A Three-Level Optimization Model for Nonlinearly Separable Clustering

Liang Bai, Jiye Liang*
Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education,
School of Computer and Information Technology, Shanxi University, China
{bailiang,ljy}@sxu.edu.cn

Abstract

Due to the complex structure of the real-world data, nonlinearly separable clustering is one of popular and widely studied clustering problems. Currently, various types of algorithms, such as kernel k-means, spectral clustering and density clustering, have been developed to solve this problem. However, it is difficult for them to balance the efficiency and effectiveness of clustering, which limits their real applications. To get rid of the deficiency, we propose a three-level optimization model for nonlinearly separable clustering which divides the clustering problem into three sub-problems: a linearly separable clustering on the object set, a nonlinearly separable clustering on the cluster set and an ensemble clustering on the partition set. An iterative algorithm is proposed to solve the optimization problem. The proposed algorithm can use low computational cost to effectively recognize nonlinearly separable clusters. The performance of this algorithm has been studied on synthetical and real data sets. Comparisons with other nonlinearly separable clustering algorithms illustrate the efficiency and effectiveness of the proposed algorithm.

Introduction

Clustering is an important problem in statistical multivariate analysis, data mining and machine learning (Jain 2008). The goal of clustering is to group a set of objects into clusters so that the objects in the same cluster are highly similar but remarkably dissimilar with objects in other clusters. To tackle this problem, various types of clustering algorithms have been developed in the literature (e.g., (Aggarwal and Reddy 2014) and references therein). According to the data distribution of clusters, data clustering problem can be categorized into two types: linearly separable clustering and nonlinearly separable clustering (Wang and Lai 2016). The k-means (MacQueen 1967) algorithm is a representative of linearly separable clustering algorithms. It is well-known for its low computational cost. However, it can not recognize nonlinearly separable clusters. Due to the complex structure of the real-world data, the data set to be partitioned may contain at least one cluster with concave boundaries or even of arbitrary shapes. Therefore, nonlinearly separable clustering is one of most popular and widely studied clustering problems.

Currently, several types of nonlinearly separable clustering approaches have been developed, such as connectivity-based, density-based, kernel-based, and multi-centers approaches. Connectivity-based approaches connect some nearby objects to form clusters based on their distance. The representative methods include different linkage algorithms (Witten and Frank 2005). Density-based approaches mainly use the density-connectivity between objects to recognize different shaped clusters. The representative methods include DBSCAN (Ester et al. 1996) and DP (Rodriguez and Laio 2014). Compared to the connectivity-based approaches, the density-based approaches fully consider noise and border points. The kernel-based approaches use a kernel function, which is an appropriate nonlinear mapping from the original (input) space to a higher dimensional feature space, to extract non-linearly separable clusters. The representative methods include kernel k-means (Scholkopf et al. 1998), spectral clustering (Shi and Malik 2000; Ng, Jordan, and Weiss 2001) and mean-shift (Cheng 1995). Unlike the k-means algorithm, the above approaches need use pairwise similarities of objects to determine the membership of each object to clusters. Since they do not select a center but use all the objects to represent a cluster, they can recognize nonlinearly separable clusters. However, they need expensive time or space costs, e.g., computing and operating the similarity matrix, which are not suitable for large-scale data sets. The multi-centers approaches (Liu, Jiang, and Kot 2009; Liang et al. 2012; Wang et al. 2013) use multiple centers to represent a nonlinearly separable cluster. They are in between cluster representations of single center and all the objects. The advantage of replacing data objects with multiple centers to describe a cluster is to avoid the over fitting of the clustering result. The disadvantage is that the performance of their clustering is very sensitive to the center selection. However, obtaining high-quality centers usually needs high-computational cost.

Currently, various types of accelerating approaches have been proposed to enhance the scalability of nonlinearly-separable clustering algorithms, such as index-based clus-

*Jiye Liang is the corresponding author: ljy@sxu.edu.cn

tering, hybrid clustering, sampling-based clustering and parallel clustering. Index-based clustering uses one of spatial index structure, such as KD-tree (Bentley 1975), R*-tree (Beckmann et al. 1990), or X-tree (Berchtold, Keim, and Kriegel 1996), to reduce the cost of computing the pairwise distance matrix in the clustering process. However, these indices are only suitable for data sets with very lower dimensions, since the computational complexity of this solution increases exponentially with the data dimensions. Hybrid clustering integrates the k-means algorithm and nonlinearly separable clustering algorithms to enhance the clustering efficiency. The representative methods include DBSCAN with k-means (Dash, Liu, and Xu 2001; Viswanath and Pinkesh 2006; Nanda and Panda 2015), linkage with k-means (Liu, Jiang, and Kot 2009), the spectral clustering with k-means (Yan, Huang, and Jordan 2009), and the multilevel clustering algorithm (Dhillon, Guan, and Kulis 2007) which uses the iterative optimization method of kernel k-means to solve the spectral clustering problem. Sampling-based clustering reduces the scale of the data set by sample techniques and implements a nonlinearly separable clustering algorithm on the set of samples, instead of the data set. For example, Williams et al. (Williams and Seeger 2001) used the Nystroem method to speed up the kernel approximation. Rahimi et al. (Rahimi and Recht 2007) approximated the feature map of the kernel by random projection. Chen et al. (Chen and Cai 2011) used a relation matrix between samples and objects, instead of the pairwise matrix, and the eigenvalue decomposition, instead of the singular value decomposition in spectral clustering. Mohan et al. (Mohan and Monteleoni 2017) integrate uniform sampling and weighted kernel k-means to speed the solution of the spectral clustering. Huang et al. (Huang et al.) proposed a representative selection method to enhance the performance of sampling-based clustering. Parallel clustering uses parallel techniques to enhance the clustering speed of the original algorithms. For example, some scholars have proposed parallel density-based clustering algorithms using the MapReduce technique (Dean and Ghemawat 2008), such as MR-DBSCAN (He, Tan, and et al. 2011), DBCURE-MR (Kim et al. 2014), and parallel spectral clustering (Chen et al. 2011). This type of algorithms needs additional platform for high performance computing to deal with large-scale data sets.

Despite the theoretical and practical advantages of the above-mentioned techniques, it is very difficult for them to balance the efficiency and effectiveness of nonlinearly separable clustering. While these approaches are lowering the computational cost, the robustness and quality of clustering results are often sacrificed. In order to solve this problem, we propose a three-level optimization model for nonlinearly separable clustering. In the new model, a nonlinearly separable clustering problem is divided into three sub-problems: the linearly separable clustering on the object set, the nonlinearly separable clustering on the cluster set and the ensemble clustering on the partition set. Cluster ensemble is one of important techniques for cluster analysis, which is used to solve the robustness problem of clustering results (Zhou 2012). Currently, lots of cluster ensemble algorithms have been developed, seen in (Yu et al. 2017). It is noted that the new model is different from traditional cluster ensemble algorithms whose objective is to get the most consensus of base clusterings. However, our objective is to solve the nonlinearly separable clustering problem. Therefore, our objective function includes three validity functions for linearly separable clustering, nonlinearly separable clustering and ensemble clustering. These validity functions are defined based on the k-means objective function. We propose an k-means-like iterative method to minimize the objective function. The k-means (MacQueen 1967), kernel k-means (Scholkopf et al. 1998) and k-means-based cluster ensemble (Wu et al. 2015) algorithms can be seen as the special case of the proposed algorithm. Compared to existing clustering algorithms, the proposed algorithm can well balance the computation cost and quality of nonlinearly separable clustering.

The outline of the rest of this paper is as follows. Section 2 presents a new optimization model for nonlinearly separable clustering. Section 3 demonstrates the performance of the proposed algorithm. Section 4 concludes the paper with some remarks.

The three-level optimization model

In this section, we propose a three-level clustering model for nonlinearly separable clustering. Fig.1 shows the clustering procedure of the proposed model. Generally speaking, different clusters on a complex data set are nonlinearly separable in the global geometric space but linearly separable in the local geometric space. Thus, we assume that a nonlinearly separable cluster is made up of several small linearly separable clusters. Based on this assumption, we divide a nonlinear clustering problem into three subproblems, i.e., linearly separable clustering, nonlinearly separable clustering and ensemble clustering. Let X be a $n \times m$ data matrix with n objects and m features. We implement the linear clustering on X, which can produce a $n \times p$ partition matrix W of objects, where p is the number of clusters in W which is required to be more than the number of real clusters k. The task of the linear clustering is to make objects similar with each other in the local geometric space into the same clusters. Furthermore, we see each cluster in W as a linear cluster and implement the nonlinear clustering to partition linear clusters into k nonlinear clusters, which can produce a $p \times k$ partition matrix H of linear clusters. Let U be a $n \times k$ desired partition matrix of objects. We hope to integrate W and H to approximate the partition matrix U, i.e., $WH \approx U$. However, the performance of the approximated result depends on the quality of W and H. Therefore, we try to produce T approximated results and integrate them to estimate U, where T is no less than 1.

The objective function and optimization problem

On the basis of the above motivation, we propose a three-level optimization model, where its objective function is made up of three validity functions: linearly separable clustering on the data set, nonlinearly separable clustering on the cluster set and ensemble clustering on the partition set. Based on the objective function of the k-means algorithm,

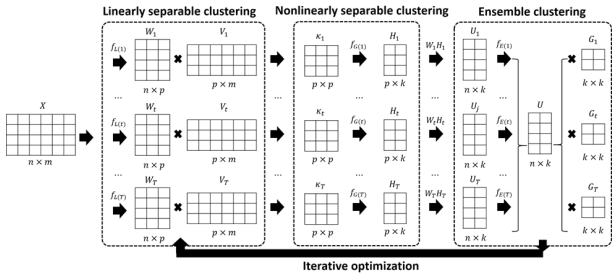


Figure 1: Three-level clustering procedure

we define them as follows.

$$f_{L(t)} = ||X - W_t V_t||_F^2 \tag{1}$$

is a validity function of the tth linear clustering on the data set X, for $1 \leq t \leq T$. It is the objective function of classical k-means algorithm. $W_t = [w_{t(ij)}]$ is the $n \times p$ partition matrix of the tth linear clustering, where $w_{t(ij)}$ is the membership of the ith object to the jth linear cluster, and V_t is the $p \times m$ center matrix of the tth linear clustering, where $\mathbf{v}_{t(j)}$ is the jth row of V_t representing the cluster center of the the jth linear cluster. If the value of $f_{L(t)}$ is low, the objects in the same linear clusters are very close to each other in the original feature space of X.

$$f_{G(t)} = ||\phi(V_t) - H_t Z_t||_F^2 = Tr(K_t) - Tr(\hat{H}_t^T K_t \hat{H}_t) \tag{2}$$

is a validity function of the tth nonlinear clustering on V_t which are used to represent the linear clusters obtained by the tth linear clustering, for $1 \leq t \leq T$. It is the objective function of the kernel k-means algorithm (Scholkopf et al. 1998). $\phi(V_t)$ and Z_t are the representations of V_t and its cluster centers in the nonlinearly embedded space by a kernel function, respectively. K_t is the $p \times p$ kernel matrix of V_t. H_t is the $p \times k$ partition matrix of V_t and $\hat{H}_t = HD_t^{-1/2}$ is the normalized matrix of H_t, where $D_t = [d_{t(j)}]$ is a $k \times k$ diagonal matrix with $d_{t(j)} = \sum_{i=1}^{p} h_{t(ij)}$. The goal of $f_{G(t)}$ is to minimize the difference of the linear clusters in the same nonlinear clusters. In (Dhillon, Guan, and Kulis 2007), it has been proved that the kernel k-means algorithm is equivalent to the spectral clustering algorithm. Thus, minimizing $f_{G(t)}$ can be solved by the eigenvalue decomposition.

$$f_{E(t)} = ||W_t \hat{H}_t - U G_t||_F^2 \tag{3}$$

is a validity function of the tth ensemble clustering, for $1 \leq t \leq T$. U is a $n \times k$ partition matrix representing the final clustering of the data set. For ease of calculation, $W_t \hat{H}_t$ is used to reflect the tth partition matrix, instead of $W_t H_t$. G_t is a $k \times k$ relation matrix between $W_t \hat{H}_t$ and U. $f_{E(t)}$ is used to evaluate the difference between each $W_t \hat{H}_t$ and U. In this case, $\sum_{t=1}^{T} f_{E(t)}$ is seen as a consensus function for cluster ensemble. We hope to minimize it to obtain the final clustering U which is the most consensus with all $W_t \hat{H}_t$ $(1 \leq t \leq T)$. If $W_t \hat{H}_t$ is fixed, $\sum_{t=1}^{T} f_{E(t)}$ is the objective function of the classical k-means algorithm. In (Wu et al. 2015), Wu and Liu et al. have proved that it can be used in cluster ensemble.

Thus, the new optimization model is formally defined as follows.

$$\min_{U} F = \sum_{t=1}^{T} \alpha f_{L(t)} + \beta f_{G(t)} + \gamma f_{E(t)}, \tag{4}$$

subject to

$$\begin{cases} w_{t(ij)} \in \{0,1\}, \sum_{j=1}^{p} w_{t(ij)} = 1, 1 < \sum_{i=1}^{n} w_{t(ij)} < n, \\ h_{t(ij)} \in \{0,1\}, \sum_{j=1}^{k} h_{t(ij)} = 1, 1 < \sum_{i=1}^{p} h_{t(ij)} < p, \\ u_{ij} \in \{0,1\}, \sum_{j=1}^{k} u_{ij} = 1, 1 < \sum_{i=1}^{n} u_{ij} < n, \end{cases} \tag{5}$$

where α, β and γ are the weights which are used to balance the importance of each term.

Solution algorithm for the optimization problem

Let $W = \{W_t\}_{t=1}^T$, $V = \{V_t\}_{t=1}^T$, $H = \{H_t\}_{t=1}^T$, $Q = \{Q_t\}_{t=1}^T$ and $G = \{G_t\}_{t=1}^T$. In order to solve the optimization problem (4), we divide it into three minimization subproblems, i.e., *Problem* P_1: Fix H, U and G, solve $\min_{W,V} F$; *Problem* P_2: Fix W, V, U and G, solve $\min_H F$; *Problem* P_3: Fix W, V and H, solve $\min_{U,G} F$. We use a k-means-like paradigm to approximately obtain its optimal solution.

Solution for Problem P_1: Given H, U and G, $\alpha f_{L(t)} + \beta f_{G(t)} + \gamma f_{E(t)}$ is independent to each other and $f_{G(t)}$ is constant, for $1 \leq t \leq T$. In this case, the minimization problem P_1 becomes

$$\min_{W_t, V_t} \alpha f_{L(t)} + \gamma f_{E(t)}, 1 \leq t \leq T. \tag{6}$$

In order to solve the problem, we have

$$\alpha f_{L(t)} + \gamma f_{E(t)} = ||X_t' - W_t V_t'||_F^2, \tag{7}$$

where $X_t' = [\alpha^{1/2} X, \gamma^{1/2} U G_t]$ is a concatenated matrix with X and UG_t, and $V_t' = [\alpha^{1/2} V_t, \gamma^{1/2} \hat{H}_t]$ is a concatenated matrix with V_t and H_t. While UG_t is seen as the new features of the original data X, X_t' is a new representation of the original data X. V_t' is seen as the cluster center matrix of X_t'.

According to Eq.(7), we transform the optimization problem into a k-means clustering problem of X_t', i.e., $\min_{W_t, V_t'} ||X_t' - W_t V_t'||_F^2$. Thus, we solve the problem by the update formula W_t and V_t' which are described as follows. Given V_t', the minimizer W_t is given by

$$w_{t(ij)} = \begin{cases} 1, & j = \arg\min_{l=1}^{p} ||\mathbf{x}_i' - \mathbf{v}_{t(l)}'||^2, \\ 0, & \textit{otherwise.} \end{cases} \tag{8}$$

for $1 \leq i \leq N$, $1 \leq j \leq P$, $1 \leq t \leq T$, where $\mathbf{x}_i'$ is the ith row of X_t' and $\mathbf{v}_{t(l)}'$ is the lth row of V_t'. Given W_t, the minimizer V_t' is given by

$$\mathbf{v}_{t(j)}' = \sum_{i=1}^{N} w_{t(ij)} \mathbf{x}_i' \Big/ \sum_{i=1}^{N} w_{t(ij)}\ , \tag{9}$$

for $1 \leq j \leq P$, $1 \leq t \leq T$. Therefore, the computational complexity of solving Problem P_1 is $O(n(m+k)poT)$ where o is the number of iterations for a linear clustering.

Solution for Problem P_2: Given W, V, U and G, $\alpha f_{L(t)} + \beta f_{G(t)} + \gamma f_{E(t)}$ is also independent to each other and $f_{L(t)}$ is constant, for $1 \leq t \leq T$. In this case, the minimization problem P_2 becomes

$$\min_{H_t} \beta f_{G(t)} + \gamma f_{E(t)}, 1 \leq t \leq T. \tag{10}$$

We have

$$f_{E(t)} = Tr(\hat{H}_t^T W_t^T W_t \hat{H}_t) - Tr(\hat{H}_t^T W_t^T \hat{U}\hat{U}^T W_t \hat{H}_t). \tag{11}$$

In this case,

$$\beta f_{G(t)} + \gamma f_{E(t)} = \beta Tr(K_t) - Tr(\hat{H}_t^T L_t \hat{H}_t) \tag{12}$$

where $L_t = \beta K_t + \gamma W_t^T (I_n - \hat{U}\hat{U}^T) W_t$. Minimizing $\beta f_{G(t)} + \gamma f_{E(t)}$ is equivalent to solving the problem as follows.

$$\max_{\hat{H}_t} Tr(\hat{H}_t^T L_t \hat{H}_t), \text{ subject to } \hat{H}_t^T \hat{H}_t = I_k. \tag{13}$$

Hence, we can solve Problem P_2 by the spectral method. The computational complexity of solving Problem P_2 is $O((k+m)p^2T)$.

Solution for Problem P_3: Given V, W and H, each $\alpha f_{L(t)} + \beta f_{G(t)}$ is constant, for $1 \leq t \leq T$. Therefore, the minimization problem P_3 becomes

$$\min \sum_{t=1}^{T} ||W_t \hat{H}_t - U G_t||_F^2.$$

We use the k-means-like paradigm to solve the problem. We provide the update formulas for G and U as follows. Given W, V, H and U, the minimizer G is given by

$$\mathbf{g}_{t(l)} = \sum_{i=1}^{n} u_{il} \mathbf{q}_{t(i)} \Big/ \sum_{i=1}^{n} u_{il} \ , \tag{14}$$

where $\mathbf{q}_{t(i)}$ is the ith row of $W_t \hat{H}_t$, for $1 \leq l \leq k$, $1 \leq t \leq T$. Given W, V, H and G, the minimizer U is given by

$$u_{il} = \begin{cases} 1, & l = \arg\min_{h=1}^{k} \sum_{t=1}^{T} ||\mathbf{q}_{t(i)} - \mathbf{g}_{t(h)}||^2, \\ 0, & \textit{otherwise,} \end{cases} \tag{15}$$

for $1 \leq i \leq n$, $1 \leq l \leq k$. Therefore, the computational complexity of solving Problem P_3 is $O(nk^2oT)$.

Based on the above update formulas, we approximately minimize the objective function F by iteratively solving the subproblems. Before the first iteration, we need to initialize the variables V, W, H, U and G. We first randomly select p objects from X to initialize V_t, for $1 \leq t \leq T$. Given V_t, we can compute W_t by the k-means algorithm. Furthermore, based on V_t, we can obtain K_t and H_t by the spectral clustering algorithm. Given V, W, and H, we use

$$\theta_t = \left(f_{L(t)} - \min_{t=1}^{T} f_{L(t)} \Big/ \max_{t=1}^{T} f_{L(t)} - \min_{t=1}^{T} f_{L(t)} \right) \left(f_{G(t)} - \min_{t=1}^{T} f_{G(t)} \Big/ \max_{t=1}^{T} f_{G(t)} - \min_{t=1}^{T} f_{G(t)} \right)$$

to evaluate the quality of Q_t. We select the partition matrix Q_r which satisfies $r = \arg\min_{t=1}^{T} \theta_t$, to initialize $U = Q_r$. Given U, we can compute G by Eq.(14).

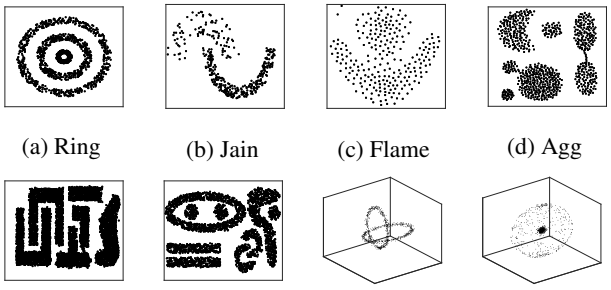
The overall clustering procedure is summarized in Algorithm 1. For the proposed algorithm, the parameters α, β, γ, p, T and τ can be used to control its efficiency and effectiveness. By default, we set $\alpha = \beta = \gamma = 1$ which assumes the three terms of the objective function F are equally important. By adjusting α, β and γ, we can conclude that the classical k-means, kernel k-means and k-means-based cluster ensemble algorithms can be seen as the special case of the proposed algorithm. If $T > 1$, $\beta = 0$, $\beta \neq 0$ and $\gamma \neq 0$, the clustering result can be seen as an integrated result of T linear (k-means) clusterings. If $p = n$, $V_t = X$. In this case, the clustering result can be seen as an integrated result of T nonlinear (kernel k-means) clusterings. If $T > 1$, $\alpha = \beta = 0$ and $\gamma \neq 0$, the proposed algorithm becomes a cluster ensemble algorithm. For the parameter τ, if it is equal to 1, the ensemble result can not be used to improve the generation of each W_t and H_t. The time complexity of the proposed algorithm is $O((n(m+k)po + (k+m)p^2 + nk^2o)T\tau)$, where τ is the maximum or desired number of iterations. According to the time complexity, we can see that the clustering efficiency depends on the parameters p, T and τ. We assume the number of linear clusters p is no more than $\sqrt{n}$ which is the maximum possible number of clusters on a data set (Yu and Cheng 2004). Therefore, the time complexity is no more than $O(kn^{3/2}T\tau)$. For a traditional nonlinear clustering algorithm, its time complexity is generally no less than $O(kn^2)$. If we set $T\tau$ to less than $n^{1/2}$, the proposed algorithm can reduce the computational cost of nonlinearly separable clustering. Besides, the algorithm has good parallelizability. Since each $\alpha f_{L(t)} + \beta f_{G(t)}$ is independent to each other, for $1 \leq t \leq T$, we can minimize them in parallel.

Algorithm 1: The NKM-NSC algorithm

Input: X, k
Output: U
Initialize V, W, H, U and G;
Repeat
Fixed H, U and G, solve Problem P_1 to obtain V and W;
Fixed V, W, U and G, solve Problem P_2 to obtain H;
Fixed V, W and H, solve Problem P_3 to obtain U and G;
Until The objective function is not changed or the desired number of iterations τ is reached.

Experimental analysis

In order to properly examine the performance of the NKM-NSC algorithm, we compare it with other nine nonlinearly separable clustering algorithms which include: DBSCAN (Ester et al. 1996) with KD-Tree (DBSCAN+KD), density peaks clustering (Rodriguez and Laio 2014) with KD-Tree (DP+KD), multi-exemplar affinity propagation



(a) Ring (b) Jain (c) Flame (d) Agg

(e) T4.8k (f) T7.1k (g) Chain (h) Atom

Figure 2: Data distribution of synthetic data

(MEAP) (Wang et al. 2013), multi-prototypes clustering (MP) (Liu, Jiang, and Kot 2009), kernel k-means (KKM) (Scholkopf et al. 1998), spectral clustering (SC) (Ng, Jordan, and Weiss 2001), spectral clustering with Nystroem (SC+N) (Williams and Seeger 2001), spectral clustering with random feature (SC+RF) (Rahimi and Recht 2007), spectral clustering with k-means (SC+KM) (Yan, Huang, and Jordan 2009).

The comparisons are carried out on 8 synthetic and 7 real data sets (Benchmarks). The synthetic data sets include Ring (1,500 objects and 3 clusters), Jain (373 objects and 2 clusters), Flame (240 objects and 2 clusters), Agg (788 objects and 7 clusters), T4.8k (7,235 objects and 6 clusters), T7.1k (3,031 objects and 9 clusters), Chain (1,000 objects and 2 clusters) and Atom (800 objects and 2 clusters). The data distributions of the synthetic data sets are shown in Fig. 2. The real data sets include Wine (178 objects, 13 features and 3 clusters), Breast Cancer (569 objects, 30 features and 2 clusters), Handwritten Digits (5,620 objects, 63 features and 10 clusters), Landsat Satellite (6,435 objects, 36 features and 7 clusters), MNIST (10,000 objects, 784 features and 10 clusters) and KDD-CUP'99 (1,048,576 objects, 39 features and 2 clusters).

The experiments are conducted on an Intel i7-4710MQ personal computer with 16G RAM. We employ the widely-used external indices (Aggarwal and Reddy 2014): the normalized mutual information (NMI) and the adjusted rand index (ARI), to measure the similarity between a clustering result and the true partition on a data set. If the clustering result is close to the true partition, then its NMI and ARI values are high.

Before the comparisons, we need to set the parameters of these algorithms as follows. For each algorithm, we first set the number of clusters k is equal to its true number of classes on each of the given data sets. Furthermore, we use Gaussian kernel function to produce the distance or similarity matrix and test each of these algorithms with different δ values of the kernel parameter, i.e., $\delta = \varepsilon_X$, $\varepsilon_X/10$, $\varepsilon_X/20$, $\varepsilon_X/30$, $\varepsilon_X/40$, $\varepsilon_X/50$ where ε_X is the average distance of data set X, to select the highest ARI and NMI values for comparison. Since the performance of the MP, KKM, SC, SC+N, SC+RF, SC+KM and NKM-NSC algorithms are affected by the selection of initial points, each of them runs 30 times to compute the mean and standard deviation of ARI and NMI on each data set. For the NKM-NSC algorithm, we set $\alpha = \beta = \gamma = 1$, the number of linear clusters $p = \lceil\sqrt{n}\rceil$, the number of ensemble clusterings $T = 12$ and the maximum number of iterations $\tau = 10$, respectively.

Table 1 shows the ARI and NMI values of these algorithms on different data sets. In the table, we see that the density-based algorithms DBSCAN and DP can effectively recognize the nonlinearly-separable clusters on these synthetic data sets. However, their performances are not good on the real data sets. Compared to other algorithms, the SC algorithm has high clustering accuracies on all the given data sets. However, the accelerated spectral clustering algorithms SC+N, SC+RF and SC+KM reduce the clustering accuracies of the original algorithm. Although the objective function of the KKM algorithm is equivalent to that of the SC algorithm, the clustering quality of KKM is weaker than SC on these data sets. According to the experimental results in this table, we see that the clustering accuracy of NKM-NSC is very close to SC. This shows that the proposed algorithm can effectively approximate the clustering results of SC. We also observe that the performance of the proposed algorithm is slightly better than SC on some data sets. The main reason is that the proposed algorithm uses linear clusters to simulate the nonlinear clusters, which can reduce the overfitting of the clustering to some extent. Besides, we can see that the standard deviation of the proposed algorithm on each data set is less than 0.1. Therefore, we can conclude that the proposed algorithm is effective and robust to deal with these data sets.

In addition to the clustering accuracy, we compare the clustering speed of these algorithms on data set MNIST. Table 2 shows the running times of these algorithms with different numbers of objects on the data set. On this test, the running time of the proposed algorithm is more than the MP, SC+N, SC+RF and SC+KM algorithms but is less than the SC, DBSCAN+KD and KKM algorithms. Compared to the DP+KD algorithm, while the size of the data set is more than 4000, the proposed algorithm is more efficient than it. The experimental result is basically consistent with their time complexities. The computational cost of the proposed algorithm is between the KKM algorithm and these accelerated SC algorithms. Compared to the compared algorithms, the proposed algorithm can better balance the effectiveness and efficiency of nonlinearly separable clustering. Therefore, we further test the efficiency of the proposed algorithm on data set KDD-CUP'99. It is worth noting that except the proposed algorithm, the compared algorithms need to compute

Table 2: Running times of different algorithms

Algorithm	2,000	4,000	6,000	8,000	10,000
DBSCAN+KD	14.69	42.47	82.31	168.70	227.70
DP+KD	5.42	19.97	43.84	79.00	118.11
MEAP	62.72	249.39	478.63	754.23	1,045.38
SC	260.69	1,197.60	4,491.70	10,306.00	20,261.00
SC+N	1.02	0.88	2.64	2.70	3.83
SC+RF	3.11	5.88	10.58	11.20	14.44
SC+KM	2.14	5.50	8.78	12.59	15.81
MP	3.30	4.89	8.13	17.61	18.33
KKM	19.19	53.22	100.50	161.86	241.47
NKM-NSC	8.46	19.81	41.40	72.39	92.34

Table 1: ARI and NMI values of different algorithms

Data	Index	DBSCAN+KD	DP+KD	MEAP	MP	KKM
Ring	ARI	**1.0000**	0.3987	0.1476	0.0068±0.0271	0.4801±0.0589
	NMI	**1.0000**	0.6161	0.4924	0.0249±0.8422	0.6214±0.0689
Jain	ARI	0.8700	0.9773	0.2315	0.3812±0.0871	0.3340±0.1372
	NMI	0.9000	0.9492	0.4902	1.6140±0.8635	0.3517±0.0783
Flame	ARI	0.8214	**1.0000**	0.6277	0.6609±0.2482	0.4947±0.1551
	NMI	0.7464	**1.0000**	0.5393	0.7163±0.2934	0.4694±0.1191
Agg	ARI	0.9018	0.9053	0.6812	0.5759±0.2911	**0.9949**±0.2294
	NMI	0.8619	0.8581	0.8376	0.6677±0.1659	0.9776±0.1491
T4.8k	ARI	0.8189	0.6880	0.7546	0.5523±0.6876	0.9818±0.0549
	NMI	0.8590	0.8105	0.8321	0.6059±0.8792	0.9898±0.0308
T7.1k	ARI	0.9952	0.4378	0.4489	0.3676±0.2937	0.9658±0.0634
	NMI	0.9913	0.7118	0.7660	0.6354±0.2072	0.9858±0.0256
Chain	ARI	1.0000	0.7367	0.6763	0.9743±0.1382	0.9777±0.0039
	NMI	1.0000	0.7761	0.6966	0.9777±0.1199	0.9558±0.0063
Atom	ARI	0.9154	0.6394	1.0000	0.5735±0.2110	0.6584±0.1164
	NMI	0.8818	0.6624	1.0000	0.6614±0.2061	0.7021±0.1482
Iris	ARI	0.5681	0.4966	0.7028	0.8586±0.0845	0.8796±0.0097
	NMI	0.7611	0.6607	0.7277	0.8623±0.0932	0.8771±0.0138
Wine	ARI	0.4272	0.4536	0.7586	0.5841±0.1727	0.8673±0.1633
	NMI	0.5309	0.5862	0.7390	0.6539±0.1319	0.8406±0.1051
Breast	ARI	0.6606	0.5038	0.7620	0.4314±0.2716	0.6949±0.0058
	NMI	0.4140	0.4758	0.6640	0.3729±0.2106	0.5723±0.0024
Digits	ARI	0.4772	0.5915	0.6626	0.7081±0.0494	0.6874±0.0656
	NMI	0.5368	0.6438	0.7294	0.8015±0.0396	0.7835±0.0316
Statlog	ARI	0.3021	0.3172	0.2840	0.3121±0.0687	0.5423±0.0510
	NMI	0.4582	0.4869	0.5370	0.4484±0.0430	0.6226±0.0186
MNIST	ARI	0.1125	0.2861	0.3975	0.3459±0.0373	0.4883±0.0423
	NMI	0.3430	0.4856	0.5484	0.5122±0.0126	0.6554±0.0107
Data	Index	SC	SC+N	SC+RF	SC+KM	NKM-NSC
Ring	ARI	**1.0000**±0.0000	0.4031±0.0442	0.3832±0.0762	0.5146±0.1782	**1.0000**±0.0000
	NMI	**1.0000**±0.0000	0.6057±0.0541	0.5758±0.1051	0.6501±0.1451	**1.0000**±0.0000
Jain	ARI	**1.0000**±0.0000	0.6247±0.0097	0.5412±0.0322	0.6199±0.3830	**1.0000**±0.0000
	NMI	**1.0000**±0.0000	0.7011±0.0093	0.6720±0.0564	0.6354±0.3646	**1.0000**±0.0000
Flame	ARI	0.9337±0.0021	0.8889±0.0612	0.3424±0.2313	0.8529±0.2437	0.9650±0.0164
	NMI	0.9269±0.0013	0.8358±0.0563	0.3482±0.2006	0.8192±0.1919	0.9276±0.0310
Agg	ARI	0.7731±0.0011	0.5131±0.2423	0.3637±0.1507	0.3312±0.1323	0.9920±0.0014
	NMI	0.8794±0.0025	0.6067±0.1393	0.5338±0.1270	0.5250±0.1241	**0.9884**±0.0018
T4.8k	ARI	**1.0000**±0.0000	0.4755±0.0503	0.4359±0.0573	0.3569±0.0742	0.8807±0.0406
	NMI	**1.0000**±0.0000	0.6481±0.0459	0.5981±0.0450	0.5540±0.0761	0.8972±0.0229
T7.1k	ARI	**1.0000**±0.0000	05156±0.1055	0.4610±0.1912	0.6057±0.1142	0.8723±0.0821
	NMI	**1.0000**±0.0000	0.6535±0.0913	0.5385±0.1352	0.7868±0.0815	0.9105±0.0483
Chain	ARI	**1.0000**±0.0000	0.9408±0.0051	0.9728±0.0122	0.6919±0.3643	**1.0000**±0.0000
	NMI	**1.0000**±0.0000	0.9024±0.0069	0.9489±0.0195	0.7042±0.3046	**1.0000**±0.0000
Atom	ARI	**1.0000**±0.0000	0.5175±0.1312	0.4354±0.2660	0.8836±0.2634	**1.0000**±0.0000
	NMI	**1.0000**±0.0000	0.5930±0.1102	0.4427±0.2752	0.8968±0.2323	**1.0000**±0.0000
Iris	ARI	**0.9038**±0.0042	0.8871±0.0447	0.8851±0.0216	0.8610±0.0976	0.8921±0.0193
	NMI	**0.8996**±0.0037	0.8871±0.0331	0.8795±0.0220	0.8616±0.0891	0.8893±0.0102
Wine	ARI	**0.9325**±0.0010	0.8906±0.1340	0.8405±0.0539	0.8339±0.1338	0.8977±0.0363
	NMI	**0.9116**±0.0015	0.8737±0.1139	0.8172±0.0513	0.8063±0.0825	0.9082±0.0341
Breast	ARI	**0.7302**±0.0034	0.7011±0.0285	0.7105±0.0387	0.7288±0.1094	0.7248±0.0150
	NMI	**0.6231**±0.0041	0.5798±0.0178	0.5939±0.0427	0.6485±0.1196	0.6230±0.0156
Digits	ARI	**0.8683**±0.0021	0.6078±0.0478	0.6305±0.0547	0.7769±0.0834	0.8434±0.0165
	NMI	**0.9337**±0.0027	0.7563±0.0159	0.6961±0.0302	0.8479±0.0394	0.8838±0.0121
Statlog	ARI	**0.6130**±0.0048	0.5502±0.0549	0.5373±0.0262	0.5306±0.0682	0.6119±0.0220
	NMI	0.6304±0.0059	0.6251±0.0243	0.6166±0.0350	0.6094±0.0412	**0.6344**±0.0178
MNIST	ARI	0.5174±0.0067	0.4078±0.0054	0.3234±0.0386	0.4575±0.0115	**0.5427**±0.0362
	NMI	0.6849±0.0071	0.5272±0.0061	0.4399±0.0276	0.6134±0.0091	0.6632±0.0222

and operate a $n \times n$ pairwise-similarity matrix of all the objects. If the number of objects is no less than $100,000$, the memory size of the computer is required to be no less than 74GB. Therefore, when these algorithms are used to cluster a data set including no less than 100,000 objects in a PC with the low configuration, the out-of-memory error is caused. Due to the fact that the proposed algorithm need not compute the $n \times n$ pairwise-similarity matrix, it can be used to cluster the data set in the PC. Fig. 3 shows the running time of the proposed algorithm with different numbers of objects which are from 100,000 to 1,000,000 on data set KDD-CUP'99, given the PC with 16GB memory. According to the figure, we can see that the proposed algorithm is suitable to cluster large-scale data sets on a PC with the low configuration.

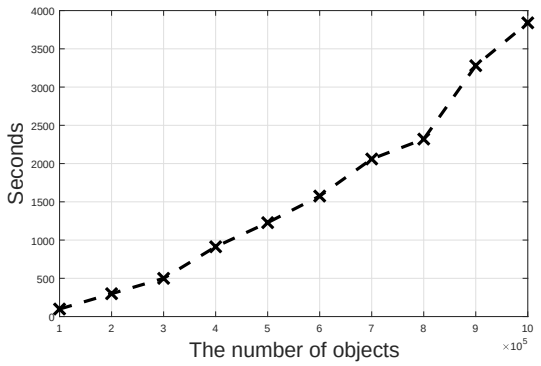
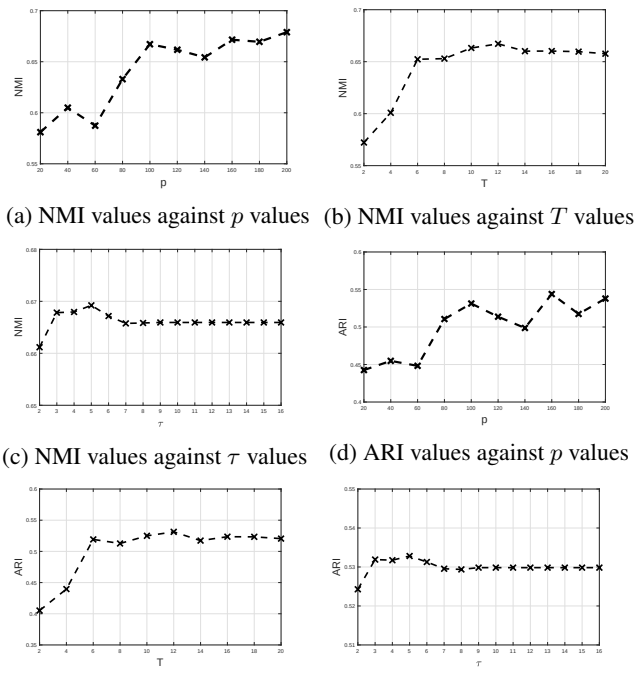


Figure 3: The running times on data set KDD-CUP'99

Next, we analyze the effects of the number of linear clusters p, the number of ensemble clusterings T and the number of iterations τ on the performance of the proposed algorithm by the experiments. Fig. 4 shows the NMI and ARI values against different p, T and τ on data set MNIST, while other parameters are fixed. According to Figs.4(a,b,d,e), we can see that the NMI and ARI values of the proposed algorithm basically increase as the value of the parameters p and T increase. However, we also see that the NMI and ARI values more slowly increase, after the parameter values are growing to a certain extent. According to the time complexity of the proposed algorithm, we know that the computing cost increases as the values of the parameters p and T increase. This experimental result tells us that we should select the suitable values of the parameters to balance the computing cost and the clustering accuracy. According to Figs.4(c,f), we can see that the NMI and ARI values do not change, after the number of iterations is more than 6. This indicates that the proposed algorithm can rapidly converge.

Conclusions

In this paper, we have proposed a new clustering algorithm for nonlinearly separable data sets. It is an extension of the k-means algorithm. Its optimization objective function is made up of three terms including a linearly separable clustering, a nonlinearly separable clustering and an ensemble clustering. The proposed algorithm can rapidly and effectively recognize non-linearly separable clusters. In the experimental analysis, we have compared the proposed algorithm with other nonlinearly separable clustering algorithms on synthetic and real data sets. The comparison results have illustrated that the performance of the proposed algorithm is very effective and efficient.



(a) NMI values against p values (b) NMI values against T values

(c) NMI values against τ values (d) ARI values against p values

(e) ARI values against T values (f) ARI values against τ values

Figure 4: Effect of the parameters

Acknowledgments

The authors are very grateful to the editors and reviewers for their valuable comments and suggestions. This work is supported by the National Natural Science Foundation of China (Nos. 61432011, 61773247, 61876103, 61976128, 61573229) and the 1331 Engineering Project of Shanxi Province, China.

References

Aggarwal, C. C., and Reddy, C. K., eds. 2014. *Data Clustering: Algorithms and Applications*. CRC Press.

Beckmann, N.; Kriegel, H.; Schneider, R.; and Seeger, B. 1990. The r*-tree: An efficient and robust access method for points and rectangles. In *In Proceedings of the ACM SIGMOD International Conference on Management of Data*, 322–331.

Benchmarks. Clustering benchmarks. https://github.com/deric/clustering-benchmark.

Bentley, J. 1975. Multidimensional binary search trees used for associative searching. *Communications of the ACM* 18(9):509.

Berchtold, S.; Keim, D.; and Kriegel, H. 1996. The x-tree: An efficient and robust access method for points and rectangles. In *In Proceedings of International Conference on Very Large Data Bases*, 28–39.

Chen, X., and Cai, D. 2011. Large scale spectral clustering with landmark-based representation. In *AAAI Conference on Artificial Intelligence*.

Chen, W. Y.; Bai, H.; Bai, H.; Chang, E. Y.; and Chang, E. Y. 2011. Parallel spectral clustering in distributed systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(3):568–586.

Cheng, Y. 1995. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Dash, M.; Liu, H.; and Xu, X. 2001. '1+1¿2: Merging distance and density based clustering. In *In Proceedings of the Seventh International Conference on Database Systems for Advanced Applications*, 32–39.

Dean, J., and Ghemawat, S. 2008. Mapreduce: simplified data processing on large clusters. *Communications of the ACM* 51:107–113.

Dhillon, I. S.; Guan, Y.; and Kulis, B. 2007. Weighted graph cuts without eigenvectors: A multilevel approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(11):1944–1957.

Ester, M.; Kriegel, H.-P.; Sander, J.; and Xu, X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. 226–231. AAAI Press.

He, Y.; Tan, H.; and et al., W. L. 2011. Mr-dbscan: An efficient parallel density-based clustering algorithm using mapreduce.

Huang, D.; Wang, C.; Wu, J.; Lai, J.; and Kwoh, C. Ultra-scalable spectral clustering and ensemble clustering. *IEEE Transactions on Knowledge and Data Engineering*.

Jain, A. K. 2008. Data clustering: 50 years beyond k-means. In Daelemans, W.; Goethals, B.; and Morik, K., eds., *Machine Learning and Knowledge Discovery in Databases*, 3–4. Berlin, Heidelberg: Springer Berlin Heidelberg.

Kim, Y.; Shim, K.; Kim, M.; and Lee, J. 2014. Dbcure-mr: An efficient density-based clustering algorithm for large data using mapreduce. *Information Systems* 42:14–35.

Liang, J.; Bai, L.; Dang, C.; and Cao, F. 2012. The k-means-type algorithms versus imbalanced data distributions. *IEEE Transactions on Fuzzy Systems* 20(4):728–745.

Liu, M.; Jiang, X.; and Kot, A. C. 2009. A multi-prototype clustering algorithm. *Pattern Recognition* 42(5):689 – 698.

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 281–297. Berkeley: University of California Press.

Mohan, M., and Monteleoni, C. 2017. Beyond the nystrom approximation: Speeding up spectral clustering using uniform sampling and weighted kernel k-means. In *Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2494–2500.

Nanda, N., and Panda, G. 2015. Design of computationally efficient density-based clustering algorithms. *Data and Knowledge Engineering* 95:23–38.

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, 849–856. MIT Press.

Rahimi, A., and Recht, B. 2007. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, 1177–1184.

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

Scholkopf, B.; Smola, A.; Smola, E.; and Mller, K.-R. 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* 10:1299–1319.

Shi, J., and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8):888–905.

Viswanath, P., and Pinkesh, R. 2006. L-dbscan : A fast hybrid density based clustering method. In *In Proceedings of 18th International Conference on Pattern Recognition*, 912–915.

Wang, C.-D., and Lai, J.-H. 2016. *Nonlinear Clustering: Methods and Applications*. Springer International Publishing. 253–302.

Wang, C. D.; Lai, J. H.; Suen, C. Y.; and Zhu, J. Y. 2013. Multi-exemplar affinity propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(9):2223–2237.

Williams, C., and Seeger, M. 2001. In Leen, T.; Dietterich, T.; and Tresp, V., eds., *Advances in Neural Information Processing Systems*, 682–688.

Witten, I. H., and Frank, E. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufman, San Francisco.

Wu, J.; Liu, H.; Xiong, H.; Cao, J.; and Chen, J. 2015. K-means-based consensus clustering: a unified view. *IEEE Transactions on Knowledge and Data Engineering* 27(1):155–169.

Yan, D.; Huang, L.; and Jordan, M. 2009. Fast approximate spectral clustering. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 907–916.

Yu, J., and Cheng, Q. 2004. The upper bound of the optimal number of clusters in fuzzy clustering. *Science in China Series F: Information Sciences* 44(2):119–125.

Yu, Z.; Zhu, X.; Wong, H.; You, J.; Zhang, J.; and Han, G. 2017. K-means-based consensus clustering: a unified view. *Distribution-based cluster structure selection* 47(11):3554–3567.

Zhou, Z.-H. 2012. *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall/CRC, 1st edition.