Postprint: A Distance and Density Based d-K-means Algorithm

Authors: Tang Zekun, Zhu Zeyu, Yang Yi, Li Caihong, Li Lian

Date: 2019-05-10T00:00:00+00:00

Abstract

The K-means algorithm is characterized by its simple implementation and fast speed, making it the most widely used clustering algorithm. To address the shortcomings of the K-means algorithm' s sensitivity to initial cluster centers and noise, the d-K-means algorithm (distance&density) is proposed. Based on the K-means algorithm, it balances the influence of density and distance on clustering, performs weighted processing on data, introduces the "min-max principle" to select initial cluster centers based on weights, and automatically determines the number of cluster centers. Experimental results demonstrate that the d-K-means algorithm can achieve good clustering performance on both low-dimensional and high-dimensional data, better handle data in low-density regions, and more effectively select cluster centers.

Full Text

Preamble

D-K-means Algorithm Based on Distance and Density

Tang Zekun, Zhu Zeyu, Yang Yi, Li Caihong, Li Lian

College of Information Science & Engineering, Lanzhou University, Lanzhou 730000, China

Abstract: The K-means algorithm is the most widely used clustering algorithm due to its simple implementation and fast speed. To address its sensitivity to initial clustering centers and noise, this paper proposes the d-K-means algorithm (distance & density), which balances the influence of density and distance on clustering by weighting the data. The algorithm introduces a "minimax principle" for selecting initial clustering centers based on these weights and automatically determines the optimal number of cluster centers. Experimental results demonstrate that the d-K-means algorithm achieves superior clustering performance

on both low-dimensional and high-dimensional data, better handles low-density region data, and improves cluster center selection.

Keywords: clustering; K-means algorithm; minimax principle; number of cluster centers

0 Introduction

Clustering is a data mining technique that partitions a collection of physical or abstract objects into multiple groups of similar objects. The clusters generated by clustering are sets of data objects where objects within the same cluster are similar to each other and dissimilar to objects in other clusters. As the saying goes, "birds of a feather flock together," and classification problems abound in both natural and social sciences [**?**].

The K-means algorithm has become the most widely used clustering method due to its straightforward concept and ease of implementation. However, it suffers from several limitations. First, the algorithm requires the number of clusters to be specified in advance, making it difficult for users to provide an appropriate value when they have insufficient knowledge about the data. Second, the random initialization of cluster centers often leads to suboptimal local solutions and unstable clustering results [**?**].

In recent years, researchers have proposed numerous optimizations to address these deficiencies in the K-means algorithm [**?**, **?**]. Reference [**?**] calculates the arithmetic mean of all sample points to form the initial clustering centers, considering the distribution of the entire dataset and improving the randomness of initial center selection, but this approach remains vulnerable to noise. Reference [**?**] constructs a minimum spanning tree and uses pruning techniques to dynamically select initial clustering centers based on data distribution, accounting for sample density but significantly increasing computational overhead. Both [**?**] and [**?**] select initial centers based on the overall sample distribution, ignoring the fact that true cluster centers typically have high surrounding density, which can lead to local optima. Reference [**?**] selects centers based on the principle that the farthest sample points cannot belong to the same cluster, avoiding local optima but neglecting density considerations, potentially selecting noise points as initial centers. Reference [**?**] calculates outlier factors to sort the dataset in ascending order, positioning centers toward the front while considering density, and combines this with the "minimax principle" for center selection, preventing local optima but potentially misclassifying low-density region data as outliers and affecting final clustering quality. Reference [**?**] sorts data objects by density and selects the midpoint between the densest point and its nearest neighbor as a new cluster center, then excludes points within a specified radius from further consideration, partially addressing [**?**]' s issues but still neglecting low-density region data when the distance between low- and high-density regions is large. All these algorithms require predetermined cluster numbers, which affects precision when researchers lack thorough understanding of the dataset. Reference [**?**]

generates a candidate set of initial cluster centers based on density parameters, then simulates clustering by traversing this set and selects points with optimal inter-class separability and intra-class compactness, fully considering final clustering quality, but suffers from high time complexity and unscientific density parameter calculation that may assign identical density values to points with significantly different actual densities, potentially removing high-density points and reducing initial center quality. Reference [**?**] combines canopy algorithm concepts with density to better handle low-density regions and automatically determine the number of cluster centers, but only stops after traversing all data points without considering clustering quality, likely classifying outliers and noise points as separate classes and affecting both clustering effectiveness and center count accuracy.

The principle for selecting cluster centers requires relatively dense surrounding points while maintaining sufficient distance between centers to ensure good distribution and prevent local optima. To address initial center selection and cluster number determination, this paper proposes the d-K-means algorithm based on distance and density. The algorithm employs weighting to balance the relationship between density and distance, solving globally to make center selection more consistent with data distribution and reduce iteration counts. By calculating dataset scale and inter-point distances, each point receives a distinct weight. The d-K-means algorithm selects cluster centers based on the "minimax principle," avoiding local optima caused by random initialization, better handling outlier and low-density region data, and automatically determining the number of cluster centers through weights and the BWP index [**?**]. Comparative experiments with four algorithms on five datasets validate that the proposed algorithm significantly improves clustering quality and accuracy.

1.1 Canopy-Kmeans Algorithm

The Canopy clustering algorithm [**?**, **?**] preprocesses data for K-means and can approximate the number of cluster centers through the number of canopies generated when manual determination is impossible. Canopy processes the dataset using two manually determined thresholds, t_1 and t_2, to classify chaotic data into several relatively regular piles. The classification effect is illustrated in [Figure 1: see original paper]. The algorithm proceeds as follows:

1) Determine two thresholds t_1 and t_2 ($t_1 > t_2$).
2) Randomly select a data point from the dataset and calculate its distance to existing canopies (if no canopy exists, the point directly becomes a canopy center).
3) If this distance is less than t_1, mark the data with a weak label and add it to this canopy (the data can also serve as a new canopy for calculating distances to other points).
4) If the distance is less than t_2, mark the data with a strong label and remove it from the dataset, considering it sufficiently close to the canopy to not form a new canopy.

5) Repeat steps 2-4 until the dataset contains no remaining data.

For the problem of difficult cluster number determination, the Canopy-Kmeans algorithm can approximate the number of cluster centers through the number of generated canopies. However, in practical applications, the selection of initial Canopy centers and region sizes significantly impacts clustering quality.

1.2 K-means++ Algorithm

The K-means++ algorithm [**?**, **?**] selects K initial cluster centers based on the following principle: assuming n initial centers have been selected ($0 < n < K$), points farther from the current n centers have higher probability of being selected as the $(n+1)$-th center. When selecting the first center ($n = 1$), the method also randomly selects from existing sample points. This aligns with the intuition that cluster centers should be far apart. Though simple and intuitive, this improvement effectively addresses the randomness in K-means initial center selection [**?**]. The probability calculation function is:

where X is the set of points in the clustering problem, and $D(x)$ calculates the distance from a point to its nearest selected cluster center. Analysis of this probability function reveals that noise points in low-density regions have relatively high probability of being selected as cluster centers, resulting in too few points belonging to such centers that are unlikely to change during subsequent K-means iterations, thereby failing to achieve the desired classification effect with k centers. A dataset composed of two clusters, one containing a noise point, is shown in [Figure 2: see original paper]. Experimental data from [Figure 3: see original paper] and [Figure 4: see original paper] show that K-means++ produces two possible partitions of the dataset. Notably, even ignoring the noise point, K-means++ cannot obtain correct clustering results [**?**].

1.3 DBSCAN Algorithm

DBSCAN [**?**] is a density-based clustering algorithm that discovers high-density connected regions, automatically determines the number of clusters, and handles noise. It forms clusters by finding core objects and connecting them with their neighborhoods. The main concepts are:

Definition 1 (Neighborhood): For a point o in sample set D, the neighborhood of o is the d-dimensional hypersphere region centered at o with radius ε.

Definition 2 (Core Object): An object whose ε-neighborhood contains at least $minPts$ points.

Definition 3 (Directly Density-Reachable): In sample set D, if object q lies within the ε-neighborhood of core object p, then q is directly density-reachable from p.

Definition 4 (Density-Reachable): In sample set D, if there exists a chain of

points $p_1, p_2, \ldots, p_n$ where $p_i \in D$ $(1 \leq i \leq n)$ and p_{i-1} is directly density-reachable from p_i, then p_n is density-reachable from p_1.

Definition 5 (Density-Connected): If there exists an object o such that both objects p and q are density-reachable from o, then p and q are density-connected.

The advantage of DBSCAN is its ability to discover clusters of arbitrary shapes based on density distribution, effectively overcoming K-means++ limitations. However, its disadvantages [**?**] mirror those of most K-means improvements: initial center selection assumes high density within a point' s ε-neighborhood, marking low-density points as noise and potentially eliminating data of interest, which affects final classification accuracy. Additionally, the settings for radius ε and minimum support $minPts$ are highly sensitive.

2 d-K-means Algorithm

2.1 Basic Definitions

The d-K-means algorithm uses Euclidean distance for all distance calculations. Following the greedy strategy from [**?**], the algorithm adaptively calculates the radius ε for each data point p:

$$\varepsilon(p) = \frac{1}{k} \sum_{i \in k\_nearest(k)} d(p, i)$$

where $k\_nearest(k)$ represents the k nearest points to p_i, and $d(\cdot)$ denotes Euclidean distance. In two-dimensional clustering, k is typically set to 4 [**?**], while in other cases it can be set to $\lfloor n/25 \rfloor$ [**?**] (where n is the total number of data samples and $\lfloor \cdot \rfloor$ denotes floor rounding). The weight w_p of object p is calculated based on distances to objects q in its ε-neighborhood, and processed to obtain the center point indicator C_p:

$$C_p = w_p \times \theta_p$$

where w_p reflects the neighborhood density of point p, and θ_p is the distance from p to its nearest existing center i:

$$w_p = \sum_{q \in \varepsilon(p)} \left(1 - \frac{d(p, q)}{range}\right)$$

$$\theta_p = \min_{1 \leq i \leq k} d(p, i)$$

$$range = \sqrt{\sum_{z=0}^{x} (\max_z - \min_z)^2}$$

where k is the number of existing centers, m is the number of data objects in p' s ε-neighborhood, *range* represents the size of the dataset' s vector space (calculated similarly to Euclidean distance), x denotes dataset dimensionality, max and min represent the maximum and minimum values of each dimension, and $\|\cdot\|_2$ denotes squared Euclidean distance. The *range* value is essentially the magnitude of the full dimensional range of the dataset. Each data point in p' s ε-neighborhood contributes a value between 0 and 1 to w_p, with closer points contributing more. A larger w_p indicates more concentrated data around p. A larger θ_p indicates greater distance from existing cluster centers. The center point indicator C_p, obtained by multiplying w_p and θ_p, reflects higher intra-cluster compactness and inter-cluster separability. Since K-means time consumption is primarily determined by iteration count, selecting centers based on C_p effectively reduces iterations and improves time performance.

The d-K-means algorithm employs a new clustering validity index (called the BWP index) from [**?**] to determine whether to continue center selection based on changes in the average BWP value:

$$BWP = \frac{1}{n}\sum_{i=1}^{n}\frac{b(j,i)-w(j,i)}{\max(b(j,i),w(j,i))}$$

where n is the dataset size, and $b(j,i)$, $w(j,i)$ are defined as follows: For dataset S with n objects partitioned into k clusters, the inter-class distance $b(j,i)$ for object i in cluster j is the minimum average distance from this sample to objects in other clusters, while the intra-class distance $w(j,i)$ is the average distance from the object to other objects within cluster j:

$$b(j,i) = \min_{1\le c\le k, c\neq j}\frac{1}{n_c}\sum_{p=1}^{n_c}\|x_p^{(c)} - x_i^{(j)}\|_2$$

$$w(j,i) = \frac{1}{n_j - 1}\sum_{\substack{p=1\\ p\neq i}}^{n_j}\|x_p^{(j)} - x_i^{(j)}\|_2$$

where c and j denote cluster labels, n_c is the number of elements in cluster c, $x_p^{(c)}$ represents the p-th data object in cluster c, and $x_i^{(j)}$ denotes the i-th object in cluster j. As shown, larger $b(j,i)$ indicates better inter-class separability, smaller $w(j,i)$ indicates better intra-class compactness, and larger BWP values indicate superior clustering quality.

2.2 d-K-means Algorithm Description

The d-K-means algorithm selects centers sequentially as follows: Based on the "minimax principle," it selects the data point with the maximum center indicator value as an experimental cluster center for pre-classification, assigning all

points to the nearest center's cluster. It then compares the change in the average BWP index before and after pre-classification. If the average BWP increases, the point becomes a cluster center, and following the canopy algorithm concept, points within this center' s ε-neighborhood are excluded from subsequent center selection. Since new centers may change each point' s nearest center, the algorithm updates all center indicators after generating each center. If the average BWP decreases or no points remain eligible for selection, the process terminates, automatically determining k cluster centers.

The center selection approach achieves clustering effects where centers are sufficiently separated while surrounded by dense points. The weight formula shows that higher surrounding density yields larger w_p. Applying the minimax principle to C_p means points with large weights and far distances from existing centers have higher selection probability.

Initially, with no centers, C_p cannot be computed due to missing θ parameters. Since more points within a spatial range indicate better convergence of the objective function when that point serves as a center, the algorithm selects the maximum-weight point as the first center to improve intra-cluster compactness. This aligns with the center selection philosophy in [**?**, **?**] and actual clustering effects where centers have dense neighborhoods. The center selection process is illustrated in [Figure 5: see original paper] through [Figure 8: see original paper]. The algorithm first selects the maximum-weight point as the initial center, then selects two additional centers based on C_p. When selecting a fourth center would decrease the average BWP, the process stops, yielding three centers.

The d-K-means algorithm balances distance and density influences through weights and center indicators. The center indicator enables low-density region data with large distances from current centers (but not necessarily large weights) to become potential centers. The minimax principle and first-center selection strategy eliminate randomness in initial center selection. Integrating the BWP index with canopy concepts enables automatic cluster number determination while avoiding the pitfall of classifying outliers as separate clusters, thereby ensuring clustering quality.

2.3 d-K-means Algorithm Flow

Input: Set X of n data objects.

Output: k cluster centers.

The algorithm flow is shown in [Figure 9: see original paper].

a) Calculate ε radius and weights for n data points.
b) Select the point with maximum weight as the first cluster center.
c) Compute each point's center indicator and select the point with maximum C_p for pre-classification.
d) Calculate the average BWP index for n points after pre-classification.

e) If the average BWP increases, the point becomes a cluster center; exclude its ε-neighborhood from further selection and proceed to step f. If the average BWP decreases, proceed to step g.
f) If clusterable points remain, return to step c; otherwise, proceed to step g.
g) Execute K-means with the generated centers as initial cluster centers to obtain final clustering results.

3 Experiments

3.1 Datasets

To validate the effectiveness of d-K-means in selecting initial cluster centers, we used UCI datasets and U.S. weather bureau climate classification data (Weather). UCI is a standard machine learning repository from the University of California, Irvine, with clearly labeled categories enabling direct observation of clustering quality. Experiments tested five datasets: Iris, Wine, Seeds, Pima, and Weather. Performance metrics included BWP index, Rand index, silhouette coefficient, Jaccard coefficient, iteration count, and accuracy, compared against traditional K-means, K-means++, and algorithms from [**?**] and [**?**]. describes the dataset parameters.

3.2 Experimental Results

Different features in datasets often have different scales, affecting classification results. To eliminate scale effects, we normalized data using Min-Max Scaling to ensure each feature contributed equally to the algorithm. For a dataset with i features, each feature was normalized as follows:

$$x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

[Figure 10: see original paper] through [Figure 15: see original paper] show the BWP index, Rand index, silhouette coefficient, Jaccard coefficient, iteration count, and accuracy for the proposed algorithm versus four comparison algorithms on Iris, Wine, Seeds, and Pima datasets.

The results in [Figure 10: see original paper] through [Figure 13: see original paper] demonstrate that d-K-means achieves significantly better BWP, Rand index, silhouette coefficient, and Jaccard coefficient than traditional K-means, K-means++, and [**?**]' s algorithm. K-means randomly selects initial centers, while K-means++ incorporates distance but retains randomness. Algorithm [**?**] selects centers based on density parameters without considering inter-point distances, yielding unscientific density calculations that affect center quality. On Iris, Wine, and Seeds, d-K-means performs similarly to [**?**], but outperforms it on Pima due to that dataset' s missing values and [**?**]' s vulnerability to outliers.

The results show d-K-means achieves better clustering with higher intra-class compactness and inter-class separability.

Since K-means and K-means++ exhibit randomness in center selection, causing unstable accuracy and iteration counts, we ran each 10 times and averaged the results. [Figure 14: see original paper] and [Figure 15: see original paper] show that except for slightly lower accuracy than [**?**] on Seeds, d-K-means achieves optimal iteration counts and accuracy on all datasets, demonstrating effective performance on both low- and high-dimensional data.

2) Weather Dataset Test

The Weather dataset includes September 12, 2018 measurements from U.S. weather stations with discrete distributions and many low-density regions. Experiments classified locations using normalized humidity and Fahrenheit temperature features. presents the Weather data, and [Figure 16: see original paper] shows its distribution. Comparing d-K-means with K-means, K-means++, and algorithms [**?**] and [**?**], summarizes the clustering results and accuracy.

Due to high discreteness, the dataset' s average distance is large, causing algorithms [**?**] and [**?**] to fail in determining the correct number of clusters. K-means randomly selected centers at (0.134,0.86), (0.413,0.873), (0.531,0.229), and (0.912,0.875), but the first two centers attracted no data, yielding only two clusters with 52.6% accuracy after three iterations. K-means++ selected points 4, 11, 10, and 6 as centers, achieving 57.9% accuracy after three iterations. Algorithm [**?**] selected points 8, 16, and 1 as centers based on a fixed distance threshold (average distance = 0.5235), generating three clusters with 68.4% accuracy after two iterations. Algorithm [**?**] similarly selected points 8, 14, and 1, also achieving 68.4% accuracy.

In contrast, d-K-means adaptively calculates each point' s ε radius, making neighborhood exclusion more scientific. Its weight calculation incorporates inter-point distances, better reflecting density distribution than [**?**]' s density parameters. The center indicator balances distance and density relationships. d-K-means selected points 8, 13, 1, and 19 as centers, achieving 89.5% accuracy after two iterations.

Experimental results across five datasets demonstrate that d-K-means consistently achieves superior clustering quality, accuracy, and iteration speed for low/high-dimensional and dense/discrete data.

4 Conclusion

As data rapidly expands and diversifies, automatic cluster number selection becomes increasingly important. The d-K-means algorithm balances distance and density through weights and center indicators. Compared to K-means' vulnerability to local optima and manual cluster number selection, the proposed algorithm automatically determines cluster numbers while improving clustering quality, iteration speed, and classification accuracy.

References

[1] Han Jiawei, Kamber M, Pei Jian, et al. Data mining: concept and technology [M]. Fan Ming, Meng Xiaofeng, Translated. 3rd ed. Beijing: Machinery Industry Press, 2012: 211-213.

[2] Celebi M E, Kingravi H A, Vela P A. A comparative study of efficient initialization methods for the K-means clustering algorithm [J]. Expert Systems with Applications, 2013, 40(1): 200-210.

[3] Bagirov A M. Modified global K-means algorithm for minimum sum-of-squares clustering problems [J]. Pattern Recognition, 2008, 41(10): 3192-3199.

[4] Tzortzis G, Likas A. The MinMax K-means clustering algorithm [J]. Pattern Recognition, 2014, 47(7): 2505-2516.

[5] Cai Longfei. Clustering analysis of improved K-means algorithm using hard C-means [J]. Science and Technology Innovation Herald, 2007(24): 144-145.

[6] Feng Bo, Hao Wenning, Chen Gang, et al. Optimization to K-means initial cluster centers[J]. Computer Engineering and Applications, 2013, 49(14): 182-185.

[7] Zhai Donghai, Yu Jiang, Gao Fei, et al. K-means text clustering algorithm based on initial cluster centers selection according to maximum distance [J]. Application Research of Computers, 2014, 31(3): 713-715,719.

[8] Tang Dongkai, Wang Hongmei, Hu Ming, et al. Optimizing initial cluster center of improved K-means algorithm [J]. Journal of Chinese Computer Systems, 2018, 39(8): 1819-1823.

[9] Zhou Weiben, Shi Yuexiang. Optimization algorithm of K-means clustering center selection based on density [J]. Application Research of Computers, 2012, 29(5): 1726-1728.

[10] Jia Ruiyu, Song Jianlin. K-means optimal clustering number determination method based on clustering center optimization [J]. Microelectronics & Computer, 2016, 33(5): 62-66,71.

[11] Zhang Geng, Zhang Chengchang, Zhang Huayu. Improved K-means algorithm based on density canopy [J]. Knowledge-Based Systems, 2018, 145: 289-297.

[12] Wang Fasheng, Lu Mingyu, Zhao Qingjie, et al. Particle filtering algorithm [J]. Chinese Journal of Computers, 2014, 37(8): 1679-1694.

[13] Zhang Lin, Mou Xiangwei. Chinese text clustering algorithm based on Canopy+K-means [J]. Library Tribune, 2018, 38(6): 113-119.

[14] Mao Dianhui. Improved canopy-Kmeans algorithm based on MapReduce [J]. Computer Engineering and Applications, 2012, 48(27): 22-26.

[15] Yoder J, Priebe C E. Semi-supervised K-means+ [J]. Journal of Statistical Computation & Simulation, 2016(3).

[16] Zhang Yazhou, Yu Zhengsheng. Video summarization generation algorithm based on K-means+ clustering [J]. Industrial Control Computer, 2017, 30(7): 129-130.

[17] Brunsch T, Röglin H. A bad instance for K-means+ [J]. Theoretical Computer Science, 2013, 505(9): 19-26.

[18] Agarwala M, Jaiswalb R, Pal A. K-means+ under approximation stability [J]. Theoretical Computer Science, 2015, 588: 37-51.

[19] Feng Zhenhua, Qian Xuezhong, Zhao Nana. Greedy DBSCAN: an improved DBSCAN algorithm on multi-density clustering [J]. Application Research of Computers, 2016, 33(9): 2693-2696,2700.

[20] Nasibov E N; Ulutagay G. Robustness of density-based clustering methods with various neighborhood relations [J]. Fuzzy Sets And Systems, 2009, 160(24): 3601-3615.

[21] Sun Lingyan. Research of clustering algorithm based on density [D]. Taiyuan: North University of China, 2009.

[22] Daszykowski M, Walczak B, Massart D L. Looking for natural patterns in data: part 1. density-based approach [J]. Chemometrics and Intelligent Laboratory Systems, 2001, 56(2): 83-92.

[23] Tang Rongzhi, Duan Huichuan, Sun Haitao. Research on data normalization for SVM training [J]. Journal of Shandong Normal University: Natural Science, 2016, 31(4): 60-65.

Note: Figure translations are in progress. See original paper for figures.