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-
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 × m data matrix with n objects and m features. We implement the linear clustering on X, which can produce a n × 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 × k partition matrix H of linear clusters. Let U be a n × k desired partition matrix of objects. We hope to integrate W and H to approximate the partition matrix U, i.e., W H ≈ 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,
we define them as follows.

\[ f_{L(t)} = ||X - WtV_t||_F^2 \]  

(1)
is a validity function of the \( t \)-th 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 \( t \)-th linear clustering, where \( w_{t(ij)} \) is the membership of the \( i \)-th object to the \( j \)-th linear cluster, and \( V_t \) is the \( p \times m \) center matrix of the \( t \)-th linear clustering, where \( v_{t(ji)} \) is the \( j \)-th row of \( V_t \) representing the cluster center of the \( j \)-th 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 \( X \).

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

(2)
is a validity function of the \( t \)-th nonlinear clustering on \( V_t \) which are used to represent the linear clusters obtained by the \( t \)-th 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 \( D_t = [d_{t(ij)}] \) is a \( k \times k \) diagonal matrix with \( d_{t(ij)} = \sum_{q=1}^p h_{t(iq)} \). 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.

\[ f_{E(t)} = ||W_t \hat{H}_t - UG_t||_F^2 \]  

(3)
is a validity function of the \( t \)-th 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 \( t \)-th 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.

\[ \min_U F = \sum_{t=1}^T \alpha f_{L(t)} + \beta f_{G(t)} + \gamma f_{E(t)}, \]  

(4)
subject to

\[ \begin{align*}
    w_{t(ij)} & \in \{0, 1\}, \sum_{i=1}^n w_{t(ij)} = 1, 1 < \sum_{i=1}^n w_{t(ij)} < n, \\
    h_{t(ij)} & \in \{0, 1\}, \sum_{j=1}^m h_{t(ij)} = 1, 1 < \sum_{j=1}^m h_{t(ij)} < n, \\
    u_{ij} & \in \{0, 1\}, \sum_{i=1}^n u_{ij} = 1, 1 < \sum_{i=1}^n u_{ij} < n,
\end{align*} \]

(5)

where \( \alpha, \beta, \gamma \) 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 P1: Fix \( H \), \( U \), and \( G \), solve \( \min_{W_t,V_t} F \); Problem P2: Fix \( W \), \( V \), and \( U \), solve \( \min_{H_t,G_t} F \); Problem P3: Fix \( W \), \( V \), and \( H \), solve \( \min_{U,G_t} F \).

We use a \( k \)-means-like paradigm to approximately obtain its optimal solution.

#### Solution for Problem P1

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 P1 becomes

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

(6)

In order to solve the problem, we have

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

(7)

where \( X_t' = [\alpha'^{1/2} X_t, \gamma'^{1/2} U G_t] \) is a concatenated matrix with \( X_t \) and \( U G_t \), and \( V_t' = [\alpha'^{1/2} V_t, \gamma'^{1/2} H_t] \) is a concatenated matrix with \( V_t \) and \( H_t \). While \( U G_t \) is seen as the new features of the original data \( X_t \), \( X_t' \) is a new representation of the original data \( X_t \). \( V_t' \) 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

\[ w_{t(ij)} = \begin{cases} 
    1, & j = \arg \min_{j=1}^p ||X_t' - V_t'(i)||^2, \\
    0, & \text{otherwise}.
\end{cases} \]

(8)

for \( 1 \leq i \leq N, 1 \leq j \leq P, 1 \leq t \leq T \), where \( x_t' \) is the \( i \)-th row of \( X_t' \) and \( V_t'(i) \) is the \( i \)-th row of \( V_t' \). Given \( W_t \), the minimizer \( V_t' \) is given by

\[ V_t'(i) = \frac{\sum_{i=1}^N w_{t(ij)} x_t'}{\sum_{i=1}^N w_{t(ij)}}, \]

(9)
for $1 \leq j \leq P, 1 \leq t \leq T$. Therefore, the computational complexity of solving Problem $P_1$ is $O(n(m+k)pO)$ 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

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

We have

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

In this case,

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

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

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

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

**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

$$\min_{U} \sum_{t=1}^{T} ||W_t \hat{H}_t - U G_t||_F^2. \quad (14)$$

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

$$g_{t(i)} = \frac{\sum_{l=1}^{n} u_{il} q_{t(i)}}{\sum_{i=1}^{n} u_{il}}, \quad (14)$$

where $q_{t(i)}$ is the $i$th 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

$$u_{il} = \begin{cases} 1, & l = \arg \min_{h=1}^{k} \sum_{t=1}^{T} ||q_{t(i)} - g_{t(h)||}^2, \\ 0, & \text{otherwise}, \end{cases} \quad (15)$$

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

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

$$\left( \frac{f_{G(t)} - \min_{t=1}^{T} f_{G(t)}}{\max_{t=1}^{T} f_{G(t)} - \min_{t=1}^{T} f_{G(t)}} \right)^{\frac{T}{\max_{t=1}^{T} f_{G(t)}} - \frac{T}{\min_{t=1}^{T} f_{G(t)}}}$$

to evaluate the quality of $Q_t$. We select the partition matrix $Q_t$ which satisfies $r = \arg \min_{t} \theta_t$, 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 $\alpha, \beta, \gamma, p, T$ and $\tau$ 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 $\alpha, \beta$ and $\gamma$, 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 $T > 1, \alpha = 0, \beta \neq 0$ 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 $T > 1, \alpha = 0$ and $\gamma \neq 0$, the proposed algorithm becomes a cluster ensemble algorithm. For the parameter $\tau$, 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 $O((n(m+k)pO + (k+m)pO)T \tau)$, where $\tau$ 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 $\tau$. 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 $\tau$ 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 (Estet et al. 1996) with KD-Tree (DBSCAN+KD), density peaks clustering (Rodriguez and Laio 2014) with KD-Tree (DP+KD), multi-exemplar affinity propagation

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 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 16GB 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 $\delta$ 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 $\varepsilon_X$ 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 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. Accord-
<table>
<thead>
<tr>
<th>Data</th>
<th>Index</th>
<th>DBSCAN+KD</th>
<th>DP+KD</th>
<th>MEAP</th>
<th>MP</th>
<th>KKM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ring</td>
<td>ARI NMI</td>
<td><strong>1.0000</strong></td>
<td>0.3987</td>
<td>0.1476</td>
<td>0.0068±0.00271</td>
<td>0.4801±0.0589</td>
</tr>
<tr>
<td>Jain</td>
<td>ARI NMI</td>
<td>0.8700</td>
<td>0.9773</td>
<td>0.2315</td>
<td>0.3812±0.0871</td>
<td>0.3340±0.1372</td>
</tr>
<tr>
<td>Flame</td>
<td>ARI NMI</td>
<td>0.8214</td>
<td>0.7464</td>
<td><strong>1.0000</strong></td>
<td>0.6277</td>
<td>0.6669±0.2482</td>
</tr>
<tr>
<td>Agg</td>
<td>ARI NMI</td>
<td>0.9018</td>
<td>0.9053</td>
<td>0.6812</td>
<td>0.5759±0.2911</td>
<td>0.9994±0.2294</td>
</tr>
<tr>
<td>T4.8k</td>
<td>ARI NMI</td>
<td>0.8189</td>
<td>0.8800</td>
<td>0.7546</td>
<td>0.5523±0.6876</td>
<td>0.9818±0.0549</td>
</tr>
<tr>
<td>T7.1k</td>
<td>ARI NMI</td>
<td>0.9952</td>
<td>0.9913</td>
<td>0.4378</td>
<td>0.4489</td>
<td>0.3676±0.2937</td>
</tr>
<tr>
<td>Chain</td>
<td>ARI NMI</td>
<td>0.9013</td>
<td>0.7961</td>
<td>0.6763</td>
<td>0.9743±0.1382</td>
<td>0.9777±0.0039</td>
</tr>
<tr>
<td>Atom</td>
<td>ARI NMI</td>
<td>0.9515</td>
<td>0.6394</td>
<td>1.0000</td>
<td>0.5735±0.2110</td>
<td>0.6584±0.1164</td>
</tr>
<tr>
<td>Iris</td>
<td>ARI NMI</td>
<td>0.5681</td>
<td>0.4966</td>
<td>0.7028</td>
<td>0.8586±0.0845</td>
<td>0.8796±0.0097</td>
</tr>
<tr>
<td>Wine</td>
<td>ARI NMI</td>
<td>0.4272</td>
<td>0.4536</td>
<td>0.7586</td>
<td>0.5841±0.1727</td>
<td>0.8673±0.1633</td>
</tr>
<tr>
<td>Breast</td>
<td>ARI NMI</td>
<td>0.6606</td>
<td>0.5386</td>
<td>0.6720</td>
<td>0.4314±0.2716</td>
<td>0.6594±0.0058</td>
</tr>
<tr>
<td>Digits</td>
<td>ARI NMI</td>
<td>0.4772</td>
<td>0.4758</td>
<td>0.6640</td>
<td>0.7081±0.0494</td>
<td>0.6874±0.0656</td>
</tr>
<tr>
<td>Statlog</td>
<td>ARI NMI</td>
<td>0.3021</td>
<td>0.3172</td>
<td>0.2840</td>
<td>0.3121±0.0687</td>
<td>0.5423±0.0510</td>
</tr>
<tr>
<td>MNIST</td>
<td>ARI NMI</td>
<td>0.1125</td>
<td>0.2861</td>
<td>0.3975</td>
<td>0.3459±0.0373</td>
<td>0.4883±0.0425</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data</th>
<th>Index</th>
<th>SC</th>
<th>SC+N</th>
<th>SC+RF</th>
<th>SC+KM</th>
<th>NKM-NSC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ring</td>
<td>ARI NMI</td>
<td><strong>1.0000±0.0000</strong></td>
<td>0.4031±0.0442</td>
<td>0.3832±0.0762</td>
<td>0.5146±0.1782</td>
<td><strong>1.0000±0.0000</strong></td>
</tr>
<tr>
<td>Jain</td>
<td>ARI NMI</td>
<td>0.9013±0.0000</td>
<td>0.6247±0.0097</td>
<td>0.5412±0.0322</td>
<td>0.6199±0.3830</td>
<td><strong>1.0000±0.0000</strong></td>
</tr>
<tr>
<td>Flame</td>
<td>ARI NMI</td>
<td>0.9337±0.0021</td>
<td>0.8889±0.0612</td>
<td>0.3424±0.2313</td>
<td>0.8529±0.2437</td>
<td>0.9650±0.0164</td>
</tr>
<tr>
<td>Agg</td>
<td>ARI NMI</td>
<td>0.7731±0.0011</td>
<td>0.5131±0.2423</td>
<td>0.3637±0.1507</td>
<td>0.3312±0.1323</td>
<td>0.9920±0.0014</td>
</tr>
<tr>
<td>T4.8k</td>
<td>ARI NMI</td>
<td>0.1000±0.0000</td>
<td>0.4755±0.0503</td>
<td>0.4359±0.0573</td>
<td>0.3569±0.0974</td>
<td>0.8807±0.0406</td>
</tr>
<tr>
<td>T7.1k</td>
<td>ARI NMI</td>
<td>0.1000±0.0000</td>
<td>0.5156±0.1055</td>
<td>0.4610±0.1912</td>
<td>0.6057±0.1142</td>
<td>0.8723±0.0821</td>
</tr>
<tr>
<td>Chain</td>
<td>ARI NMI</td>
<td>0.1000±0.0000</td>
<td>0.5408±0.0501</td>
<td>0.9728±0.0122</td>
<td>0.6919±0.3643</td>
<td>0.1000±0.0000</td>
</tr>
<tr>
<td>Atom</td>
<td>ARI NMI</td>
<td>0.1000±0.0000</td>
<td>0.5175±0.1312</td>
<td>0.4354±0.2660</td>
<td>0.8386±0.2634</td>
<td>0.1000±0.0000</td>
</tr>
<tr>
<td>Iris</td>
<td>ARI NMI</td>
<td>0.9038±0.0042</td>
<td>0.8871±0.0447</td>
<td>0.8851±0.0216</td>
<td>0.8610±0.0976</td>
<td>0.3892±0.0193</td>
</tr>
<tr>
<td>Wine</td>
<td>ARI NMI</td>
<td>0.9325±0.0010</td>
<td>0.8906±0.1340</td>
<td>0.8405±0.0539</td>
<td>0.8392±0.1338</td>
<td>0.8977±0.0363</td>
</tr>
<tr>
<td>Breast</td>
<td>ARI NMI</td>
<td>0.7302±0.0034</td>
<td>0.7011±0.0285</td>
<td>0.7105±0.0387</td>
<td>0.8172±0.0513</td>
<td>0.9063±0.0285</td>
</tr>
<tr>
<td>Digits</td>
<td>ARI NMI</td>
<td>0.3683±0.0021</td>
<td>0.0678±0.0478</td>
<td>0.6305±0.0547</td>
<td>0.7769±0.0834</td>
<td>0.8434±0.0065</td>
</tr>
<tr>
<td>Statlog</td>
<td>ARI NMI</td>
<td>0.6130±0.0048</td>
<td>0.3502±0.0549</td>
<td>0.5373±0.0262</td>
<td>0.5360±0.0682</td>
<td>0.6119±0.0220</td>
</tr>
<tr>
<td>MNIST</td>
<td>ARI NMI</td>
<td>0.5174±0.0067</td>
<td>0.4078±0.0054</td>
<td>0.3234±0.0386</td>
<td>0.4375±0.0115</td>
<td><strong>0.5427±0.0362</strong></td>
</tr>
</tbody>
</table>

Table 1: ARI and NMI values of different algorithms
Table 2: Running times of different algorithms

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

Table 2: Running times of different algorithms

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.

Next, we analyze the effects of the number of linear clusters $p$, the number of ensemble clusterings $T$ and the number of iterations $\tau$ on the performance of the proposed algorithm by the experiments. Fig. 4 shows the NMI and ARI values against different $p$, $T$ and $\tau$ 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.

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


