landmark points and the ends) on the opposite side determine a line $l_i (1 <= i <= 10)$ along the second

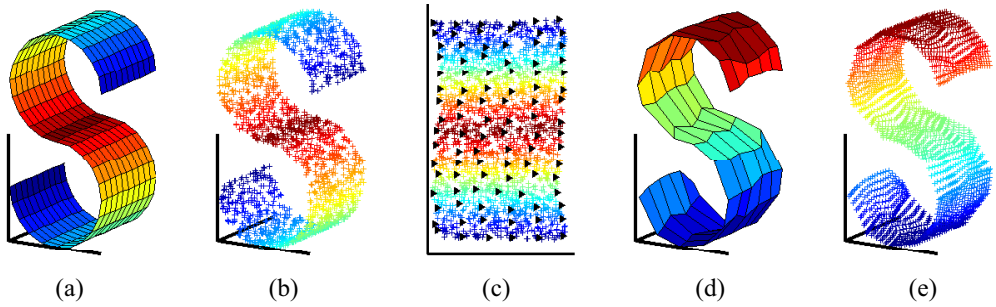


Fig. 3. (a) S-surface manifold appeared in [9]. (b) Randomly sampled 2000 data points X from the manifold. (c) Embedding find by LLE, and $16 * 6$ landmark points (black filled triangles) found by Algorithm 2. (d) Manifold descriptor of the dataset X. (e) Reconstruction of the data points $\tilde{X}$ to approximate dataset X.

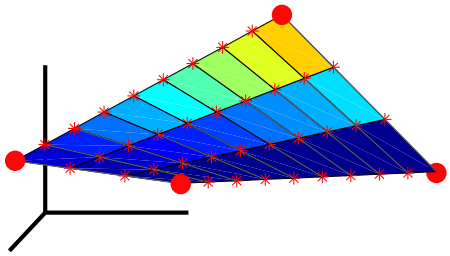


Fig. 4. Illustration of data points reconstruction on a single patch. Conventionally, a curve surface is best approximated by a collection of triangles. However, for the consistence with higher dimension situation, quadrilaterals are used here.

dimension. Then, two points are uniformly generated between the two ends along each l_i.

The number of points along each dimension is determined by the cardinality of the corresponding subset of objects and the ratio of line segment length on each dimension of this patch. If the total number does not fit in the length ratio exactly, we arrange the residual data points on the last line of the last dimension. Although the discussion is on a 2-D patch, it is not difficult to extend this method to higher dimensionality, where the patch is not a surface anymore, but a 3-D cube or hyper-cube with dimensionality greater than three.

C. Evaluate the Descriptors

With the computed manifold descriptors, one can reconstruct the data points $\tilde{X}$ using the method presented in Section IV-B. The descriptor is evaluated by determining the dissimilarity between the original dataset X and the reconstructed dataset $\tilde{X}$. The dissimilarity metric (named *sketch error*) is inspired by EMD [11], and defined as

$$\text{SketchError} = \sum_{i=1}^{N_{lp}} \sum_{j=1}^{N(lp_i)} \sum_{k=1}^{N(lp_i)} \frac{\left\| x_{ij} - \widetilde{x}_{ik} \right\|_2}{N(lp_i)} \tag{13}$$

where $\bigcup_{i,j} x_{ij} = X$, $\bigcup_{i,j} \tilde{x}_{ij} = \widetilde{X}$, and $\sum_i N(lp_i) = N$. For each linear patch, the indices of $\{x_{ij}\}$ are recorded in $\boldsymbol{I}^{lp}$ (see Algorithm 2).

We observed in experiments that the more landmark points sampled, the lower sketch error, which means better approximation of the original data. This is in accordance with our common sense. With many choices of the N^{lp}, one may seek to find an optimization standard to determine its value. We hereby propose a principle to choose N^{lp}.

The smaller N^{lp}, the better, if the requirement of a particular application is met.

For example, if the application requirement is "seeing the overview of the distribution of the data," then we would say the reconstruction in Fig. 11(d) is better than that in Fig. 11(f).

V. Interpretability of the LoDOG IGs

Generally, the semantic of the LoDOG IGs can be interpreted as the collection of subtrees split from the whole LT built from all the data. Quantifying of the interpretability of IGs is still an open problem [21]. Instead researchers from this literature usually compute the two elements of the justifiable granularity, namely coverage and specificity [4]. Since the subtrees are arbitrarily shaped, and all the data points that belong to an IG are connected into the corresponding subtree, we can conclude that every LoDOG IG (as a tree) has the coverage of 100%. However, the specificity of the IGs is intractable because it is hard to develop a general geometrical description for the closure of each subtree.

When an IG can be described with the manifold descriptor, we have an alternative method to quantify the coverage and specificity. Since the manifold descriptors are a collection of linear patches, one can use the amount of data lying on the patches as the coverage, and use the summation of the length, area, or volume of the linear patches to define specificity. But directly counting the data points that strictly lie on the patches is not reasonable, because this number simply equals to the number of landmark points, and fails to represent well the semantic of coverage. Therefore, we propose to relax the position requirement on data points to compute coverage and define an ε-equivalent coverage (denoted as coverage$_\varepsilon$) based on sketch error. Intuitively, smaller sketch error means that the reconstructed data simulate the original ones better (see Fig. 11), and also can be interpreted as that the original data points lie nearer to the linear patches, hence greater ε-equivalent coverage.

To define ε-equivalent coverage, we virtually project some data onto the linear patch and drag the others further from the patch, keeping the overall EMD unchanged. This idea is illustrated in Fig. 5.

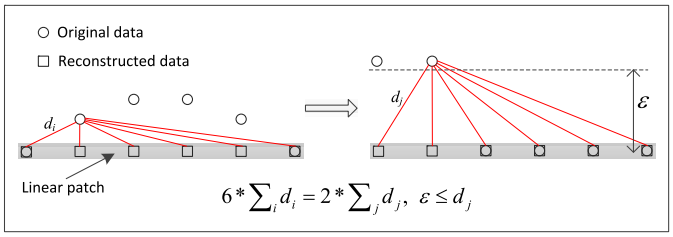


Fig. 5. Illustrative example of ε-equivalent coverage on a linear patch. In practice, the parameter ε is decided adaptively with a percentile technique.

From Fig. 5 and (13), one can write

$$N(lp_i) * \text{SketchError}_i \geqslant N_{\text{far}_i} * N(lp_i) * \varepsilon \tag{14}$$

where N_{far_i} is the number of data points being dragged outside the ε-neighborhood of the linear patch, and SketchError_i is for the linear patch i. So we can get

$$N_{\text{far}_i} \leqslant \text{SketchError}_i/\varepsilon. \tag{15}$$

Finally, it is obvious that

$$\text{coverage}_\varepsilon(\Omega_m) = N_m - \sum\nolimits_i N_{far_i} \geqslant N_m - \frac{\text{SketchError}}{\varepsilon} \tag{16}$$

where N_m is the size of Ω_m. In practice, we can use the lower bound to estimate $\text{coverage}_\varepsilon(\Omega_m)$.

The specificity of IG Ω_m is defined as

$$\text{specificity}(\Omega_m) = \left(\sum\nolimits_{j=1}^{N_m^{lp}} \text{size}\left(LP_j\right)\right)^{-1} \tag{17}$$

where $\text{size}(LP_j)$ means the geometrical magnitude of linear patch j, namely length for a line segment, area for a quadrilateral, and volume for a hexahedron.

Using this approach to quantify the interpretability, we will show in the experiment section that less linear patches lead to lower coverage and higher specificity, while more linear patches lead to higher coverage and lower specificity. This is in accordance with the principle of justifiable granularity.

VI. Time Complexity and Relationship to Related Works

A. Complexity Analysis on LoDOG

The time complexity of LoDOG algorithm consists of two parts. One is for the construction of the LT, the other is for computing the objective function value for every possible number of granules. As stated in [1], the time complexity of constructing an LT is $O(DN^2)$, where N is the size of the dataset X and D is the dimensionality of the space in which X are embedded.[1] We do not need to evaluate function J on every i from 1 to N in Algorithm 1, because too many (say thousands of, or even more) granules would violate the spirit of GrC. So, one can empirically set a maximum value MaxNg for possible greatest N_g^{opt}. The complexity for computing $\sum_{i=1}^{\text{MaxNg}} \text{DistCost}(\Omega_i)$ is $O(N)$, and that for incrementally computing $\sum_{i=1}^{Ng} \text{DistCost}(\Omega_i)$ for Ng from $\text{MaxNg}-1$ to 2 is $O(\text{MaxNg})$. So the complexity for the second part is $O(N+\text{MaxNg})$. Therefore, the time complexity of LoDOG is

$$\text{TC}_{\text{LoDOG}} = O\left(DN^2 + N + \text{MaxNg}\right) \approx O\left(DN^2\right). \tag{18}$$

By contrast, ElliGra needs to sweep through a range of k from 2 to MaxNg. For each k, it needs to perform an fuzzy c-means (FCMs) or Gustafson-Kessel (GK)-clustering to find prototypes, and perform a differential evolution (DE) optimization to decide the IGs' shape before scoring the corresponding granulation solution. Therefore, the time complexity of ElliGra (FCM version) is

$$\begin{aligned}\text{TC}_{\text{EllipGra}} &= O\left(\sum_{k=2}^{\text{MaxNg}} (NkD + MSQ)\right) \\ &\approx O\left(Nk^2D + kMSQ\right)\end{aligned} \tag{19}$$

in which, Nk^2D is for most efficient FCM [22] and $kMSQ$ is for the DE algorithm. M is the number of iterations; S is the population of the swarm; and Q is the length of the solution coding.

B. Complexity Analysis on the Manifold Descriptor

When analyzing the complexity of the manifold descriptor related tasks, we take into account three aspects: 1) sampling the landmark points on the embedding; 2) reconstructing the finest-grained data points; and 3) evaluating the manifold descriptor. The complexity of LLE, for straightforward implementation without additional effort to reduce the complexity, consists of three parts reflecting the three steps of the construct, namely

$$\text{TC}_{\text{LLE}} = O\left(DN^2 + DNk^3 + dN^2\right) \approx O\left(DN^2\right) \tag{20}$$

where k is the number of nearest neighbors, and d is the dimensionality of the target embedding [8]. Another key step in the descriptor construction is sampling the landmark points from the embedding. One can find out that the complexity of Algorithm 2 [sampling the landmarks from embedding (SLME)] is

$$\text{TC}_{\text{SLME}} = O\left(NN^{lp} + N^{lm}\left(d + N/N^{lp}\right)\right) \approx \left(NN^{lp}\right) \tag{21}$$

where N^{lp} is the number of linear patches and N^{lm} is the number of landmark points. The step of tracking back the indices of landmark points from Y to X does not require any actual operations. Therefore, the overall complexity for constructing manifold descriptor is

$$\text{TC}_{\text{ManiDes}} = TC_{\text{LLE}} + TC_{\text{SLME}} \approx O\left(DN^2\right). \tag{22}$$

As described in Section IV-B, the process of reconstruction of data points on original manifold is linear to the number of dimensionality D and linear to the size of the dataset N, that is

$$\text{TC}_{\text{Recons}} = O(DN). \tag{23}$$

To evaluate the quality of the reconstruction with the metric sketch error, we have to compute the regularized EMD

[1]If X is not sampled from a D-dimensional space, such as the case of social networks, then the complexity of computing distance matrix should be discussed otherwise.

on each linear patch. Thus, the complexity for evaluating the reconstruction is

$$TC_{\text{SketchErr}} = O\left(N^{lp} D\left(N/N^{lp}\right)^2\right) = O\left(DN^2/N^{lp}\right). \quad (24)$$

C. Relationships to Other Researches

1) LoDOG is applicable to all the contexts where the assumption of DPClust holds: "cluster centers are surrounded by neighbors with lower local density and that they are at a relatively large distance from any points with a higher local density" [12]. The choice of the value of the parameter α may affect N_g^{opt}. Higher values of α yield smaller N_g^{opt}, and vice versa. Therefore, different values of α would lead to the similar hierarchical clustering as in [7]. With LoDOG, the data points are granulated as N_g^{opt} micro-clusters to achieve the minimum value of function J in (9), which can capture the hierarchical characteristic of the data distribution with varying N_g^{opt}. When we previously used DPClust to cluster data streams [1], a static n_f (number of the fat nodes) policy is applied. This paper may be improved if n_f is adaptively determined by LoDOG.
2) The manifold-landmark descriptor and reconstruction share some common ideas with sparse representation [23], [24]. They both select some typical samples to represent all the data. However, the difference between the two approaches is also apparent. Our descriptor aims to efficiently sketch the whole data, and the presentation are localized and in linear complexity. By contrast, sparse representations seek to more accurately present the whole dataset with a small number of prototypes, therefore they have to optimize over the whole dataset by solving a series of equations.
3) Manifold descriptor outperforms other descriptors with respect to the metric sketch error where the data points are generated from a manifold. The reason is that the manifold descriptor reflects the real shape of the dataset by sampling the representative landmark points, while the comparative models use a general method to include the data points in hyper boxes or hyper ellipsoids. However, our manifold descriptor would lose its advantages if the dataset is generated otherwise. In this situation, one can use an approach similar to over-complete dictionary [24] to represent the IGs.
4) ElliGra [5] is still a reasonable choice when compared with our methods if the dataset actually contains only ellipsoidal IGs, because it has wrapped both granulation and representation into one procedure. Therefore, ElliGra is easier to be understood and implemented. By contrast, our methods need to model granulation and representation separately, and the underlying mechanisms behind them are a little harder to understand.
5) LoDOG constructs IGs with much different mechanism from fuzzy sets. Because fuzzy sets IGs are derived from natural-language-described concepts and a corresponding fuzzy graph, and each datum may bear several fuzzy IG labels with different memberships [25]. By contrast, every object belongs to only one crisp LoDOG IG. Rough sets can build various types of IG, since there are many extensions from classic Pawlak's [26] rough set model, among which LoDOG shares most common characteristics with neighborhood rough sets [27] because it considers the distance between any pair of objects and a cut-off distance parameter as well. We may make a closer investigation between LoDOG and neighborhood rough set in the future.

Although the IGs derived from different models have different inner structures, there have been some researches considering the communications between the heterogeneous IGs. Recently, Qian *et al.* [28] proposed the difference measure between granular structures and put it into the k-means framework to group the IGs derived from different models.

TABLE I
DATASETS USED IN EXPERIMENTS

Name	Source	Dimension	# Objects	# IGs
Curves	Artificial	2	764	4
ToyDS	Artificial	2	59	7 or 2
Ellipses	Artificial	2	338	5
S-Surface	Artificial	3	2000	1
USPS (subset)	Real	256	3300	3
PIE (subset)	Real	1024	170	–
AstroPh	Real	2	396,160	9 or 7

VII. EXPERIMENTAL STUDIES

All the experiments are conducted on a PC equipped with 8-GB memory and an Intel i5-2430M CPU. Operating system is Windows 7 and the programming language is MATLAB. The effectiveness of LoDOG and the manifold description of the IGs is validated on seven datasets, among which four are artificial and the other three are from real world. The brief information on the seven datasets is summarized in Table I. When demonstrating the results of FCM and GK-clustering in our experiments, we employ the standard implementations of FCM and GK-clustering from the MATLAB toolbox developed by Balasko *et al.* [29]. The codes are available upon request.

A. Artificial Datasets

1) Curves Dataset: The curves dataset is composed of four curves: 1) the left one is a quadratic curve generated by $y = x^2$ with ensuing rotation and translation; 2) the middle two are both sampled from $y = \sin(x)$ with domain of $(0, \pi)$; and 3) the right one is an ordinary spiral curve generated by the parametric equations $x = r*\cos(\theta)$, $y = r*\sin(\theta)$ with r and θ growing simultaneously. This curves dataset is used to show the robustness of the LoDOG against parameter α and the capability of LoDOG to detect arbitrarily shaped IGs. Besides, the quadratic curve in the dataset is used to test the manifold description of IGs.

As shown in Fig. 6, LoDOG automatically finds the correct number of IGs and accurately detects the four curve IGs that perfectly match human's intuition. The optimal solution of

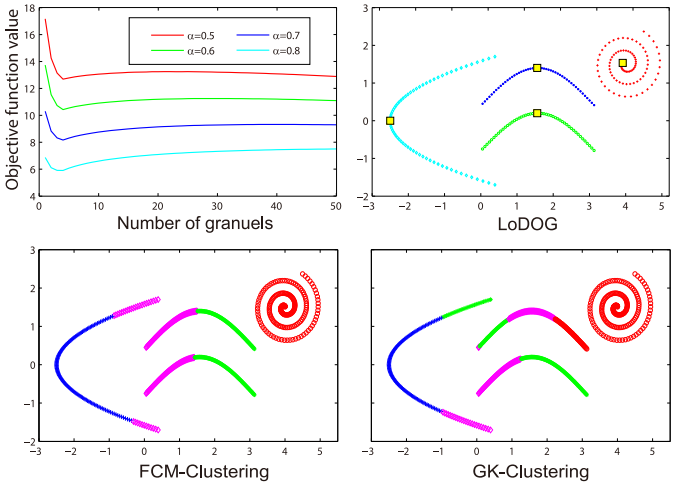


Fig. 6. LoDOG result on curves dataset (top panel) and FCM/GK-clustering results on this dataset (bottom panel). The filled yellow squares are the centers and this convention goes for other figures throughout this paper.

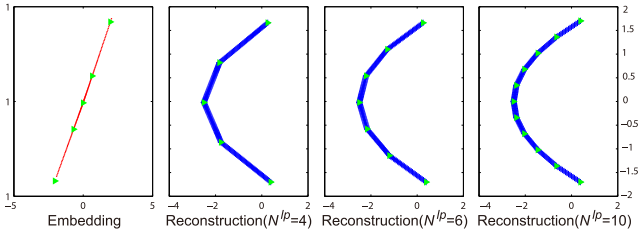


Fig. 7. Reconstructions of the quadratic curve in curves dataset with different granularity. The filled green triangles are the landmark points.

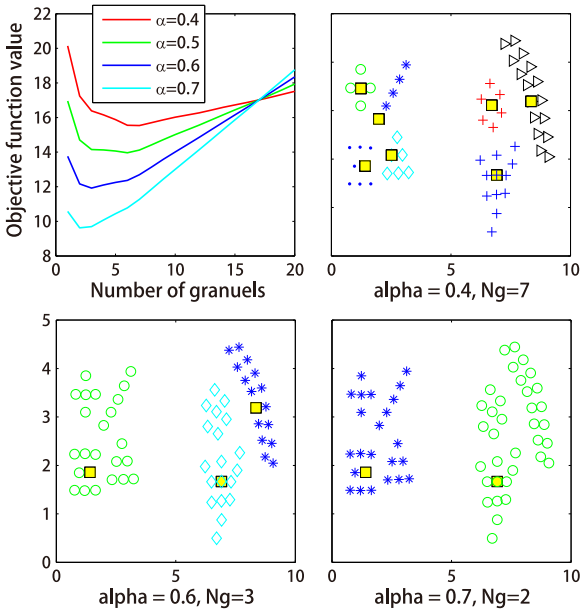


Fig. 8. LoDOG as an approach to hierarchical clustering by tuning the parameter α.

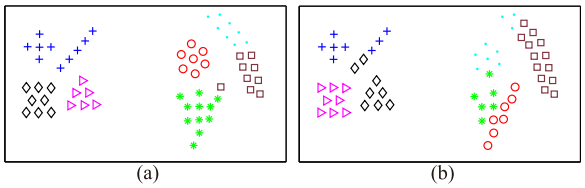


Fig. 9. Clustering results of (a) FCM and (b) GK-clustering, where number of clusters is set to be 7.

N_g keeps unchanged when the parameter α varying in a wide range from 0.5 to 0.8. However, FCM and GK-clustering cannot find the right IGs even the number of clusters is manually set to 4.

We reconstruct the quadratic curve, denoted as X_{quad}, without knowing its underlying generating function (see Fig. 7). With the method described in Section IV, we generate the points on the line segments determined by the landmark points sampled from the embedding, to imitate the original data points. The number of linear patches N^{lp} has assumed the values from {4, 6, 10}. One can find out that the similarity between X_{quad} and the reconstructed dataset ($\tilde{X}_{\text{quad}}$) grows along with the increasing of N^{lp}. When using the descriptor that consists of 11 landmarks to present the original 171 data points, we can hardly tell the difference between X_{quad} and $\tilde{X}_{\text{quad}}$.

Finally, we quantify the dissimilarity between X_{quad} and $\tilde{X}_{\text{quad}}$ using the metric sketch error defined in (13). The sketch errors are 102.38, 69.23, and 44.57, when N^{lp} equals to 4, 6, and 10, respectively. For easy reference, we also put all sketch errors, coverage$_\varepsilon$, and specificity about the manifold descriptions in Table II.

2) ToyDS Dataset: The second artificial dataset is meant to demonstrate that LoDOG can serve as an efficient hierarchical clustering approach as well. For this dataset, we can observe that different choices of the parameter α in (9) lead to different levels of granularity of the IGs. Higher value of α implies that the objective function emphasizes N_g, hence the smaller number of IGs (see Fig. 8). Also with this toy dataset, we can show that the granulation method-based FCM or GK-clustering cannot find reasonable IGs (see Fig. 9).

Another hierarchical clustering method named DenPEHC [7] is also based on the structure of LT. In DenPEHC, the cluster hierarchy is determined through the analysis of γ value curve, which involves two parameters (*LocalR* and *GlobalR*). And the foundation of DenPECH is based on observations of the so-called "stairs" in the γ curve, so it is empirical and heuristic. In contrast, LoDOG has sounder rationale because of the formally defined optimization objective function (9). Additionally, LoDOG can achieve the hierarchical clustering only by varying one parameter α, so it is easier to manipulate and interpret the result.

3) Ellipses Dataset: This dataset is used to demonstrate that LoDOG can also efficiently and accurately find IGs in the ellipse-shaped data. As shown in Fig. 10, the dataset is granulated into five IGs in the same way as human's perception. ElliGra with FCM kernel (EG_{FCM}) and ElliGra with GK-Clustering kernel (EG_{GK}) are both able to find the same granules as LoDOG, but their time consumptions are longer than LoDOG (EG_{FCM} and EG_{GK} takes 4.9 times and 119 times as long as LoDOG, respectively). All the details about time consumptions of the competing models and LoDOG on the five datasets are collected and shown in Table IV.

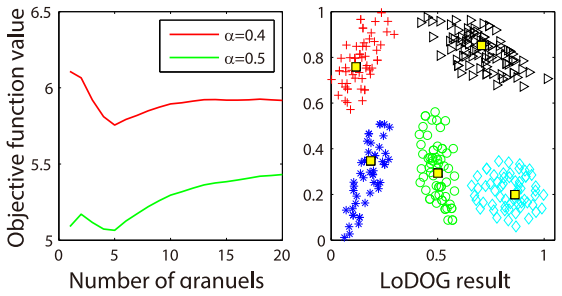


Fig. 10. LoDOG result on the ellipses dataset.

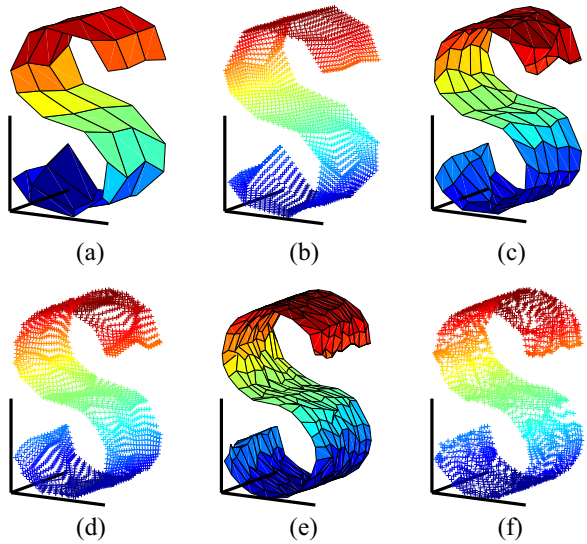


Fig. 11. Manifold descriptors and reconstructed dataset $\tilde{X}$ for three granularities. (a) and (b) $N^{lp} = 12 * 3$. (c) and (d) $N^{lp} = 20 * 5$. (e) and (f) $N^{lp} = 30 * 10$.

4) S-Surface Dataset: We have shown the manifold description and reconstruction of a curve on 2-D plane with curves dataset in Fig. 7. Here we continue to show the manifold descriptor construction and the reconstruction of data points, using Algorithm 2 to sample the landmark points, on the 3-D S-surface appeared in [9]. The descriptor and the reconstructions are on three granularities. As usual, the more landmark points sampled, the smaller sketch error we got (see Fig. 11 and Table II).

Intuitively, the coverage and specificity of the IG descriptor formed by a set of linear patch are much higher than that by a box or ellipsoid (see Fig. 2). But it is not suitable to compare the two approaches with the same evaluation metric, because for the manifold descriptor the "volume" used to measure specificity has become a 2-D area. Therefore, we turn to use the proposed sketch error to measure the quality of a manifold descriptor. Besides, it is not appropriate either to compare the descriptor in terms of sketch error with the ellipsoidal or hyper-box approaches, because in this situation their reconstruction will be totally dissimilar to the original data.

B. Real Datasets

1) USPS Dataset: USPS dataset has 11 000 samples of 16×16-pixeled gray images for handwriting digits "0"–"9" (1100 images for each). This dataset is quite suitable for IGs' manifold description because the observations are sufficiently sampled. As in [8], we first choose three digits (0, "5," and 9 in this paper) to form a subset, and then employ LLE algorithm to reduce the dimensionality of the data to 2 (initially the d parameter in LLE is set to 3, and then only the first two dimensions are remained because the third dimension takes almost the same value).

TABLE II
EVALUATION OF THE MANIFOLD DESCRIPTIONS

IG (#Points, ε)	d_Y	N^{lp}	Sketch Error	Coverage$_\varepsilon$	Specificity
Curves (171, 1.57)	1	4	102.38	106	6.75^{-1}
		6	69.23	127	6.80^{-1}
		10	44.57	143	6.96^{-1}
S-Surface (2000, 2.69)	2	12*3	1382	1486	44.86^{-1}
		20*5	922	1657	50.57^{-1}
		30*10	635	1764	68.16^{-1}
USPS '5' (1100, 6.86)	2	10	6165	201	62.66^{-1}
		20	6122	208	137.16^{-1}
		50	5888	242	336.45^{-1}
USPS '9' (1100, 7.12)	2	10	5913	270	69.32^{-1}
		20	5804	285	119.91^{-1}
		50	5648	307	282.11^{-1}

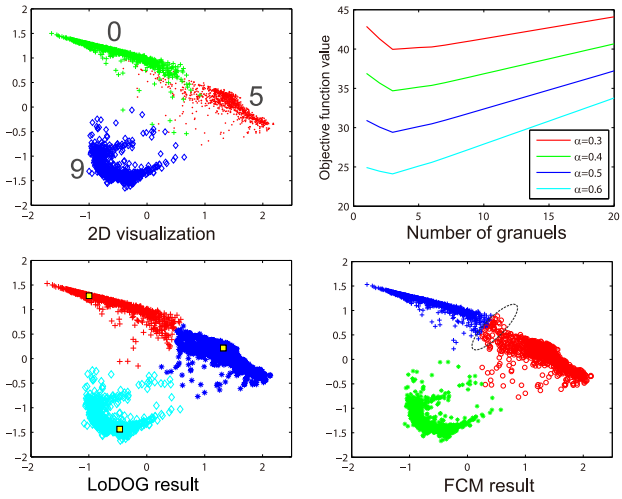
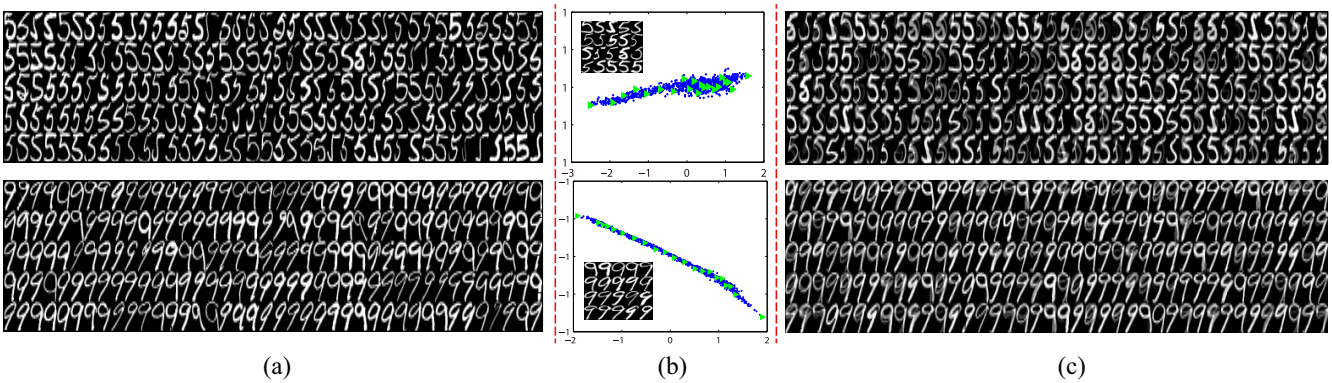


Fig. 12. LoDOG and FCM results on the USPS handwriting digits (0, 5, and 9).

The result of dimensionality reduction is shown in the top-left panel of Fig. 12. We can find out the digits are well separated in the 2-D space except for a few overlaps, so the ensuing granulation has chance to yield good result. It is clear, however, the shapes of the clusters are not spherical or ellipsoidal, so the classical FCM or GK method may have some data wrongly clustered. The misclustered region of the digit 0 is marked by a dashed ellipse in the bottom-right panel of Fig. 12. The experimental result also shows that LoDOG leads to a sound granulation when the parameter α varies from 0.3 to 0.6.

The data points of 5 and 9 are reconstructed with {10, 20, 50} landmark points, respectively. The corresponding sketch



(a) (b) (c)

Fig. 13. Manifold descriptors and results of reconstruction of handwriting digits (5 and 9). The subset of the first 200 images in each digit are displayed for illustration. The filled green triangles in (b) are the landmark points. (a) Original digits (5 and 9) from USPS dataset. (b) Landmark points in X and LLE $(X, 18, 2)$. (c) Reconstructions of digits (5 and 9) with the manifold descriptors.

errors are tabulated in Table II. Although the size of the landmarks are relatively small with respect to the total amount of the original data (20 versus 1100), we see the reconstructions are quite similar to the original data, except for a few blurs and ghosts. In Fig. 13(b), the landmark points are visualized to show their representing power. In other words, we find out that the image of each landmark point is much different from others. Therefore, with these typical shapes of a particular digit, all other shapes of this digit can be approximately reconstructed by linear combinations of neighboring landmark points. As discussed in Section VI-C, one can find that this approach shares similar idea with sparse representation, but their purpose and implementation are not the same.

2) PIE Face Dataset: The CMU PIE face dataset includes 20 000 face images of 68 subjects collected at different pose, illumination, and expression conditions. We select 170 images of the first female subject and apply LoDOG to granulate the set of images, since one can find that there are many groups of images sharing very similar appearances. The images are first downsampled to a lower resolution of 32×32, then their dimensionality is reduced to 3 using standard LLE. But only the first two dimensions are used in LoDOG processing because the last dimension takes almost equal values (like in USPS dataset).

As in Fig. 14, the images can be granulated into 15 or 4 IGs, when α takes the value of 0.3 or 0.4, respectively. We choose to display the result of 15 IGs in Fig. 15, from which we can note that quite a few IGs have high similarities among their elements (e.g., IG_1, IG_3, IG_5, IG_11, IG_14, etc.), despite some IGs that have one or two apparently different images in themselves, e.g., IG_7, IG_9, and IG_15. Overall, LoDOG granulates the total 170 images into 15 groups, and the similarity within each group is obvious. This demonstrates well that GrC is helpful in supporting human perception and description.

Considering the vast combinations of poses, illuminations, and expressions for an individual's face image, 170 photographs are far from sufficient to form a smooth and continuous manifold.[2] So we do not construct the manifold descriptors on the PIE subset, and the reconstruction process is skipped as well.

[2]Saul and Roweis [8] used 1965 images of a person to discover the underlying embedding.

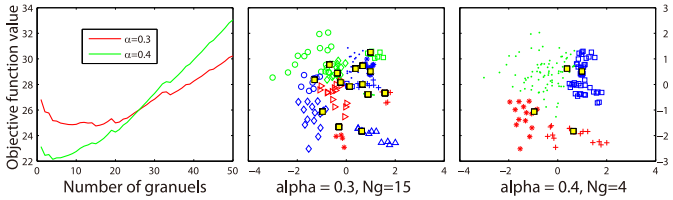


Fig. 14. Objective function value versus the number of granules, and the resultant IGs of LoDOG on the PIE subset (after dimensionality reduction).

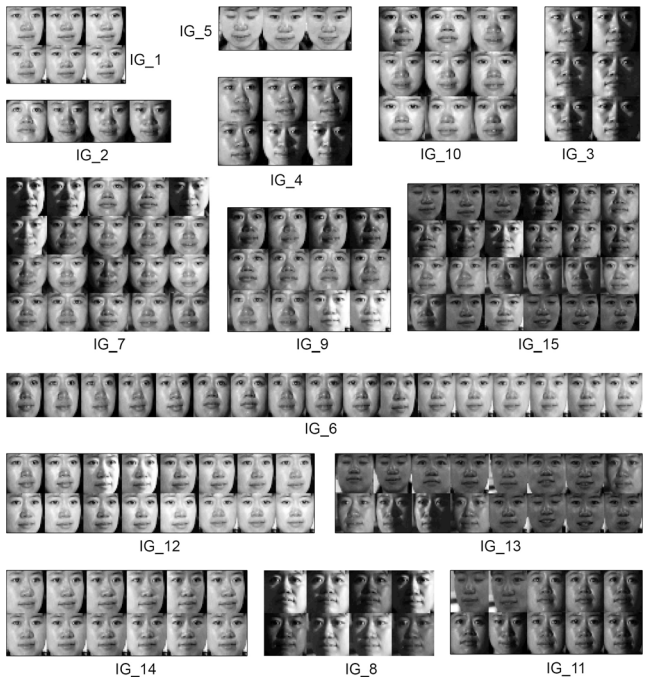


Fig. 15. Granulation result of LoDOG on the PIE subset coming in the form of original images.

For the convenience of checking details of the experiments, the parameter configurations in LoDOG are listed in Table III.

3) Social Network: Detecting communities in social networks has attracted much attention in recent years, due to its applications such as personalized recommendation,

TABLE III
PARAMETER CONFIGURATIONS IN LODOG EXPERIMENTS

Dataset	α	d_c percentile	$H(x)$	N_g^{opt}
Curves	[0.5, 0.8]	[0.2, 2]	$log(x)$	4
ToyDS	[0.4, 0.7]	6	x	{7, 6, 3, 2}
Ellipses	[0.4, 0.5]	[4, 8]	$log(x)$	5
USPS	[0.3, 0.6]	[3, 8]	x	3
PIE	[0.3, 0.4]	1	x	{15, 3}
AstroPh	[0.55, 0.7]	1	$50 * x$	{9, 7}

TABLE IV
COMPUTING OVERHEAD IN LODOG EXPERIMENTS

Dataset	Time (s)			Acceleration ratio	
	EG_{FCM}	EG_{GK}	LoDOG	EG_{FCM}	EG_{GK}
Curves	1.15	5.85	**0.27**	4.18	21.24
ToyDS	3.30	4.58	**0.12**	28.30	39.26
Ellipses	0.64	15.67	**0.13**	4.90	119.00
USPS	15.34	63.87	**3.25**	4.72	19.63
PIE	2.63	21.71	**0.18**	14.54	120.01
AstroPh	–	–	2.82	–	–

public opinion monitoring, and others. Clustering is a traditional method to find the community structure in social networks [13]. The communities in a social network can be disjoint or overlapping, so the methods of dealing with each situation are usually different, except that recently Chakraborty *et al.* [30] proposed GenPerm to solve the two problems within a single framework. Normally, the overlapping communities are expanded from the detected nonoverlapping ones (e.g., [31] and [32]).

LoDOG exhibits great potential in detecting communities from social network, because LoDOG is efficient in accurately determining the optimal IGs and DPClust is especially designed to take the distance matrix as input. Therefore, the request that X is embedded in a D-dimensional space is not a necessary condition, which makes LoDOG applicable to the networks that use a tuple $[\text{vertex}_1, \text{vertex}_2]$ to represent a correlation between two vertices.[3] To avoid diverging from the focus of this paper, we stick to nonoverlapping community discovery although overlapping communities may be more common (in fact, we would expand LoDOG to form overlapping communities in the future). For the same reason, we do not perform extensive comparisons with other state-of-the-art community detection methods here.

We use the dataset ca-AstroPh, downloaded from [33], which is about the collaboration relationship among the astrophysics researchers who have submitted papers to the e-print service *arXiv*. If researcher i and researcher j co-authored a paper, then a bigram $[i, j]$ is added to the data.

The distance metric used here is the *overlap* between the neighborhoods $\Gamma(i)$ and $\Gamma(j)$ of vertices i and j, given by the proportion of the intersection divided by the union of the neighborhoods [13], that is

$$d_{ij} = \frac{|\Gamma(i) \cap \Gamma(j)|}{|\Gamma(i) \cup \Gamma(j)|}. \tag{25}$$

Straightforward implementation of (25) has the time complexity of $O(N^3)$, so it needs to be accelerated by considering the sparsity of the adjacent matrix. The running time of LoDOG on the six datasets (ToyDS, curves, Ellipses, USPS, PIE, and AstroPh), and that of the competing model (EG_{FCM} and EG_{GK}) are shown in Table IV, from which we can see that LoDOG uses only a fraction of the time that EG_{FCM} consumes. EG_{GK} constantly takes a longer time than EG_{FCM} because it uses adaptive distance norm policy and has the ability to detect ellipsoidal clusters of arbitrary directions.

[3]The social network is not sampled from D-dimensional space, so it does not need consider the problem of manifold descriptor.

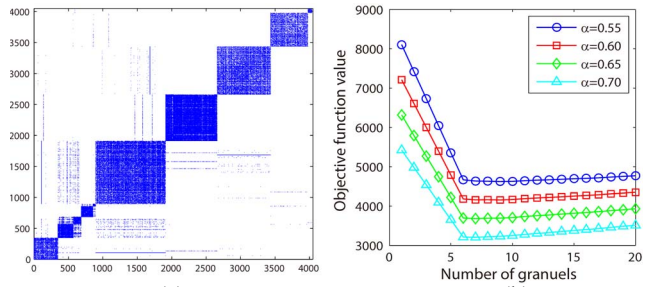


Fig. 16. (a) Visualization of the adjacent matrix of AstroPh network. (b) Objective functions versus the number of IGs in LoDOG.

The original data of AstroPh is arranged such that closely related authors have the neighboring indices. So, by visualizing the adjacent matrix, one could roughly see the ground truth of the communities in this collaborating networks, that is, there are nine or seven communities in it [see Fig. 16(a)].

The experiment is conducted as follows. First, the input (in the form of $[\text{vertex}_1, \text{vertex}_2]$) is transformed into an adjacent matrix. Then, the distance between each pair of vertexes is computed via (25), thus the distance matrix Dist becomes formed. At last, LoDOG takes Dist as input and yields the granulation result as indicated in Fig. 16(b), with the corresponding parameters configuration listed in Table III. LoDOG finds the correct number of communities.

The experiment is performed on the whole AstroPh dataset, but for clarity we only choose a subnetwork (the first 900 authors) to visualize. The free software NetDdraw [34] is used here, but the vertices are tagged by the result of LoDOG. As shown in Fig. 17, one can find out that the result of LoDOG is in accordance with the layout of the graph, except for a few red-triangle vertices. This convincingly demonstrates the effectiveness of LoDOG in the task of community detection. Using the *modularity* metric defined in [35], we obtained a Q value of 0.683, while the typical values fall in the range from about 0.3 to 0.7 [35]. Therefore, it is quantitatively verified that LoDOG achieves a good performance on the AstroPh dataset.

C. Scalability of LoDOG

Like many other machine learning models, LoDOG has the bottle neck of computing the distance matrix that requires $O(N^2)$ time complexity. However, the positive information is that scientific computing platforms (such as MATLAB, Octave, R, etc.) and CPU manufacturers have made great effort

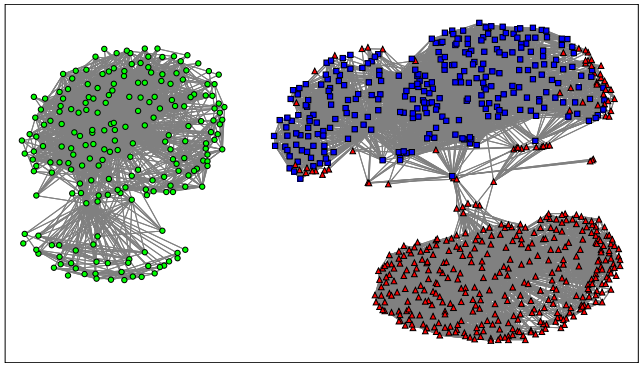


Fig. 17. Granulation result of LoDOG on the subnetwork of AstroPh.

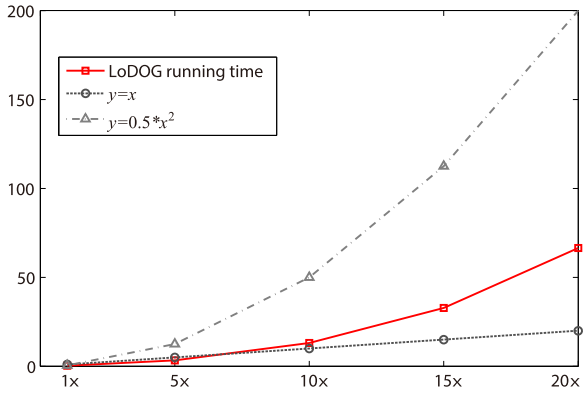


Fig. 18. Running time growth trend on the five versions of curves dataset (each contains 764, 3820, 7640, 11 460, and 15 280 data points, respectively).

to improve the efficiency of distance matrix computing, due to its widespread application [36]. We illustrate the scalability of LoDOG by testing the running time on the dataset curves and its 4 more intensively sampled versions (with the size of 5×, 10×, 15×, and 20×, respectively). The running time growth against the data volume is shown in Fig. 18, from which one can roughly read that time consumption of LoDOG is between quadratic and linear. It approximately has the form of ξN^2 with ξ much less than 0.5.

In the big data context, linear complexity algorithms are desirable. The precise version of LoDOG still have the difficulty of $O(N^2)$ time and space complexity, despite the small coefficient ξ. So, there need some parallel computing platforms and/or approximate technologies (e.g., local sensitive hashing [37], [38]) to help LoDOG scale for big data. We plan to address this issue in the future.

VIII. Conclusion

This paper presented an efficient and accurate granulation method based on the LT structure, and also proposed was a manifold IG descriptor and the reconstruction of data points if the data have an underlying distribution from a manifold. To evaluate the quality of reconstruction, we developed a metric (*sketch error*) to measure the dissimilarity between the reconstructed and the original data. The interpretability is quantified using approximate coverage and specificity when a LoDOG IG has its manifold descriptor. The proposed method was compared with the state-of-the-art models through the theoretical analysis and empirical validation. It showed that the proposed method is more efficient and able to detect any shape of IGs. The proposed manifold IG description can faithfully reflect the distribution of the original data. The relationships between LoDOG and another hierarchical clustering method DenPEHC, and between the manifold IGs description and sparse representation, were discussed.

The potential of LoDOG in detecting communities from social networks is explored in the experiment. We plan to extend this paper in several directions: 1) by applying it to online communities' discovery in large scale social networks and 2) further performing decision making algorithms (e.g., classification or regression) based on the resultant IGs in a big data context.

Acknowledgment

The authors would like to thank the anonymous referees for their valuable comments. The work of J. Xu was supported by the Guizhou Provincial Key Laboratory of Public Big Data.

References

[1] J. Xu, G. Wang, T. Li, W. Deng, and G. Gou, "Fat node leading tree for data stream clustering with density peaks," *Knowl. Based Syst.*, vol. 120, pp. 99–117, Mar. 2017.

[2] C. L. P. Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on big data," *Inf. Sci.*, vol. 275, pp. 314–347, Aug. 2014.

[3] W. Pedrycz and W. Homenda, "Building the fundamentals of granular computing: A principle of justifiable granularity," *Appl. Soft Comput.*, vol. 13, no. 10, pp. 4209–4218, 2013.

[4] W. Pedrycz, G. Succi, A. Sillitti, and J. Iljazi, "Data description: A general framework of information granules," *Knowl. Based Syst.*, vol. 80, pp. 98–108, May 2015.

[5] X. Zhu, W. Pedrycz, and Z. Li, "Granular data description: Designing ellipsoidal information granules," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2016.2612226.

[6] W. Pedrycz and A. Bargiela, "An optimization of allocation of information granularity in the interpretation of data structures: Toward granular fuzzy clustering," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 3, pp. 582–590, Jun. 2012.

[7] J. Xu, G. Wang, and W. Deng, "DenPEHC: Density peak based efficient hierarchical clustering," *Inf. Sci.*, vol. 373, pp. 200–218, Dec. 2016.

[8] L. K. Saul and S. T. Roweis, "Think globally, fit locally: Unsupervised learning of low dimensional manifolds," *J. Mach. Learn. Res.*, vol. 4, no. 2, pp. 119–155, 2003.

[9] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[10] X. He and P. Niyogi, "Locality preserving projections (LPP)," in *Proc. NIPS*, vol. 16. 2002, pp. 186–197.

[11] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, 2000.

[12] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[13] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, nos. 3–5, pp. 75–174, 2010.

[14] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[15] G. Peters, "Granular box regression," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 6, pp. 1141–1152, Dec. 2011.

[16] T.-W. Huang and H.-T. Chen, "Landmark-based sparse color representations for color transfer," in *Proc. IEEE Int. Conf. Comput. Vis.*, Kyoto, Japan, 2009, pp. 199–204.

[17] I. Kolesov, J. Lee, G. Sharp, P. Vela, and A. Tannenbaum, "A stochastic approach to diffeomorphic point set registration with landmark constraints," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 238–251, Feb. 2016.
[18] B. Kégl, "Intrinsic dimension estimation using packing numbers," in *Proc. NIPS*, 2002, pp. 681–688.
[19] I. Triguero, S. Garcia, and F. Herrera, "SEG-SSC: A framework based on synthetic examples generation for self-labeled semi-supervised classification," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 622–634, Apr. 2015.
[20] P. Melville and R. J. Mooney, "Creating diversity in ensembles using artificial data," *Inf. Fusion*, vol. 6, no. 1, pp. 99–111, 2005.
[21] M. J. Gacto, R. Alcalá, and F. Herrera, "Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures," *Inf. Sci.*, vol. 181, no. 20, pp. 4340–4360, 2011.
[22] J. F. Kolen and T. Hutcheson, "Reducing the time complexity of the fuzzy c-means algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 2, pp. 263–267, Apr. 2002.
[23] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.
[24] M. Aharon, M. Elad, and A. Bruckstein, "*K*-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Proces.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.
[25] W. Pedrycz, R. Al-Hmouz, A. Morfeq, and A. Balamash, "The design of free structure granular mappings: The use of the principle of justifiable granularity," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 2105–2113, Dec. 2013.
[26] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data*, vol. 9. Dordrecht, The Netherlands: Springer, 2012.
[27] Q. Hu, D. Yu, J. Liu, and C. Wu, "Neighborhood rough set based heterogeneous feature subset selection," *Inf. Sci.*, vol. 178, no. 18, pp. 3577–3594, 2008.
[28] Y. Qian *et al.*, "Grouping granular structures in human granulation intelligence," *Inf. Sci.*, vols. 382–383, pp. 150–169, Mar. 2017.
[29] B. Balasko, J. Abonyi, and B. Feil, *Fuzzy Clustering and Data Analysis Toolbox for Use With MATLAB*, Veszprem Univ., Veszprém, Hungary, 2008.
[30] T. Chakraborty, S. Kumar, N. Ganguly, A. Mukherjee, and S. Bhowmick, "Genperm: A unified method for detecting non-overlapping and overlapping communities," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 2101–2114, Aug. 2016.
[31] P. G. Sun, L. Gao, and S. S. Han, "Identification of overlapping and non-overlapping community structure by fuzzy clustering in complex networks," *Inf. Sci.*, vol. 181, no. 6, pp. 1060–1071, 2011.
[32] X. Wang, L. Jiao, and J. Wu, "Adjusting from disjoint to overlapping community detection of complex networks," *Phys. A Stat. Mech. Appl.*, vol. 388, no. 24, pp. 5045–5056, 2009.
[33] J. Leskovec and A. Krevl. (Jun. 2014). *SNAP Datasets: Stanford Large Network Dataset Collection*. [Online]. Available: http://snap.stanford.edu/data
[34] S. P. Borgatti, *NetDraw: Graph Visualization Software*, Anal. Technol., Harvard, MA, USA, 2002.
[35] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 2, pp. 1–15, 2004.
[36] J. J. Dongarra, J. Du Croz, S. Hammarling, and I. S. Duff, "A set of level 3 basic linear algebra subprograms," *ACM Trans. Math. Softw.*, vol. 16, no. 1, pp. 1–17, 1990.
[37] S. Har-Peled, P. Indyk, and R. Motwani, "Approximate nearest neighbor: Towards removing the curse of dimensionality," *Theory Comput.*, vol. 8, no. 1, pp. 321–350, 2012.
[38] J. Ji, J. Li, S. Yan, B. Zhang, and Q. Tian, "Super-bit locality-sensitive hashing," in *Proc. NIPS*, 2012, pp. 108–116.

Ji Xu received the B.S. degree from Beijing Jiaotong University, Beijing, China, in 2004, and the M.S. degree from Tianjin Normal University, Tianjin, China, in 2008. He is currently pursuing the Ph.D. degree with Southwest Jiaotong University, Chengdu, China.

He has published a number of papers in refereed international journals and conferences. His current research interests include data mining, granular computing, and machine learning.

Guoyin Wang (M'98–SM'03) received the B.S., M.S., and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1992, 1994, and 1996, respectively.

He was a Visiting Scholar with the University of North Texas, Denton, TX, USA, and the University of Regina, Regina, SK, Canada, from 1998 to 1999. Since 1996, he has been with the Chongqing University of Posts and Telecommunications, Chongqing, China, where he is currently a Professor, the Director of the Chongqing Key Laboratory of Computational Intelligence and the National International Scientific and Technological Cooperation Base of Big Data Intelligent Computing, and the Dean of the Graduate School. He has authored 15 books, edited dozens of proceedings of international and national conferences, and has over 200 reviewed research publications. His current research interests include rough set, granular computing, knowledge technology, data mining, neural network, and cognitive computing.

Dr. Wang was the President of the International Rough Sets Society (IRSS) for the period 2014–2017. He is the Chairman of the Steering Committee of IRSS, and the Vice President of the Chinese Association of Artificial Intelligence.

Tianrui Li (SM'10) received the B.S., M.S., and Ph.D. degrees from Southwest Jiaotong University, Chengdu, China, in 1992, 1995 and 2002, respectively.

He was a Post-Doctoral Researcher with SCK•CEN, Mol, Belgium, from 2005 to 2006, and a Visiting Professor with Hasselt University, Hasselt, Belgium, in 2008, the University of Technology Sydney, Ultimo, NSW, Australia, in 2009 and the University of Regina, Regina, SK, Canada, in 2014. He is currently a Professor and the Director of the Key Laboratory of Cloud Computing and Intelligent Techniques, Southwest Jiaotong University. He has authored or co-authored over 150 research papers in refereed journals and conferences. His current research interests include big data, cloud computing, data mining, granular computing, and rough sets.

Witold Pedrycz (F'98) received the M.Sc., Ph.D., and D.Sci. degrees from the Silesian University of Technology, Gliwice, Poland.

He is a Professor and the Canada Research Chair of computational intelligence with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He is with the Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland, where he is also a Foreign Member. He has authored 15 research monographs covering various aspects of computational intelligence, data mining, and software engineering. His current research interests include computational intelligence, fuzzy modeling, and granular computing, knowledge discovery and data mining, fuzzy control, pattern recognition, knowledge-based neural networks, relational computing, and software engineering. He has published numerous papers in the above areas.

Dr. Pedrycz was a recipient of the IEEE Canada Computer Engineering Medal, the Cajastur Prize for Soft Computing from the European Centre for Soft Computing, the Killam Prize, and the Fuzzy Pioneer Award from the IEEE Computational Intelligence Society. He is intensively involved in editorial activities. He is the Editor-in-Chief of *Information Sciences*, *WIREs Data Mining and Knowledge Discovery* (Wiley,) and the *International Journal of Granular Computing* (Springer). He currently serves as a member of a number of editorial boards of other international journals. He is a fellow of the Royal Society of Canada.