A Robust Density-based Clustering Algorithm for Multi-Manifold Structure

Jianpeng Zhang, Mykola Pechenizkiy, Yulong Pei, Julia Efremova
Department of Mathematics and Computer Science
Eindhoven University of Technology, 5600 MB Eindhoven, the Netherlands
{j.zhang.4, m.pechenizkiy, y.pei.1, i.efremova}@tue.nl

ABSTRACT

In real-world pattern recognition tasks, the data with multiple manifolds structure is ubiquitous and unpredictable. Performing an effective clustering on such data is a challenging problem. In particular, it is not obvious how to design a similarity measure for multiple manifolds. In this paper, we address this problem proposing a new manifold distance measure, which can better capture both local and global spatial manifold information. We define a new way of local density estimation accounting for the density characteristic. It represents local density more accurately. Meanwhile, it is less sensitive to the parameter settings. Besides, in order to select the cluster centers automatically, a two-phase exemplar determination method is proposed. The experiments on several synthetic and real-world datasets show that the proposed algorithm has higher clustering effectiveness and better robustness for data with varying density, multi-scale and noise overlap characteristics.

CCS Concepts

•**Information systems** → **Clustering;**

Keywords

cluster analysis; density peaks; manifold distance; power-law distribution, PauTa criterion

1. INTRODUCTION

The data with multi-manifold structure [7] is ubiquitous in real-world pattern recognition tasks, and designing an effective clustering algorithm for this type of data is a challenging problem. In many practical problems, e.g., handwritten digit recognition, image segmentation, and web analysis etc., clustering analysis is aimed at finding correlations within subsets of the dataset and assessing similarity among elements within these subsets. One of the most famous clustering methods is *K-means* [8] algorithm. It attempts to cluster data by minimizing the distance between cluster centers and other data points. However, K-means is known to be sensitive to the selection of the initial cluster centers and is easy to fall into the local optimum. K-means is also not good at detecting non-spherical clusters which has poor ability to represent the manifold structure. Another clustering algorithm is *Affinity propagation* (AP) [6]. It does not require to specify the initial cluster centers and the number of clusters but the AP algorithm is not suitable for the datasets with arbitrary shape or multiple scales. Other density-based methods, such as *DBSCAN* [5], are able to detect non-spherical clusters using a predefined density threshold. They can easily find clusters with different sizes and shapes. One of the most significant disadvantages of these algorithms is their weakness to recognize clusters with variant densities. In general, these algorithms use a global neighborhood radius to determine whether two points are in the same cluster or not. Therefore, they might merge two adjacency clusters with different densities. If a smaller value is chosen, then sparser clusters will be over clustered. Besides, this method is also limited in its ability to represent the data accurately because choosing an appropriate threshold is nontrivial.

Recently, a new clustering algorithm (in this paper, we name it *DensityClust* for simplicity) has been introduced [13] and it clusters data by fast search and then finds density peaks. DensityClust algorithm assumes that the cluster centers are characterized by a higher density than their neighbors and by a relatively large distance from points with higher densities. It recognizes the clusters based on the similarity distance, and detects distinct non-spherical clusters in the datasets. Furthermore, it identifies the number of clusters in the decision graph intuitively through an interactive process. The algorithm has two effective criteria: the local density ρ and the minimum distance δ. Based on the data distribution assumption, only density peaks have large δ and relatively high ρ, so the density peaks can pop out and be separable from the remaining points in the decision graph.

Although DensityClust algorithm is simple and can solve the problems in previous methods to some extend, the characteristics of real-world data are more unpredictable and complicated, and the Euclidean distance measure may not fully reflect the complex spatial distribution of the datasets. The local density selection is still quite sensitive to parameter settings. In addition, facing the less-than-ideal decision graphs, users have difficulty in making decision on the

SAC 2016,April 04-08, 2016, Pisa, Italy

http://dx.doi.org/10.1145/2851613.2851644

Table 1: Main difference of clustering algorithms: A(Excellent), B(Good), C(Fair), D(Poor), F(Failure).

Clustering algorithm	non-predetermined clusters	varying density	arbitrary shapes	multi-scale	noise sensitivity	high-dimensional	algorithmic efficiency
K-means [8]	F	F	D	D	F	D	C
Affinity propagation [6]	A	F	D	D	F	D	C
DBSCAN [5]	A	D	A	D	A	D	C
Spectral clustering [16]	F	D	B	C	F	A	D
DensityClust [13]	F	D	C	D	B	D	A
RDensityClust	A	A	A	A	A	B	C

number of clusters since the decision graphs do not provide any other useful information of the original datasets. To solve those problems, in this paper we propose a robust density-based clustering (**RDensityClust**) for multi-manifold structure. First, we design a manifold distance as the similarity measure which reflects the inherent manifold structure information effectively. Second, we define a new local density calculation based on the topology structure of manifold. It is less sensitive to parameter settings and can represent local density better. Finally, in order to select the cluster centers automatically, we describe a two-phase exemplar determination method. In order to demonstrate an advantage of our algorithm, we compare performance results to several state-of-the-art clustering techniques. Table 1 presents main characteristics of applied algorithms. It can be observed that our approach outperforms in dealing with the data with varying density, multi-scale and noise overlap characteristics.

To summarize, the main contributions of this work include:

- We present a new manifold distance which connects two points in the same manifold by shorter edges, while connect two points in different manifolds by longer edges. Our innovation is that by marking the small components in the graphs, it is able to eliminate the the effects of outliers in noisy case. As a result, the proposed manifold distance is very robust against the noise and outliers.
- We propose a new local density definition based on the proposed manifold distance. It represents the local density better as well as reflects the global consistency of the datasets. This algorithm is more accurate than the original algorithm, and simultaneously it is very stable to the selection of parameters.
- We propose a novel two-phase exemplar determination method in order to select the cluster centers automatically. The approach can reject the real outliers effectively and determine the final cluster centers accurately and automatically.

The rest of the paper is organized as follows. In Section 2, we present a brief overview of the basic DensityClust algorithm. In Section 3, we introduce the robust density-based clustering algorithm in detail. Experimental results on several synthetic and real-world datasets are presented in Section 4, and Section 5 concludes the paper.

2. DENSITYCLUST ALGORITHM

DensityClust algorithm assumes that the cluster centers are characterized by a higher density than their neighbors and by a relatively large distance from points with higher densities. There are two leading criteria in this method: local density(ρ_i) of each point i and minimum distance (δ_i) from other points with higher density, ρ_i is defined as:

$$\rho_i = \sum_j \chi\left(d_{ij} - d_c\right) \ , \tag{1}$$

where ρ_i is the local density, in which,

$$\chi(x) = \begin{cases} 1 & x \leq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

and d_c is a cutoff distance. ρ_i is basically equal to the number of points that are closer than d_c to point i. It should be noticed that this algorithm is sensitive to the relative magnitude of ρ in different points. For large datasets, this algorithm gives an empirical value selection of d_c which makes the average number of neighbors is around 1% to 2% of the total number in the dataset, but empirical analysis shows that the results are less robust with respect to the choice of d_c. δ_i is measured by computing the minimum distance between the point i and any other point with higher density: $\min_{j:\rho_j>\rho_i}\left(d_{ij}\right)$ and for the point with highest density, the paper defines it as: $\max_j\left(d_{ij}\right)$. For more details about this algorithm, reader can refer to [13].

3. PROPOSED ROBUST DENSITYCLUST (RDENSITYCLUST) ALGORITHM

Since the local density selection of DensityClust algorithm relies on the cutoff threshold d_c, it is quite sensitive to the parameter settings. If d_c is set too large, the overlapping neighborhood can even include the data from other clusters, and it may cause that each point has similar heavy density neighborhood; if d_c is set too small, every point has similar sparse density neighborhood. In both cases, the density has less significant discriminative power. DensityClust uses a heuristic value which makes the average number of neighbors is around 1% to 2% of the total number in the dataset. However, the local density (ρ) computation may be sensitive to the parameter d_c and it is not clear how to choose that sensitive parameter without human intervention. Besides, the selection of d_c only depends on the Euclidean distance which only reflects the local information of the datasets, and it is also a major drawback of the d_c selection.

Thus, in this paper we combine the global structure and local information of the dataset and propose the manifold distance which is based on the global consistency [15]. Meanwhile, instead of counting the number of neighbors within a hard cutoff distance threshold d_c, ρ is re-defined as a new cohesion which is inversely proportional to the mean manifold distance to the nearest M neighbors.

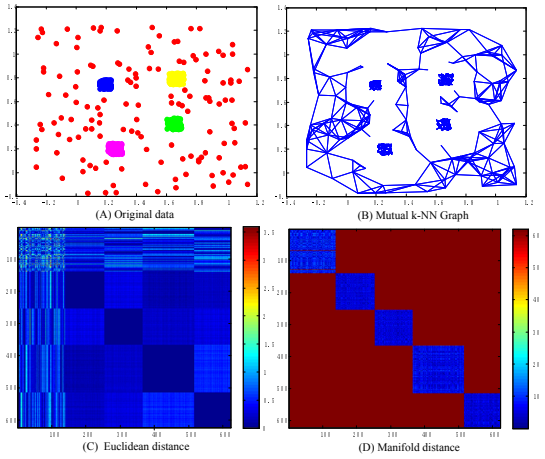

Figure 1: Mutual kNN graph and distance matrix. (A) Original data distribution. (B) Mutual k-NN graph. (C) Euclidean distance. (D) Manifold distance.

3.1 Manifold distance based on global consistency

In this section, the novel manifold distance measure with the ability of reflecting both local and global consistency [15] is introduced and the calculation steps of manifold distance is given as follows:

Step 1. Search all mutual k-nearest neighbors [10] of data points to construct an undirected weighted graph $G = (V, E)$, where V is vertex set and E is edge set. If the point x_j is the mutual k-nearest neighbor of point x_i, the two points are connected, and otherwise disconnected.

Step 2. This graph G can be different depending on whether the dataset contains noise or not. For the noisy case, it should identify and eliminate 'outliers' before calculating the manifold distance [12]. The outliers in the graph are supposed to be the connected components of the graph which are too small and contain less than $\zeta * N$ vertices (N is the total number of data point, ζ is a small value determined later). Then we mark them as the outliers and remove them from the original dataset.

Step 3. Calculate the shortest path $d_{md}(i, j)$ built by Floyd-Warshall algorithm [3] in the adjacency graph G. So manifold distance is defined as:

$$d_{md}(i,j) = \begin{cases} dist(x_i, x_j), & \text{if } x_j \in V_k(x_i) \& x_i \in V_k(x_j) \\ \min_{p \in P_{ij}} \sum_{k=1}^{|p|-1} dist(x_k, x_{k+1}), & \text{otherwise} \end{cases} \quad (3)$$

where P_{ij} the set of all paths connecting vertices x_i and x_j on graph G, and $dist(x_i, x_j)$ is the Euclidean distance between x_i and x_j.

Figure 1 illustrates mutual kNN graph and distance matrix for a synthetic dataset, Figure 1(C) and Figure 1(D) show similarity matrix respectively using the Euclidean distance and manifold distance. Obviously, manifold distance matrix has a well block-diagonal structure which has a better similarity measure between data points. It is because that the manifold distance reflects the intrinsic manifold structure which connects two points in the same manifold by shorter edges, while connects the points in different manifolds by longer edges. Thus it not only achieves the purpose of amplifying the distance between different manifolds but also shortens the distance in the same manifold.

3.2 Novel density peak calculation

Based on the manifold distance, the density peak ρ for each point is redefined according to the global consistency. We first define the density factor $mPer$ to be a percentage of the total number of points, and the quantity of the nearest M neighbors of each point should be $mPer * N$, where N is the total number of points. Then the local density ρ_i of point i is redefined as:

$$d_{md\_av}(i) = \frac{\sum_j d_{md}(i,j)}{M}, \quad j \in V_M(i) \quad (4)$$

$$\rho(i) = \frac{1}{d_{md\_av}(i)}, \quad (5)$$

where $V_M(i)$ is the nearest M neighbors of point i, so the new definition of ρ should be understood as a cohesion which is inversely proportional to the average manifold distance to the nearest M neighbors. The shorter the average distance is, the higher the local density is, and vice versa. The experiment in Section 4 shows the improvement of accuracy is not sensitive to the selection of density factor $mPer$.

3.3 Two-phase exemplar determination

Because the decision graph provides a visual distribution layout(a scatterplot of δ and ρ for all points), the detection process of finding density peaks are performed by user interactive operation. The DensityClust algorithm requires selecting cluster centers(exemplars) manually through an intuitive interface which puts a certain advantage over alternative approaches.

Our target is to find points that have both large ρ and high δ which are considered as cluster centers. Therefore, the two points: ρ and δ have to meet a condition of cluster centers. To solve this problem, we propose a two-phase exemplar determination method (TED). Firstly, the method calculates the mean and standard deviation of ρ and δ, and then it chooses the points that the values meet the PauTa criterion [18] as potential cluster centers. This is a coarse screening stage, and then it utilizes density-peak product γ ($\gamma = \rho * \delta$) which is sorted in decreasing order. The γ fitting curve follows a power-law distributions [4], and it can be observed that the γ value will be a significant jump when the transition from the cluster center to the non-cluster center occurs. In this phase, we choose the points that the gradient decline fastest as turning points to determine the clustering centers accurately. Finally, the final cluster centers are obtained based on the two phases. This technique is accurate and simple to select cluster centers without human intervention.

3.3.1 *Coarse screening stage using improved PauTa criterion*

PauTa criterion is the most commonly used criterion to deal with the abnormal value. For a set of data $\{x_1, x_2, ...x_n\}$, we

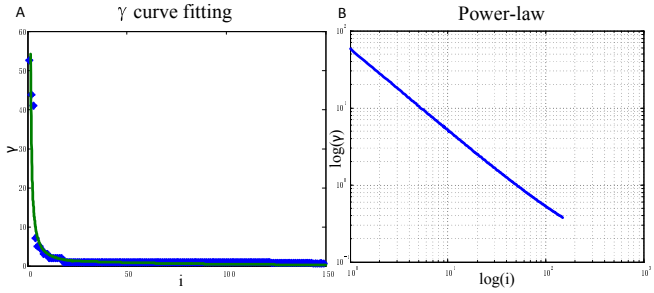

Figure 2: The power-law fitting and the log-log plot for density-peak product γ. (A) Power-law fitting of γ. (B) The log-log plot for power-laws, the slope corresponds to the power law exponent.

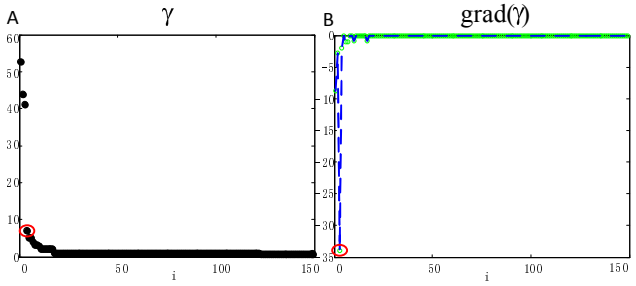

Figure 3: The density-peak product γ and the gradient of γ curve. (A) The density-peak product γ. (B) The gradient of γ curve.

calculate the mean $\overline{X}$ and standard deviation s, and $|v_i| = |x_i - \overline{X}|$ is defined as a residual error.

If x_i satisfies $|v_i| > \theta * s$ (θ usually is set to be 3), x_i would be considered as the suspicious data with abnormal value. This is the Pauta criterion of measurement error theory. In coarse screening stage, we only need to select the points which have anomalously large ρ and δ as potential cluster centers. Take δ as the example, this method calculates its mean value δ_{mean} and the standard deviation s_δ.

According to PauTa criterion, δ value of a potential center shall satisfy

$$\delta(i) - \delta_{mean} > \theta * s_\delta \tag{6}$$

Similarly, the local density ρ should satisfy

$$\rho(i) - \rho_{mean} > \theta * s_\rho \tag{7}$$

Then we initialize screening ρ and δ value based on the Pauta criterion, and it lays very good foundation for further selection of the cluster centers.

3.3.2 *Fine screening stage based on power-law*

In this stage, we calculate the γ value for each data point and sort γ in decreasing order. The quantity of $\gamma(i)$ is distributed according to a power-law distribution, and the exponent of power-law depends on the intrinsic dimension of the dataset [13]. For Iris dataset, the density-peak product γ and power-law fitting is shown in Figure 2. It shows the power-law fitting is well and the straight line on a log-log plot is a strong evidence for power-laws which the slope of the straight line corresponds to the power law exponent. This observation provides the basis criterion for the automatic selection of the cluster centers.

Concerning the issue of finding the cluster centers automatically by using the γ diagram, we find that the quantities of γ of cluster centers are anomalously larger than other points. Therefore, we make use of the gradient of γ for each point, and choose the point which the gradient declines fastest as the turning point. As a result, the points with larger γ than the turning point are treated as the potential cluster centers. According to gradient formula, the gradient of each point is calculated as follows:

$$grad(i) = \frac{\gamma(i+1) - \gamma(i)}{T(i+1) - T(i)}, \tag{8}$$

where $T(i)$ means the sorted order of point i, and $\gamma(i)$ is the density-peak product of point i. The gradient is calculated according to the γ value of each point and the turning point is marked in the Figure 3. It is clear that the γ value of non-cluster centers are smooth low, and the γ value will be a significant jump when the transition from the cluster center to the non-cluster center occurs.

The final cluster centers are obtained by the intersection of potential cluster centers in two stages. In this way, the method is able to reject the fake clusters effectively and determine the final cluster centers automatically.

3.4 Algorithm description

The description of our proposed RDensityClust is shown in Algorithm 1. This algorithm is an extension of DensityClust algorithm. Using this framework, it is possible to discover clusters in the datasets with multi-manifold structure and also to determine a number of clusters in the data automatically. Also note that the proposed manifold distance is very robust to the noise and outliers.

Algorithm 1 Framework of RDensityClust algorithm.

Input:
Dataset: $X = \{x_i, x_2, ..., x_n\}$;
Average density factor: $mPer$;
The k-nearest neighbors: k;
Outlier factor: ζ

Output:
The final clustering index cl;

1: construct an undirected weighted graph $G = (V, E)$ using mutual k-nearest neighbors of data points;
2: outliers removal from the graph using outlier factor ζ;
3: calculate the manifold distance according to Formula (3);
4: calculate ρ for point i according to Formula (4);(5);
5: calculate δ and plot the decision graph of the datasets;
6: use the two-phase exemplar determination method to choose the points with high ρ and high δ as the cluster centers ;
7: assign other remaining points into the same cluster with nearest neighbor of higher density;
8: **return** The final clustering index and the clustering evaluation.

4. EXPERIMENTS

In order to validate the performance of the proposed RDensityClust algorithm, we compare RDensityClust with the o-

riginal DensityClust algorithm, DBSCAN [5], AP (Affinitiy Propagation [6]) algorithm and Self_tuning SC (Self-tuning Spectral Clustering [17]). We test them in two different types of datasets: synthetic datasets and real-world datasets including UCI standard datasets and handwritten digit recognition datasets. The experiments employ the Fowlkes-Mallows, Silhouette and clustering accuracy to evaluate the clustering results.

4.1 Experimental environment

The experimental computer environment: Processor Core 1.9GHz, Memory 4GB, HDD 500G, Windows 7 Ultimate Edition, and the programming language is MATLAB 2015b.

Experiment parameters: In RDensityClust algorithm, $mPer$ is set to be 0.02, k should be chosen in the order of $O(\log n)$ according to [12], and for the noisy case, ζ is set to be 0.02 by default. In DensityClust algorithm: d_c is equal to 1.5% of the total number. AP algorithm [6] uses the default settings: $maxits = 1000$, $conv = 50$, and $damping\_factor = 0.90$. DBSCAN set $MinPts = 4$, and Eps depends on the $4-dist$ value of the threshold point [5]. The kernel width δ in Self_tuning SC algorithm are obtained by using local scaling strategy [17].

Three datasets are utilized in the experiments including synthetic and real-world datasets:

- **Synthetic datasets**. To examine the ability of the proposed algorithm in finding the clustering structure visually, 5 synthetic challenging datasets [1] and the Chameleon dataset are used. They have different manifold structure and can be used to examine the clustering performance on different structural data.
- **UCI benchmark datasets**. 6 standard datasets from the UCI machine learning databases [2] are employed to further validate the superiority of the proposed algorithm. Datasets are described in Table 2.
- **Handwritten digit recognition**. The handwritten digit recognition datasets include the $MINIST$ [9] and USPS [11]. The important statistics of these datasets are summarized as: (1) The MNIST database has a training set of $60,000$ examples, and a test set of $10,000$ examples. The digits have been size-normalized and centered in a fixed-size image as 28×28 dimensional vector. (2) The USPS database obtained from the scanning of handwritten digits from envelopes by the U.S. Postal Service. There are 7291 training observations and 2007 test observations, each image here has been size normalized, resulting in 16×16 grayscale images.

4.2 Evaluation metrics

The experiment takes Fowlkes-Mallows and Silhouette metrics to evaluate the clustering results. The Fowlkes-Mallows metric is an external evaluation method defined as: $FMI = \sqrt{\frac{TP}{TP+FP} * \frac{TP}{TP+FN}}$, where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives. The greater value Fowlkes-Mallows is, the more accurate the algorithm is.

Table 2: Attributes of UCI standard datasets.

Datasets	samples	classes	dimensions
Iris	150	3	4
Wine	178	3	13
Ionosphere	351	2	34
Glass	214	7	10
Wdbc	569	2	30
Image-segment	2310	7	19

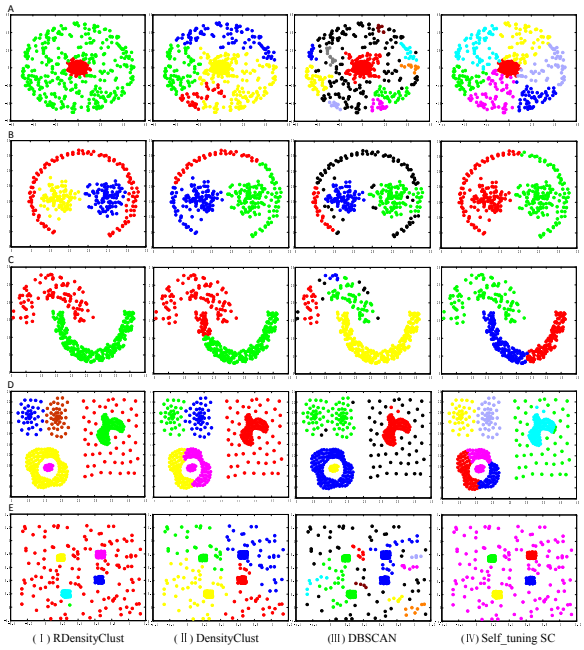

Figure 4: The clustering results for synthetic datasets, different clusters are marked in different colours.

Silhouette provides a succinct graphical representation of how well each point lies within its cluster, it is defined as $Sil(i) = \frac{b(i)-a(i)}{max\{a(i),b(i)\}}$, $a(i)$ is the average dissimilarity of the point i with all other points within the same cluster. $b(i)$ be the lowest average dissimilarity of the point i to any other cluster which point i is not a member. The average Sil_{avg} over all data of a cluster is a measure of how tightly grouped all the data in the cluster are, Thus Sil_{avg} close to one means that the datum is appropriately clustered.

4.3 Experimental results and analysis

4.3.1 *Synthetic datasets*

Figure 4 shows the clustering results on the 5 datasets using four algorithms which can identify the manifold structure. The clustering results illustrate that the proposed R-DensityClust algorithm can accomplish the clustering for the datasets successfully, and the performance in handling multi-scale and the overlapping data is significantly better than Self-tuning SC and DBSCAN algorithm. It is observed that DBSCAN algorithm does not deal efficiently with clusters of varying densities and multi-scale character. It is also

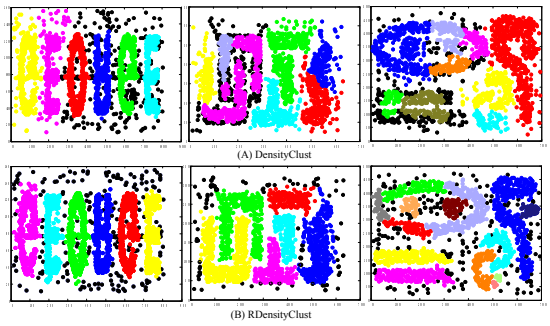

Figure 5: The results on Chameleon datasets.

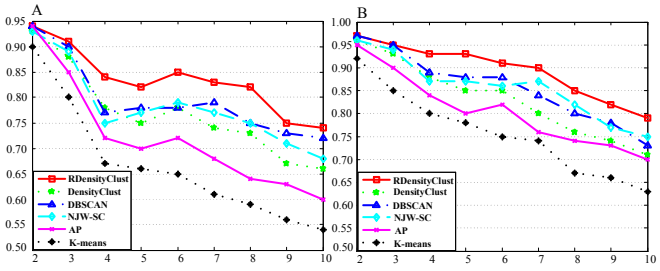

Figure 6: Clustering accuracy vs. the digital category. (A) The MNIST database. (B) The USPS database.

sensitive to the selection of Eps and $MinPts$. For spectral clustering, even though the selection of parameter adopts the self-tuning method, the result is still not satisfactory for handling multi-scale and background clutter data. Also the clustering result appears instability performance, this is because spectral algorithm uses the simple K-means processing method as afterwards data processing and K-means is extremely sensitive to the selection of the initial cluster centers. RDensityClust is relatively accurate in data clustering for different manifolds and overcomes Euclidean distance which can not reflect the global consistency. Meanwhile, the new local density calculation can better represent the local structure information, thereby improve the accuracy.

In addition, we test the multi-manifold datasets under noisy environment. As the Figure 5 shows, the experiments are conducted on the Chameleon dataset that is used to evaluate the capability to correctly identify clusters in noisy environments. We can see that RDensityClust identifies the noise effectively and get effective clusters in the dataset.

4.3.2 *UCI benchmark datasets*

Table 3 shows the Fowlkes-Mallows and Silhouette metrics of the 10 times random experiments on the UCI datasets. As the results show, for Self_tuning SC algorithm, even though the self-tuning method is used for the selection of parameter, the performance is still worse than RDensityClust. The reason is that it uses Euclidean distance as the similarity measure which can not reflect the complex structure in the datasets. While the results of DBSCAN is better than AP algorithm and lower than RDensityClust, but the parameters of DBSCAN are very demanding. The clustering performance of RDensityClust algorithm is better than the other four algorithms, this is because the proposed manifold distance captures the local and global structures much better. In addition, the redefinition of rho expresses the concept of local density more accurately.

4.3.3 *Handwritten digit recognition*

The general problem in the handwritten digit clustering is the similarity between the digits like 0 and 6, 3 and 5, 3 and 8, 9 and 8 etc. Also the same handwritten digits are wrote in many different ways. The diversity and uniqueness of different individuals handwriting also affect the formation and appearance of the digits. Therefore, our approach to solve this problem can be divided into two steps: 1) we reduce the dimensions of the dataset using principal component analysis (PCA) [14] which represents the input digit images by projecting them onto a low dimensional space, which is constituted by a small number of basis images. These basis images are derived by finding the most significant eigenvectors of the pixel wise covariance matrix. 2) we compare the performance of these algorithms to cluster the digit in the low dimensional space.

We conduct the following experiments, we use the first 300 images of each digital category as the input. The evaluations are conducted with the digital category increasing from two categoryies ($'0' -' 1'$) to the ten categories ($'0' -' 9'$). Every time adding a new digital category, 10 test runs are conducted on different randomly chosen clusters and the average performance are given. From the Figure 6, we draw the following conclusions:

(1) As expected, the proposed RDensityClust approach outperforms five other methods in terms of accuracy. The algorithms DBSCAN and K-means showed the lowest performance. We demonstrated, that the designed RDensityClust approach fully use the local structure information in the datasets.

(2) Self-tuning SC generally outperforms AP method on the image datasets since there are clear nonlinear intrinsic manifolds in those datasets, and the data structural information is crucial for image clustering, but the repeatedly experiment observes that the Self-tuning SC and K-means appeared instability performance because of the initial selection of the cluster centers.

(3) RDensityClust algorithm outperforms other methods when the number of category increases. It makes the inter-cluster connections relatively weaker and the within-cluster connections relatively stronger.

4.4 Parameter sensitivity analysis

We analyze the parameter sensitivity by tuning the values of the density factor $mPer$ in RDensityClust algorithm. The density factor $mPer$ controls the number of the nearest neighbors. In this experiment, we set $mPer$ increase gradually from 0.01 to $min(1/nClu, 0.1)$ in steps of 1%, where $nClu$ denote the number of the clusters. We compare the Fowlkes-Mallows metric and the results are shown in Figure 7 and it manifests that the clustering quality is very good and stable in the range 1% to 6% if the known clusters

Table 3: Clustering results of UCI datasets.

DATASETS	NJW-SC		AP		DBSCAN		DensityClust		RDensityClust	
	FMI	Sil	FMI	Sil	FMI	Sil	FMI	Sil	FMI	Sil
Iris	0.73	0.49	0.81	0.54	0.76	0.51	0.74	0.52	0.85	0.56
Wine	0.58	0.49	0.40	0.65	0.54	0.58	0.62	0.49	0.77	0.65
Ionosphere	0.62	0.39	0.54	0.29	0.63	0.36	0.60	0.25	0.66	0.40
Glass	0.50	0.31	0.60	0.53	0.59	0.35	0.54	0.52	0.65	0.54
Wdbc	0.76	0.59	0.79	0.70	0.73	0.68	0.75	0.69	0.82	0.69
Im-segment	0.41	0.44	0.41	0.37	0.36	0.44	0.44	0.38	0.60	0.39

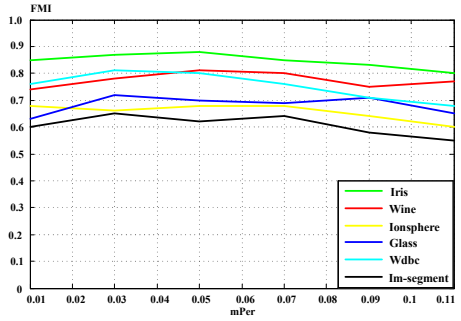

Figure 7: FMI changes with $mPer$.

are fewer than approximately 30. The results demonstrate that the Fowlkes-Mallows metrics are very robust to the average density factor $mPer$. This observation confirms that RDensityClust reflects the inherent manifold structures of datasets and it also verifies that the mean manifold distance to M nearest neighbors be a good approach due to low sensitivity to threshold choice.

5. CONCLUSIONS

In this paper, we proposed the RDensityClust algorithm. This algorithm has several advantages: 1) it used our proposed manifold distance to capture the global consistency; 2) it considered both local and global intrinsic structure information contained in datasets using a new local density calculation, which was based on the proposed manifold distance; 3) it selected the cluster centers automatically through a two-phase exemplar determination method. The experimental results on several synthetic and real-world datasets indicated that the proposed RDensityClust algorithm performed significantly better than the other algorithms in the data with various densities, multi-scale and noisy overlapping, and was very robust to the parameters.

There are a number of possible extensions for future work. Firstly, because of the higher complexity of the manifold distance, it will be our future work to reduce the complexity of manifold distance calculation. Besides, in the real-world pattern recognition problems, the data distributions are usually high-dimensional and in the form of stream. Therefore, an extension of the proposed algorithm to high-dimensional and streaming data is also our future research.

6. REFERENCES

[1] J. Alcalá, A. Fernández, and Luengo. Keel data mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, 17(255-287):11, 2010.

[2] K. Bache and M. Lichman. Uci machine learning repository. *URL http://archive.ics.uci.edu/ml*, 901, 2013.

[3] T. M. Chan. More algorithms for all-pairs shortest paths in weighted graphs. *SIAM Journal on Computing*, 39(5):2075–2089, 2010.

[4] G. C. Crawford, H. Aguinis, B. Lichtenstein, P. Davidsson, and B. McKelvey. Power law distributions in entrepreneurship: Implications for theory and research. *Journal of Business Venturing*, 2015.

[5] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[6] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

[7] D. Gong, X. Zhao, and G. Medioni. Robust multiple manifolds structure learning. *arXiv preprint arXiv:1206.4624*, 2012.

[8] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[9] Y. LeCun and C. Cortes. The mnist database of handwritten digits, 1998. *Available electronically at http://yann. lecun. com/exdb/mnist*, 2012.

[10] T.-t. Li, B. Jiang, Z.-z. Tu, B. Luo, and J. Tang. Image matching using mutual k-nearest neighbor graph. In *Intelligent Computation in Big Data Era*, pages 276–283. Springer, 2015.

[11] C.-L. Liu, K. Nakashima, H. Sako, and H. Fujisawa. Handwritten digit recognition: benchmarking of state-of-the-art techniques. *Pattern Recognition*, 36(10):2271–2285, 2003.

[12] M. Maier, M. Hein, and U. von Luxburg. Optimal construction of k-nearest-neighbor graphs for identifying noisy clusters. *Theoretical Computer Science*, 410(19):1749–1764, 2009.

[13] A. Rodriguez and A. Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014.

[14] J. Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.

[15] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[16] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[17] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in neural information processing systems*, pages 1601–1608, 2004.

[18] H. Zhao, F. Min, and W. Zhu. Test-cost-sensitive attribute reduction of data with normal distribution measurement errors. *Mathematical Problems in Engineering*, 2013, 2013.