Fast Estimation for the Number of Clusters

Xiaohong Zhang, Zhenzhen He, Zongpu Jia, and Jianji Ren(✉)

College of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 45400, Henan, China
{xh.zhang,jiazp,renjianji}@hpu.edu.cn, hezzedu@163.com

Abstract. Clustering analysis has been widely used in many areas. In many cases, the number of clusters is required to been assigned artificially, while inappropriate assignments affect analysis negatively. Many solutions have been proposed to estimate the optimal number of clusters. However, the accuracy of those solutions drop severely on overlapping data sets. To handle the accuracy problem, we propose a fast estimation solution based on the cluster centers selected in a static way. In the solution, each data point is assigned with one score calculated according to a density-distance model. The score of each data point does not change any more once it is generated. The solution takes the top k data points with the highest scores as the centers of k clusters. It utilizes the significant change of the minimal distance between cluster centers to identify the optimal number of the clusters in overlapping data sets. The experiment results verify the usefulness and effectiveness of our solution.

Keywords: Clustering · The number of clusters · Density · Distance

1 Introduction

Clustering analysis is one of the ways to perform unsupervised analysis [1]. It is dedicated to dividing data into clusters with the goal of the similarity between data within the cluster and minimizing the similarity between data between clusters [2]. It has been widely used in many areas such as image processing, bioinformatics, in-depth learning, pattern recognition and so on. Clustering analysis can be classified into partition-based clustering, density-based clustering [3], grid-based clustering, hierarchical clustering [4] and so on [5,6]. However, in many cases, the number of clusters must be assigned artificially, while inappropriate assignments affect analysis results negatively. If the number of clusters is much larger than the actual number of clusters, the resulting clustering results will be very complicated and the characteristics of the data cannot be analyzed. If the number of clusters is much smaller than the actual number of clusters, some valuable information will be lost in the clustering results. The loss of this information leads to the inability to obtain valuable information in later data mining.

X. Wang et al. (Eds.): 6GN 2020, LNICST 337, pp. 357–370, 2020.
https://doi.org/10.1007/978-3-030-63941-9_27

Therefore, many solutions have been proposed to determine the optimal number of clusters. Some solutions utilize cluster validity Indexes, e.g., DB Index [7], I Index [8] and Xie-Beni Index [9], to determine the optimal number. Some solutions exploit heuristics to deduce the number. For example, the solution of Laio et al. [10] is one of those solutions. It estimates the optimal number according to density and distance (the density of data and the distance between). However, the heuristic still needs to input the number of clusters artificially, and cannot fully cluster automatically. Recently, Gupta et al. [11] propose a solution to identify the optimal number according to the last leap and the last major leap of the minimal distances between cluster centers. However, when running on overlapping data sets, the accuracy of most of those solutions drops severely.

In order to solve the problem of poor estimation of cluster numbers on overlapping data sets, we propose an algorithm that focuses on cluster number estimation on overlapping data sets. And this method has a very fast speed. The solution selects cluster centers in a static way. It generates a score for each data point according to a density-distance model. The score of each data point does not change any more once it is generated. The solution takes the top k data points with the highest scores as the centers of k clusters. It utilizes the significant change of the minimal distance between cluster centers to identify the optimal number of the clusters in overlapping data sets.

The rest of this paper is organized as follows: we review the relevant published work in Sect. 2. After analyzing the estimation problem of the number of the clusters in overlapping data sets in Sect. 3, we elaborate our solution in Sect. 4. The experimental results are discussed in Sect. 5. Finally, the paper concludes in Sect. 6.

2 Related Work

It is very important to determine the number of clusters which data points are grouped into, especially for partition-based clustering solutions. However, it is not easy to estimate the optimal value of the number. Fortunately, cluster validity Indexes [12–16] provide a useful tool for the estimation. Davies and Bouldin [7] proposed DBI index based on inter-cluster similarities to obtain the best cluster number. Xie and Beni [9] put forward Xie-Beni Index based on intra cluster compactness and inter cluster separation, and utilize the index to determine the optimal number. Bensaid [17] and Ren et al. [18] improve the solution of Xie and Beni to enhance the reliability and robustness of that solution, respectively. Some other validity Indexes [19], e.g., Bayesian Information Criterion (BIC) [20], diversity [21], intra-cluster coefficient and inter-cluster coefficient [22], are also exploited to estimate the number of clusters. To obtain better estimation, some solutions even apply multiple indexes in the estimation [5].

In addition to cluster validity indexes, some other factors are also utilized for the estimation. The solutions proposed by Wang et al. [23] and Laio et al. [10] perform the estimation according to the factors related to density. Concretely, the solution of Laio et al. calculates a produce of density and distance for each

data point and estimations the number of clusters based on those products. Because the user is required to input the number of clusters according to the visualization. Therefore, it is impossible to process data in batches. Recently, Gupta et al. [11] observed that the last significant drop of the distance between cluster centers indicated the natural number of clusters. Based on the observation, they proposed the Last Leap solution (LL) and the Last Major Leap solution (LML) to estimate the natural number of clusters.

Many algorithms already have a very high accuracy for determining the number of clusters in some simple data sets. However, when running on the data sets with overlapping clusters, the accuracy of those solutions drops severely. Here, the overlapping data set refers to a data set with no obvious boundary between clusters. For example, Fig. 1 shows a non-overlapping two-dimensional data set, and Fig. 2 shows an overlapping two-dimensional data set. At the same time, a detailed explanation of the notions mentioned below is shown in Table 1.

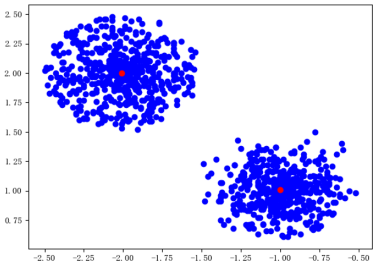


Fig. 1. Non-overlapping

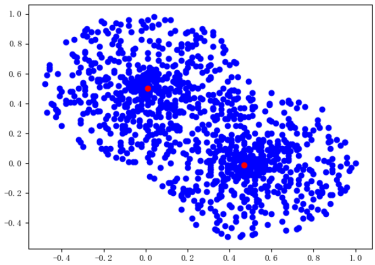


Fig. 2. Overlapping

3 Motivation

Given a data set-P, $P = \{p_1, p_2, \ldots, p_m\}$, where $(\forall p_i)p_i \in \mathbb{R}^d$, partition-based solutions try to divide P into k subsets noted as C_1, C_2, ..., C_k. Each of those subsets is known as a cluster and identified by a cluster center. Partition-based solutions require users to offer the number of clusters, i.e., the number of k, which indicates that users are involved in the procedure of clustering somehow. To make sure that clustering is truly unsupervised, clustering solutions should be equipped with the ability of estimating the optimal number of clusters. The objective of this work is to search for the optimal number from a set-$K = \{k_i | k_i \in N^* and\ k_i < k_{max}\ and\ k_{max} = \sqrt{m}\}$.

Gupta et al. observed that the last significant drop of the minimal distances between cluster centers indicates the optimal number of clusters. Based on the observation, they proposed the Last Leap solution (LL) and the Last Major Leap solution (LML). LL and LML work well on the data sets in which the clusters are

Table 1. Notions.

Notions	Description
P	Data set
C	Center set
M	Data set size
K	Number of clusters
k_{max}	Maximum number of clusters
Weight	Density weight
$P_{p_i}^h$	Relative high-density point set
$dist$	Density-bound minimum distance
$score_{d \cdot d}$	Density-distance score
cls	Cluster center closeness
k_{break}	The break value describes the minimum value of k which satisfies that $cls(k) > 1$
f	Calculate the change of minimum distance between center points
$\overline{\rho}$	Density mean of the dataset
k_i	The number of clusters is i

well-separated and have equal sizes and variances. They even do better than most solutions. However, they encounter severe accuracy degradation on overlapping data sets. Many other solutions also have the same problem on overlapping data sets. In this work, we focus on the accuracy problem on overlapping data sets, and try to find a solution for that problem.

4 A Fast Estimation Solution

In this section, we elaborate a fast estimation solution for the number of clusters in overlapping data sets. The minimum distance between the center points is used to measure the change in the degree of separation between clusters. The solution exploits a density-distance model to select cluster centers in a static way. In order to realize the automatic determination of the number of clusters, it is necessary to use the formula to determine the degree of change in distance. However, when the value of k is greater than the optimal value, the following situations are likely to occur. On the whole, the distance between the center points has not changed much at this time. But from a local perspective, it is a big change. This leads to misjudgment. Therefore, we use the tightness of the center point to narrow the K value range to avoid the distance between the center points being too small. Finally, we take the number satisfying the constraint of the minimum distances as the optimal number of clusters.

4.1 Selecting Cluster Centers

Density-based clustering solutions have the ability of selecting global optimal points as cluster centers without iterations. In order to the performance of clustering, Laio et al. proposed a fast clustering algorithm based on the cluster centers selected based on the products of the density and distance of each data point.

Here, the distance of a data point represents the minimum distance between that point and any other point which has higher density than that point. The algorithm has the ability which can select proper cluster centers in non-sphere and strongly overlapping data sets.

In order to avoid the influence of outliers on judging the change in minimum distance between the center points, we introduce the density-distance model designed based on density weight. Density weight is defined to measure the importance of the density of a data point.

Definition 1. ***Density weight.*** *$(\forall p_i)p_i \in P$, the density weight of p_i describe the importance of $\rho(p_i)$ in deciding whether to accept p_i as a cluster center. It is calculated by Function 1.*

$$weight(p_i) = \frac{\rho(p_i)}{\overline{\rho}} \tag{1}$$

Definition 2. ***Relative high-density point set.*** *$(\forall p_i)p_i \in P$, the relative high-density point set of p_i consists of the points which have a higher density than p_i. It is noted as $P^h_{p_i}$, and described as*

$$P^h_{p_i} = \{p_j | \exists (p_j \in P \text{ and } \rho(p_j) > \rho(p_i))\}.$$

Definition 3. ***Density-bound minimum distance.*** *$(\forall p_i)p_i \in P$, the density-bound minimum distance represents the minimum distance from p_i to any point with higher density. It is denoted as $dist_{min\cdot\rho}(p_i)$, and calculated by Function 2, where $dist(p_i, p_k) = (p_i - p_k)^2$.*

$$dist_{min\cdot\rho}(p_i) = min_{(p_k \in P^h_{p_i})} dist(p_i, p_k) \tag{2}$$

Definition 4. ***Density-distance score.*** *$(\forall p_i)p_i \in P$, the density-distance score of p_i is defined to measure whether p_i is suitable for a cluster center. It is denoted as $score_{d\cdot d}(p_i)$.*

The density-distance score is calculated by a density-distance model.

The score can be calculated according to Function 3. Considering Function 1, Function 3 can be transformed into function 4.

$$score_{d\cdot d}(p_i) = \rho(p_i) \cdot weight(p_i) \cdot dist_{(min\cdot\rho)}(p_i) \tag{3}$$

$$score_{d\cdot d}(p_i) = \frac{\rho(p_i)^2}{\overline{\rho}} \cdot dist_{min\cdot\rho}(p_i) \tag{4}$$

The Points with high density-distance scores are more suitable for being cluster centers than the points with low scores. Therefore, the interference of outliers is reduced by assigning lower scores.

To select cluster centers quickly, our approach calculates a density-distance score for each data point, and sorts all these data points in the descending orders of density-distance scores. $(\forall k_i)k_i \in K$, it takes the top k_i data points as the centers of k_i clusters.

4.2 Adjusting the Search Space of the Optimal Number

The aim of our solution is to find the optimal number of the clusters in an overlapping data set from a search space described by $S = \{1, 2, \ldots, k_{max}\}$. The size of the search space is decided by k_{max}. If k_{max} is much larger than the optimal number of clusters, some relatively close data points are probably selected as cluster centers. Those data points lead to the misjudgement on the significant changes of the minimum distances between cluster centers. This results in the erroneous estimation of the optimal number of clusters. Furthermore, the too large value of k_{max} increases additional search cost.

To avoid the erroneous estimation and the additional search cost, we introduce the definition of cluster center closeness. Cluster center closeness is introduced to assist the proper assignment of k_{max}. It describes the tightness among cluster centers. The lower value of the closeness indicates the sparser distribution of the cluster centers, and vice versa.

Definition 5. ***Cluster center closeness****. The cluster center closeness of k clusters describes the adjacency of those centers. It is noted as cls(k) and calculated by Function 5.*

$$cls(k) = \frac{dist_{m \cdot f \cdot c}(\overline{dist}, count_{\rho<\overline{\rho}})}{\min\limits_{c_i \in C_k, c_j \in C_k \text{ and } i \neq j} dist(c_i, c_j)} \tag{5}$$

In Function 5, $dist_{m \cdot f \cdot c}(\overline{dist}, count_{\rho<\overline{\rho}})$ describes the minimum distance of the centers. It is calculated by Function 6, where $count_{\rho<\overline{\rho}}$ denotes the total number of points of which the densities are smaller than the average density.

$$dist_{m \cdot f \cdot c}(\overline{dist}, count_{\rho<\overline{\rho}}) = \frac{\overline{dist} \cdot count_{\rho<\overline{\rho}}}{m} \tag{6}$$

The case that $cls(k_i)$ exceeds 1 indicates that some centers of the k_i clusters from the same cluster. This means that k_i overpasses the optimal number of clusters. In this situation, it is unnecessary to search the optimal number from k_i to k_{max}. Here, we define break value to describe k_i in this situation.

Definition 6. ***Break value****. The break value describes the minimum value of k which satisfies that $cls(k) > 1$. It is denoted as k_{break} and calculated by Function 7*

$$k_{break} = \min_{k_i \in K \text{ and } cls(k_i) > 1} k_i \tag{7}$$

To improve performance and avoid misjudgement, it is necessary to exclude the values from k_{break} to k_{max} from the search space for the optimal number of clusters. The new search space is described as $S' = \{k_i \mid k_i \in N^* \text{and } k_i < k_{break}\}$. Consequently, the cluster center set is also adjusted to be consistent with the change of the search space. Actually, it is narrowed to only include $(k_{break} - 1)$ points with the highest density-distance scores.

4.3 Identifying the Optimal Number of Clusters

Cluster centers are selected only according to density-distance scores. Concretely, The top k data points with the highest scores are taken as the centers of k clusters, while the top $k+1$ data points with the highest scores are taken as the centers of $k+1$ clusters. Therefore, cluster centers can be selected ahead. Correspondingly, the minimum distance between cluster centers is fixed for a given number of clusters.

Based on the observation of Gupta et al., the significant change of the minimum distance between cluster centers is exploited to identify the optimal number of clusters. Furthermore, the optimal number is identified according to the last significant change. Here, we utilize Function 8 to discover any significant change.

$$f(C_k, C_{k+1}) = \frac{\min\limits_{c_i \in C_k, c_j \in C_k \text{ and } i \neq j} dist(c_i, c_j)}{\min\limits_{c_i \in C_{k+1}, c_j \in C_{k+1} \text{ and } i \neq j} dist(c_i, c_j)} \tag{8}$$

If $f(C_k, C_{k+1})$ reaches up to or overpass a predefined threshold, the significant change of the minimum distances occurs when k increases to $(k+1)$. All the significant changes of the minimum distances can be discovered by calculating the values of $f(C_k, C_{k+1})$ when the number of cluster increases from 1 to k_{break}. The number related to the last significant change of the minimum distances is taken as the optimal number of clusters.

The framework to estimate the optimal number of clusters is described as following:

Step 1. $k_{max} \leftarrow \sqrt{m}$;

Step 2. *Calculate a density-distance score for each data point according to Function 4;*

Step 3. *Select the top K_{max} points with the highest density-distance scores as centers of K_{max} clusters;*

Step 4. *Calculate a $cls(k)$ for each k from 1 to k_{max} according to Function 5;*

Step 5. *Calculate k_{break}, and reconstruct the search space of the optimal number as S';*

Step 6. *Calculate all the significant change of the minimum distances between cluster centers according to S' and Function 8;*

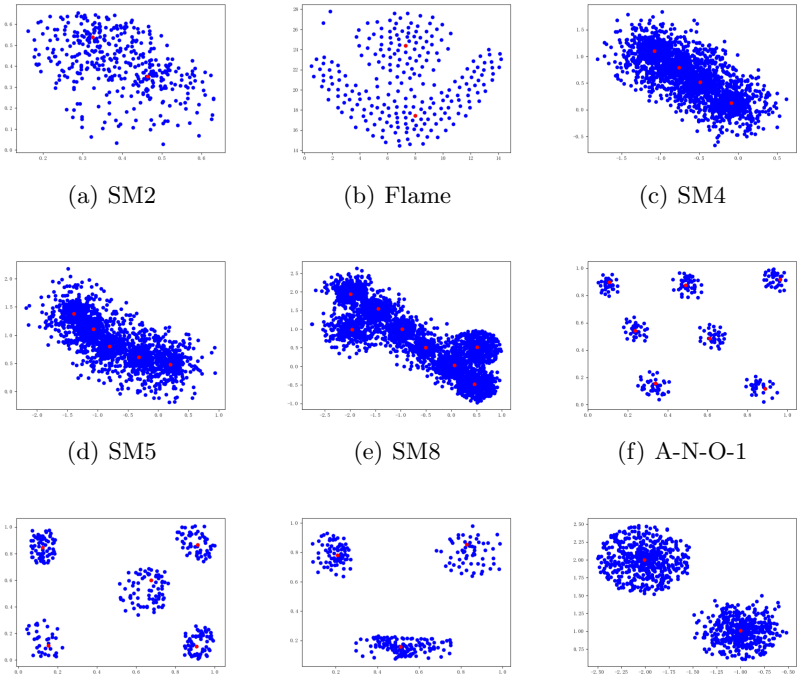
Step 7. *Take the element related to the last significant change of the minimum distances in S' as the optimal number.*

5 Experiments and Discussion

In this section, we present extensive experiments on artificial data sets and real data sets to evaluate our approach. In the experiments, we compare our approach with ten different solutions. Before going into details, we introduce the data sets and performance metrics used in the experiments.

5.1 Data Sets and Metrics

To evaluate our approach, we carried out extensive experiments on thirteen data sets. This includes nine overlapping data sets and four non-overlapping data sets. The overlapping data sets are SM2, Flame, SM5, SM6, SM8, Wine, Seeds, Iris and Ionoshpere. Five of those data sets are artificial data sets and the rest of the data sets are real data sets. There are also four non-overlapping data sets, namely A-N-O-1, A-N-O-2, A-N-O-3 and A-N-O-4. The details of these data sets are described in Table 2. Nine artificial data sets are shown in Fig. 3.



(a) SM2 (b) Flame (c) SM4

(d) SM5 (e) SM8 (f) A-N-O-1

(g) A-N-O-2 (h) A-N-O-3 (i) A-N-O-4

Fig. 3. Distribution of data sets

Accuracy and execution time are adopted as two metrics used to evaluate our approach. Accuracy is exploited to measure the effectiveness of our approach while execution time is utilized to measure the performance.

Table 2. Details of dataset.

Number	Dataset	Features	Clusters	Instances
1	SM2	2	2	403
2	Flame	2	2	240
3	SM4	2	4	3200
4	SM5	2	5	2600
5	SM8	2	8	4400
6	Iris	4	3	150
7	Seeds	7	3	210
8	Wine	13	3	178
9	Ionoshpere	34	2	351
10	A-N-O-1	2	7	264
11	A-N-O-2	2	5	306
12	A-N-O-3	2	3	280
13	A-N-O-4	2	2	1100

In evaluation, our approach is compared with ten different approaches showed in Table 3. All those approaches are executed 20 times on all of the data sets and all the results discussed in the rest of this section are the average results of the 20 executions.

Table 3. Approaches to be compared with ours

Approaches	Selection criteria for k	Min no. of clusters
PC [28]	max	2
ZXF [30]	Knee	2
LL	max & 1	1
LML	max & 1	1
I Index [8]	max	2
BIC	max	2
CH Index [24]	max	2
CE [25]	min	2
FHV [26]	min	1
Jump [27]	max	2
Our approach	max & 1	1

5.2 Experiment Results and Analysis

We track the 20 executions of each approach on the overlapping data sets and record the estimated number of clusters in Table 4. If an approach gets two different numbers of clusters in a data set during 20 executions, the results are record as the two number separated by a slash. If an approach obtains multiple numbers of clusters, the results are described as two numbers connected by a dash. One of the two number is the obtained minimal number, and the other is the obtained maximal number.

According to the Table 4, LML estimates the number of clusters correctly only on the Iris data set. BIC does better than LML. It obtains the correct estimation on SM2 and SM8 data sets. Jump and I also exhibit relatively high accu-

Table 4. Estimation of the optimal number of clusters in overlapping data sets

Approaches	Iris	Seeds	Wine	Ionoshpere	SM2	Flame	SM4	SM5	SM8
PC	2	2	2	2	2	2	2	2	2
ZXF	6	6	6	4–7	6	4	10/11	11–14	9/11
LL	2	2	2	1–9	1	4	1/2	2	8
LML	3	2	2	1–9	1	4	3/55	2	8–12
I	3	3	7	2	2	4	3	3	2
BIC	8–12	14	13	15–18	2	4	3	3/15	8
CH	3	3	13	2	2	8	2	2	8
CE	2	2	2	2	2	2	2	2	2
FHV	2	1	1	1	1	1	1	1	2
Jump	3	3	10–13	2	12–18	4	2–56	2–50	8
Our approach	3	3	3	2	2	2	4	5	8

Table 5. Estimate of the optimal number of clusters in non-overlapping data sets

Approaches	A-N-O-1	A-N-O-2	A-N-O-3	A-N-O-4
PC	7	5	3	2
ZXF	7	5	3	10/11
LL	7	5	3	2
LML	7	5	3	2–32
I	7	5	3	2
BIC	7	11	8	2
CH	7	11/12	4	2
CE	7	5	3	2
FHV	7	5	3	2
Jump	7	5	3	2
Our approach	7	5	3	2

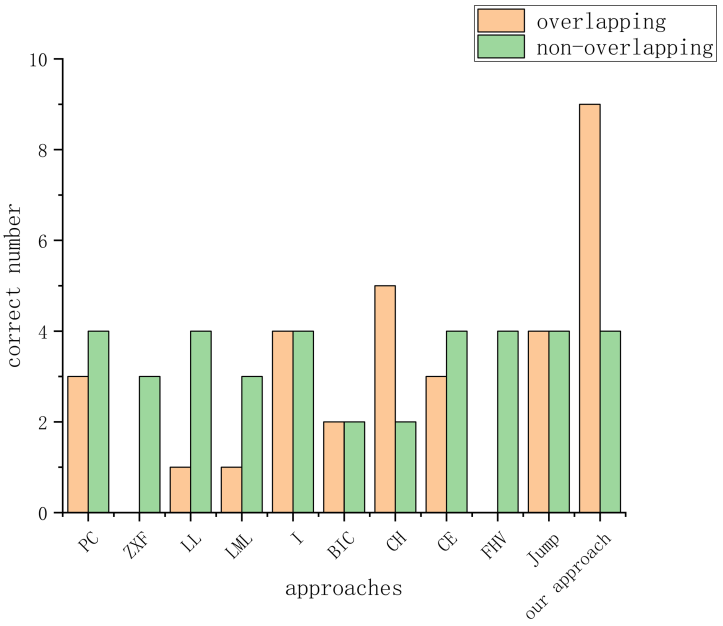


Fig. 4. The correct number of each method on the overlapping data set and the non-overlapping data set.

Table 6. Execution time of approaches

Approaches	A-N-O-1	A-N-O-2	A-N-O-3	A-N-O-4	Iris	Seed	Wine	Ionoshpere	SM2	Flame	SM4	SM5	SM8
PC	2.23	2.40	2.47	18.16	1.37	2.05	1.58	4.17	13.49	2.26	145.44	91.73	279.05
ZXF	2.22	2.30	2.50	18.21	1.37	2.03	1.57	4.13	13.63	2.24	148.9	92.75	259.20
LL	2.25	2.29	2.44	18.11	1.36	2.04	1.61	3.98	13.79	2.26	151.75	92.85	260.87
LML	2.22	2.29	2.44	18.14	1.40	2.04	1.58	3.98	13.67	2.34	152.10	93.35	260.65
I	2.23	2.30	2.45	18.14	1.37	2.13	1.58	4.06	13.74	2.25	153.12	94.22	261.63
BIC	2.23	2.28	2.42	18.17	1.35	2.02	1.57	3.96	13.79	2.41	143.81	92.52	261.83
CH	2.21	2.28	2.44	18.15	1.35	2.04	1.56	3.94	13.44	2.41	143.91	93.85	263.74
CE	2.18	2.23	2.39	18.21	1.32	1.99	1.53	3.99	13.72	2.39	144.75	93.40	262.61
FHV	2.23	2.29	2.44	18.28	1.40	2.09	1.62	4.06	13.76	2.29	143.83	93.84	272.38
Jump	2.28	2.34	2.48	18.27	1.39	2.08	1.60	4.05	13.58	2.26	143.19	92.83	269.28
Our approach	0.027	0.033	0.028	0.20	0.015	0.033	0.032	0.066	0.049	0.034	1.03	0.724	1.70

*Each execution time is measured in second.

racy. They estimate the number of clusters correctly on 4 overlapping datasets. They are followed by CE, and PC which estimate the number of clusters correctly on 3 data sets. CH can correctly estimate the number of clusters on the five data sets. Among those solutions, our approach does best. It estimates the number of clusters correctly on all the overlapping data sets.

We also evaluate the accuracy of our approach on non-overlapping data sets, and describe the results in Table 5. According to the table, BIC and CH estimate the number of clusters correctly on two data sets, while LML and ZXF do better.

They correctly estimate the number of clusters on three data sets. PC, LL, I, FHV, CE, Jump and our approach does best. They obtain the results consistent with the natural number of clusters on each of the non-overlapping data sets.

Figure 4 shows the accuracy of each method on overlapping and non-overlapping data sets. Based on the estimation on both overlapping data sets and non-overlapping data sets, our approach exhibit highest accuracy than all of the other solutions.

In addition to effectiveness, we also evaluate the performance of our approach on all of the data sets. Concretely, we calculated the average execution time of all the approaches and depict the results in Table 6. According to the Table, our approach spends less time on each of the data sets than each of the other approaches does. Our approach spends 0.015 s to 1.7 s on those data sets, while other approaches take 1.32 s to 279.05 s.

Our approach estimates the number of clusters with higher performance than other solutions do. All the other solutions exhibit the similar performance on the same data set.

Our approach exhibits highest performance in all the approaches for two reasons. The first reason is that our approach selects cluster centers in a static way. It calculates a score for each data points. Once a score is calculated, it will not change any more. Our approach takes the top k data points with the highest scores as the centers for k clusters. The other way to choose a cluster center is through iteration. When certain conditions are met, the iteration will stop and the center point will be obtained. The second reason is that our approach narrow the search space of the optimal number of cluster, and hence degrading search cost.

6 Conclusion

In this work, we focused on the problem to estimate the optimal number of clusters in overlapping data sets. To deal with the problem, we proposed a fast estimation approach. The approach selects cluster centers in a static way according to density and distance. It utilizes the significant change of the minimal distance between cluster centers to identify the optimal number of clusters. The experimental result demonstrated the usefulness and effectiveness of our approach. In the future, we will conduct research on estimating the optimal number of the clusters in more complex data sets.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (61602156 and 61433012), the project of the Scientific and Technological in Henan province (172102310677), the project of the Basic and Frontier Technology in Henan province (142300 410147) and the PhD foundation of Henan Polytechnic university (B2012-099).

References

1. Anil, K.: Data clustering: 50 years beyond K-means. Pattern Recogn. Lett. **31**(8), 651–666 (2010)

2. He, Z., Jia, Z., Zhang, X.: A fast method for estimating the number of clusters based on score and the minimum distance of the center point. Information **11**, 16 (2020)
3. Chen, Z.W., Chang, D.X.: Automatic clustering algorithm base on density difference. J. Softw. **29**(4), 935–944 (2018)
4. Jia, R.Y., Li, Z.: The level of K-means clustering algorithm base on minimum spanning tree. Microelectron. Comput. **33**(3), 86–93 (2016)
5. Ünlü, R., Xanthopoulos, P.: Estimating the number of clusters in a dataset via consensus clustering. Expert Syst. Appl. **125**, 33–39 (2019)
6. Bai, L., Cheng, X., Liang, J., Shen, H., Guo, Y.: Fast density clustering strategies based on the k-means algorithm. Pattern Recogn. **71**, 375–386 (2017)
7. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE Trans. Pattern Anal. Mach. Intell. **1**(2), 224–227 (1979)
8. Maulik, U., Bandyopadhyay, S.: Performance evaluation of some clustering algorithms and validity indices. IEEE Trans. Pattern Anal. Mach. Intell. **24**(12), 1650–1654 (2002)
9. Beni, G., Xie, X.: A validity measure for fuzzy clustering. IEEE Trans. Pattern Anal. Mach. Intell. **13**(8), 841–847 (1991)
10. Rodriguez, A., Laio, A.: Machine learning clustering by fast search and find of density peaks. Science **344**(619), 1492 (2014)
11. Gupta, A., Datta, S., Das, S.: Fast automatic estimation of the number of clusters from the minimum inter-center distance for k-means clustering. Pattern Recogn. Lett. **116**, 72–79 (2018)
12. He, L., Wu, L.D., Cai, Y.C.: Survey of clustering algorithms in data mining. Appl. Res. Comput. **71**, 375–386 (2017)
13. Macqueen, J.: Some methods for classification and analysis of multivariate observations. In: Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, no. 14, pp. 281–297. California Press, Berkely (1967)
14. Zhai, D.H., Yu, J., Gao, F.: K-means text clustering algorithm based on initial cluster centers selection according to maximum distance. Appl. Res. Comput. **31**(3), 713–719 (2014)
15. de Amorim, R.C., Hennig, C.: Recovering the number of clusters in data sets with noise features using feature rescaling factors. Inf. Sci. **324**, 126–145 (2015)
16. Teklehaymanot, F.K., Muma, M., Zoubir, A.M.: A novel Bayesian cluster enumeration criterion for unsupervised learning. IEEE Trans. Signal Process **66**(20), 5392–5406 (2018)
17. Bensaid, A.M., Hall, L.O., Bezdek, J.C.: Validity-guided (re)clustering with applications to image segmentation. IEEE Trans. Fuzzy Syst. **4**, 112–123 (1996)
18. Ren, M., Liu, P., Wang, Z., Yi, J.: A self-adaptive fuzzy c-means algorithm for determining the optimal number of clusters. Comput. Intell. Neurosci. 3–15 (2016)
19. Sweeney, T.E., Chen, A.C., Gevaert, O.: Combined mapping of multiple clustering algorithms (COMMUNAL): a robust method for selection of cluster number, K. Sci. Rep. **5**, 16971 (2015)
20. Wang, M., Abrams, Z.B., Kornblau, S.M.: Thresher: determining the number of clusters while removing outliers. BMC Bioinformatics **19**(1), 9 (2018)
21. Kingrani, S.K., Levene, M., Zhang, D.: Estimating the number of clusters using diversity. Artif. Intell. Res. **7**(1), 15 (2018)
22. Doan, H., Nguyen, D.: A method for finding the appropriate number of clusters. Int. Arab J. Inf. Technol. **15**(4), 675–682 (2018)
23. Wang, Y., Shi, Z., Guo, X., Liu, X., Zhu, E., Yin, J.: Deep embedding for determining the number of clusters. In: AAAI (2018)

24. Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. Commun. Stat. **3**(1), 1–27 (1974)
25. Bezdek, J.C.: Mathematical models for systematics and taxonomy. In: Eighth International Conference on Numerical Taxonomy, vol. 3, pp. 143–166 (1975)
26. Dave, R.N.: Validating fuzzy partitions obtained through c-shells clustering. Pattern Recognit. Lett. **17**(6), 613–623 (1996)
27. Sugar, C.A., James, G.M.: Finding the number of clusters in a dataset: an information-theoretic approach. J. Am. Stat. Assoc. **98**(463), 750–763 (2003)
28. Bezdek, J.C.: Cluster validity with fuzzy sets. J. Cybernet. **3**(3), 58–73 (1973)
29. Pakhira, M.K., Bandyopadhyay, S., Maulik, U.: Validity index for crisp and fuzzy clusters. Pattern Recognit. **37**(3), 487–501 (2004)
30. Zhao, Q., Xu, M., Fränti, P.: Sum-of-squares based cluster validity index and significance analysis. In: Kolehmainen, M., Toivanen, P., Beliczynski, B. (eds.) ICANNGA 2009. LNCS, vol. 5495, pp. 313–322. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04921-7_32