A Quantitative Discriminant Method of Elbow Point for the Optimal Number of Clusters in Clustering Algorithm

Congming Shi[1], Bingtao Wei[2,3], Shoulin Wei[2,3,*], Wen Wang[2,3], Hai Liu[1], Jialei Liu[1]

[1] School of Software Engineering, Anyang Normal University, China
[2] Faculty of Information Engineering and Automation, Kunming University of Science and Technology, China
[3] Computer Technology Application Key Lab of Yunnan Province, Kunming University of Science and Technology, China

{shicongming@astrolab.cn, weibingtao@cnlab.net, weishoulin@kust.edu.cn, wangwen@cnlab.net, 01740@aynu.edu.cn, 01850@aynu.edu.cn }

Abstract: Clustering, as a traditional machine learning method, is still playing a significant role in data analysis. The most of clustering algorithms depend on a predetermined exact number of clusters, whereas, in practice, clusters are usually unpredictable. Although elbow method is one of the most commonly used methods to discriminate the optimal cluster number, the discriminant of the number of clusters depends on manual identification of the elbow points on the visualization curve, which will lead to the experienced analysts not being able to clearly identify the elbow point from the plotted curve when the plotted curve being fairly smooth. To solve this problem, a new elbow point discriminant method is proposed to work out a statistical metric estimating an optimal cluster number when clustering on a dataset. Firstly, the average degree of distortion obtained by Elbow method is normalized to the range of 0 to10; Secondly, the normalized results are used to calculate Cosine of intersection angles between elbow points; Thirdly, the above calculated Cosine of intersection angles and Arccosine theorem are used to compute the intersection angles between elbow points; Finally, the index of the above computed minimal intersection angles between elbow points is used as the estimated potential optimal cluster number. The experimental results based on simulated datasets and a public well-known dataset (Iris Dataset) demonstrated that the estimated optimal cluster number output by our newly proposed method is better than widely used Silhouette method.

Keywords: Machine Learning, Clustering, Elbow Method, Silhouette Coefficient, Cosine Law

1. Introduction

In terms of machine learning, clustering, as a common technique for statistical data analysis, has been widely used in a large amount of fields, and holds an important status in unsupervised learning. Data analysts can use the clustering to exploit the potential optimal cluster number for the analyzed dataset containing similar characteristics. The area of clustering had produced various implementations over the past decade or so. An exhaustive list refers to[1]. Nevertheless, determining the optimal cluster number is

always the difficult part, especially for a dataset with little prior knowledge. A fair percentage of partitional clustering algorithm(e.g., K-means[2], K-medoids[3], PAM[4]) need to specify the cluster number as the input parameter in advance of training. Hierarchical clustering(e.g., BIRCH[5], CURE[6], and ROCK[7]), and clustering algorithm basing on fuzzy theory(e.g., FCM[8], FCS[9] and MM[10].), also have disadvantages in the number of clusters needed to be preset.

Furthermore, estimating the potential optimal cluster number for the analyzed dataset is a fundamental issue in clustering algorithms. With little prior information of the properties of a dataset, there are still a few methods to evaluate the potential optimal cluster number. As an oldest visual method for estimating the potential optimal cluster number for the analyzed dataset, the elbow method [11; 12] usually needs perform the K-means on the same dataset with a contiguous cluster number range:[1, L] (L is an integer greater than 1), then, compute the Sum of Squared Errors (SSE) for each user-specified cluster number k, plotting a curve of the SSE against each cluster number k. Finally, the experienced analysts estimate the optimum elbow point by analyzing the above mentioned curve, i.e., the optimum elbow point corresponds to the estimated potential optimal cluster number with high probability. However, when the relationship curve of the SSE against each value of k is a fairly smooth curve, the experienced analysts cannot clearly identify the 'elbow' from the plotted curve. That is, the elbow method does not always work well for determining the optimal cluster number[13]. The cluster number obtained by using the elbow method is a subjective result since it is a visual method[14], and it does not give a measurement metric to show which elbow point is the optimal explicitly. To overcome these shortcomings of the elbow method, a quantitative discriminant method is proposed to work out a straightforward value as the estimated potential optimal cluster number for the analyzed dataset. Our newly proposed method is based on Elbow method[15], K-means++[16], MinMaxScaler [17] for normalization, and cosine of interaction angle of elbow as criteria.

In the rest of sections, a brief overview of the related work is introduced in section 2. Our newly proposed method will be introduced in section 3. In section 4, the simulated datasets and a common benchmark dataset are used to test and verify the validity of our newly proposed method. Finally, it provides conclusions and discussion are given in section 5.

2. Related Work

There is a major challenge that how to obtain the optimal cluster number in cluster analysis. The potential optimal cluster number needs to be given in advance for the partitioning clustering algorithms, which is an important input parameter. In some cases, there exists enough priori information regarding the dataset, so the potential appropriate cluster number can be intuitively assigned for these partitioning clustering algorithms; however, in general, there does not exist enough priori information to determine an appropriate cluster number which can be specified in advance for the value of an important input parameter of these partitioning clustering algorithms. Therefore, it is

necessary to estimate a potential optimal cluster number for the dataset to be analyzed. In the case of an unknown number of clusters, the first step is usually to specify a potential estimated range for the optimal cluster number in almost all of methods to distinguish the optimal cluster number.

There are many methods used to determine the cluster number of analyzed dataset[18]. Elbow method and Silhouette method are the two state-of-the-art methods used to identify the correct cluster number in dataset[19]. The called elbow method [13] is the oldest method to distinguish the potential optimal cluster number for the analyzed dataset, whose basic idea is that specify the $K = 2$ as the initial optimal cluster number K, then keep increasing the K by step 1 to the maximal specified for the estimated potential optimal cluster number, finally distinguish the potential optimal cluster number K corresponding to the plateau. The optimal cluster number K distinguished is that before reaching K, the cost rapidly decreases to the called cost peak value, and after exceeding K, it continues to increase with the called cost peak value almost unchanged, as is shown in Fig.1(a) with an explicit elbow point. Meanwhile, the optimal cluster number corresponding to the elbow point depends on manmade selection. However, there is a problem with the elbow method that the elbow point cannot be unambiguously distinguished by the experienced analysts when the plotted curve being fairly smooth, as is shown in Fig.1(b) with an ambiguous elbow point.

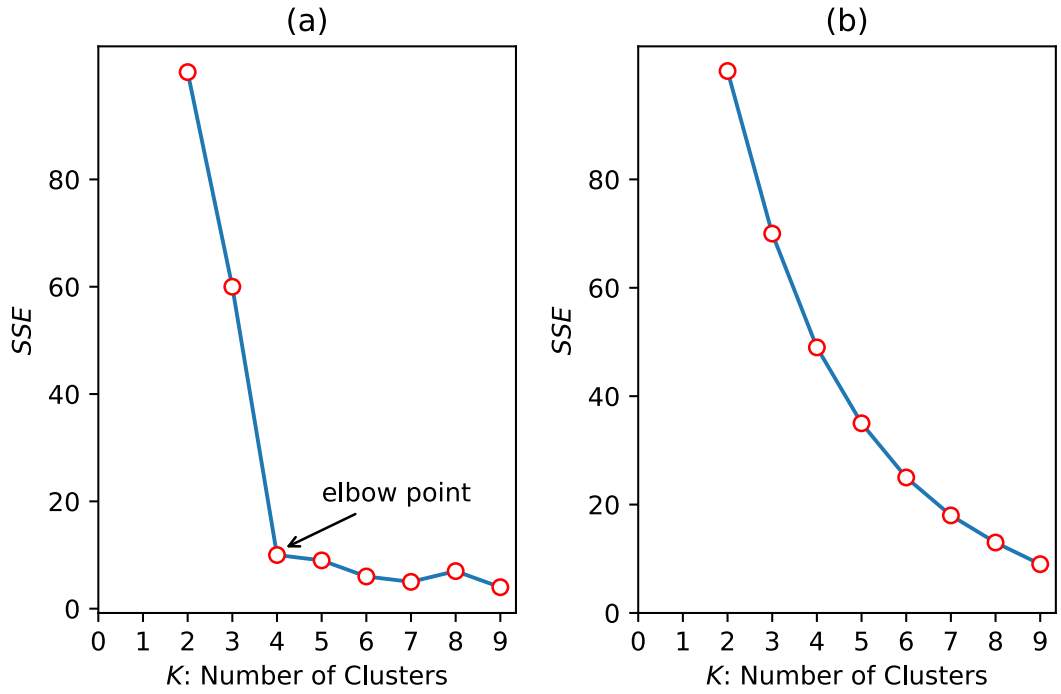

Fig.1 (a) A visual curve with an explicit elbow point. (b) A visual curve being fairly smooth with an ambiguous elbow point.

Silhouette method[20; 21] is another well-known method with decent performance

to be estimated the potential optimal cluster number, which uses the average distance between one data point and others in the same cluster and the average distance among different clusters to score the clustering result. The metric of scoring of this method is named silhouette coefficient (*S*), and *S* is defined as $\frac{(b-a)}{max(a,b)}$, where a and b respectively represent the mean intra-cluster distance and the mean nearest-cluster distance. The interval of the *S* values is $-1 \leq S \leq 1$. The value of *S* closer to 1 indicates that a sample is better clustered, and if closer to -1, the sample should be categorized into another cluster. And this method is preferable for estimating the potential optimal cluster number. Meanwhile, the Silhouette index can evaluate the best number of clusters in most cases for many distinct scenarios[22].

Meanwhile, the gap statistic methods can be used to identify the optimal cluster number in the analyzed dataset. The gap statistic method [23; 24] obtains the potential optimal cluster number through the following steps: it obtains the output of *K*-means, compares the change in within-cluster dispersion, and get the appropriate cluster number. The hierarchical agglomerative clustering (HAC[25]) usually performs the *K*-means *N* times, obtains the dendrogram, and gets the potential optimal cluster number[26]. And the *v*-fold cross-validation method [27] is an approach to estimate the most appropriate cluster number depending on the *K*-means[28] clustering algorithm or expectation maximum (EM). At the same time, there are some methods based on information criteria, which can also be used to score the most appropriate cluster number[29]. For example, the Akaike information criterion (AIC) or Bayesian information criterion (BIC) is used in the *X*-means clustering to discriminate the potential optimal cluster number for the analyzed dataset[30]. The rate distortion theory can be used for estimating the cluster number for a wide range of simulated and real datasets; meanwhile it can identify the underlying structure, which is given a theoretical justification[4]. Smyth [31] presented cross-validation approach to score the potential optimal cluster number depending on the cluster stability. And this approach tends to repeatedly generate similar cluster for the dataset originated from the same data source. That is to say, this approach is stable for the input randomization[14].

However, as above mentioned, when the elbow point is ambiguous, Elbow method will become not very reliable. To overcome shortcomings of Elbow method, we present a new method to calculate an clear metric to indicate Elbow point for the potential optimal cluster number.

3. Proposed Method

3.1 Principle

Given a dataset *X* with *N* points and *K* clusters, we define $X = \{x_1, x_2, \cdots, x_N\}$ and $\mathbb{C} = \{C_1, C_2, \cdots, C_K\}$ where C_i represents the ith individual cluster and $K \leq N$. The centroids of *K*-clustering of *X* are defined as $\{\mu_1, \mu_2, \cdots, \mu_K\}$ where $\mu_k = \frac{1}{|C_k|}\sum_{x_i \in C_k} x_i$, μ_k is the centroids corresponding to the cluster C_k, $k \in [1,2,\cdots,K]$, and $|C_k|$ represents the number of entries of the cluster C_k.

In theory, data points that are in the same cluster should have maximal similarity, while data points in different clusters should have highly dissimilar properties and/or features relatively. Meanwhile, the similarity between different data objects is measured by the distance between them. Furthermore, the sum of the squared Euclidean distances abbreviated as *SSE* is one of the most widely used cluster distance criteria, which is used to measure the sum of the square distance between each data point belonging to the same cluster and its cluster centroid $\{\mu_1, \mu_2, \cdots, \mu_K\}$:

$$SSE = \sum_{k=1}^{K} \sum_{x_i=C_k} \|x_i - \mu_k\|_2^2 \quad (1)$$

The result of *SSE* divided by *N* is called the mean distortion (*MD*) of the dataset *X* of *N* points, given by:

$$MD_i = \frac{SSE}{N} = \frac{\sum_{k=1}^{K} \sum_{x_i=C_k} \|x_i-\mu_k\|_2^2}{N} \quad (2)$$

Note that MD_i represents the mean distortion of the dataset *X* of *N* points, which has *i* as the cluster number for the analyzed dataset *X*. Meanwhile, the clustering error is usually quantified using *SSE*. Furthermore, we transform each *MD* by MinMaxScaler and scale each tranformed value $N_{(md)}$ to a given range[0-10], which is an empirical value. We consider the complete normalized *SSE* data space as $N = [n_1, n_2, \cdots, n_k]$ in which n_i has the definition:

$$n_i = \frac{MD_i - MD_{(min)}}{MD_{(max)} - MD_{(min)}} * 10 \quad (3)$$

For two given adjacent two-dimensional data points *i* and *j* where $i, j \in [1,2, \cdots, K]$ and *n, k* respectively represents the first dimension and the second dimension of two-dimensional data point, we can use the formula (4) to calculate the Euclidean distance between them.

$$E_{ij} = \sqrt{\left(n_i - n_j\right)^2 + \left(k_i - k_j\right)^2} \quad (4)$$

Let us assume that the three adjacent two-dimensional data point $i, j, k \in [1,2, \cdots, K]$ create an angle $\angle\alpha_j$, and we can obtain the value of the angle $\angle\alpha_j$ with the formula (5).

$$\alpha_j = arccos \frac{E_{ij}^2 + E_{jk}^2 - E_{ik}^2}{2E_{ij}E_{jk}} \quad (5)$$

In the space of $\alpha \in [\alpha_1, \alpha_2, \cdots, \alpha_{k-2}]$, we use the smallest α indicating the optimum elbow point corresponding to the estimated potential optimal cluster number with high probability, and call the index of minimal α as K_{opt} which is considered as the estimated optimal cluster number for the analyzed dataset.

3.2 Implementation

For a given dataset *X* with *N* points, the estimated optimal cluster number is defined as K_{opt}. The first step is to initialize an estimated range of K_{opt} as $[K_{min}, K_{max}]$. Note that the value of K_{min} and K_{max} is 1 and $int(\sqrt{n}) + 1$ respectively ($int()$ means taking the integer portion). The second step is to compute the sum of squared errors (*SSE*) with the formula (1), mean distortion by the formula

(2) and normalized value with the formula (3), for each value of $k \in [K_{min}, K_{max}]$. The procedure can be described in algorithm 1.

We make additional comments about algorithm 1 as follows:

• Line 3: We take k as the input parameters, fit the dataset X to K-means++ and start training.

• Line 4-5: After training, we assign the centroids and the related sub-clusters to μ and $\mathbb{C}$.

• Line 6-8: This is the computation of normalized mean distortion value for each value of k, and appending to $N_{(md)}$.

Algorithm 1 computing the normalized value for each k

1: Initialize an empty list $N_{(md)} \leftarrow [\,]$
2: **for** $k = K_{min} \rightarrow K_{max}$ **do**
3: $k, X \rightarrow$ *k-means++* and train
4: $\mu \leftarrow \{\mu_1, \mu_2, \cdots, \mu_k\}$
5: $\mathbb{C} \leftarrow \{C_1, C_2, \cdots, C_k\}$
6: Compute *SSE* based on $\mathbb{C}$ and μ by the formula (1)
7: $MD = \frac{SSE}{N}$
8: Compute the normalized value $N_{(md)k}$ by the formula (3)
9: Append the $N_{(md)k}$ into $N_{(md)}$
10: **end for**
11: return $N_{(md)}$

Then, we can use the formula (4) to calculate Euclidean distance between two adjacent points, and find K_{opt} with the formula (5). The algorithm of our method can be described in detail as algorithm 2.

Algorithm 2 estimating the potential optimal cluster number for the analyzed dataset

1: $\alpha_{min} = \pi, K_{opt} = 0$
2: $PL \leftarrow$ zip the $N_{(md)}$ and $[K_{min}, K_{max}]$
3: **for** $i = 0 \rightarrow K_{max} - K_{min} - 2$ **do**
4: $j \leftarrow i + 1, k \leftarrow i + 2$
5: $P_i \leftarrow PL[i], P_j \leftarrow PL[j], P_k \leftarrow PL[k]$
6: $a \leftarrow$ *Euclidean* distance between P_i and P_j
7: $b \leftarrow$ *Euclidean* distance between P_j and P_k
8: $c \leftarrow$ *Euclidean* distance between P_k and P_i
9: $\angle P_i P_j P_k = \alpha \leftarrow arccos \frac{a^2 + b^2 - c^2}{2ab}$
10: **if** $\alpha < \alpha_{min}$ **then**
11: $\alpha_{min} = \alpha$
12: $K_{opt} = j$
13: **end if**
14: **end for**
15: return α_{min}, K_{opt}

The following additional comments can be made about algorithm 2:

• Line 2: For convenience of calculations, we zip $N_{(md)}$ and $[K_{min}, K_{max}]$, and form a list of two-dimensional data point pairs, PL, i.e., $PL = \{\langle N_{(md)0}, K_{min}\rangle, \langle N_{(md)1}, K_{min}+1\rangle, \cdots\}$

• Line 4-5: Considered that the three adjacent points, where the point P_i is $\langle N_{(md)i}, K_i\rangle$, P_j is $\langle N_{(md)j}, K_j\rangle$, and P_k is $\langle N_{(md)k}, K_k\rangle$.

• Line 6-8: Compute the *Euclidean* distance between P_i, P_j and P_k and represented by *a, b* and *c*.

• Line 9: Compute the angle which is formed by every three adjacent two-dimensional data point pairs in *PL* by the formula (5).

• Line 10-13: Find the minimal angle (α_{min}), and the index of optimal cluster number as K_{opt}.

4. Performance Evaluation

There are four experiments to test and verify the validity of our proposed method on the test datasets including two kinds of datasets, which are the simulated dataset and a public benchmark dataset with Iris Dataset[32]. Experimental results based on these datasets are plotted in a figure with three subplots, namely, Scatter, Silhouette, and Proposed Method. Scatter is the scatter plot of the experimental dataset, Silhouette is the plot of the silhouette coefficient score and the corresponding cluster number, and Proposed Method is the plot of cluster number and the corresponding α produced by our proposed method.

The above mentioned experiments were tested on a server with 12 Xeon CPUs E5-2620 v2 @ 2.10 GHz cores, 32 GB memory, 2TB hard disk, and 1 GeForce GTX TITAN X (rev a1). It ran the CentOS version 7.3.1611 (core) with the Python version 2.7.12, Anaconda version 4.2.0 (64-bit), the sklearn[33] version 0.16.0.

4.1 Simulated Datasets

Three simulated datasets are respectively followed uniform distribution, the multivariate normal distribution and mixed the uniform distribution and multivariate normal distribution.

Uniformly distributed dataset (Dataset1) has 2000 two-dimensional points attributed to 4 clusters. It is straightforward to exploit the function of *Numpy, numpy.random.uniform(low, high, size)*. Table 1 lists parameters used by the function uniform.

Table 1 The parameters list of uniformly distributed dataset.

Cluser No	low	high	size
1	0.5	1.5	500
2	2.5	3.5	500
3	4.5	5.5	500
4	6.5	7.5	500

Multivariate normal dataset (Dataset2) also includes 2000 two-dimensional points

and 4 clusters, which are generated by the function of *numpy.random.multivariate_normal(mean,cov,size)*. The parameters used are listed in Table 2.

Table 2 The parameters list used for generating multivariate normal dataset.

Cluser No	mean	cov	size
1	[0.0, 0.0]	[[1.0, 0.0], [0.0, 1.0]]	500
2	[2.0, 2.0]	[[1.0, 0.0], [0.0, 1.0]]	500
3	[4.0, 4.0]	[[1.0, 0.0], [0.0, 1.0]]	500
4	[6.0, 6.0]	[[1.0, 0.0], [0.0, 1.0]]	500

Dataset (Dataset3) mixed uniformly distributed and multivariate normal also includes 2000 two-dimensional points and 4 clusters (2 clusters obeyed uniform distribution and 2 clusters obeyed uniform distribution and two multivariate distribution), which are generated by the function of *numpy.random.uniform* and *numpy.random.multivariate_normal*. The parameters of two cluster obeyed uniform distribution are listed in Table2 as Cluster No 3 and 4. The parameters of two multivariate distribution are listed in Table2 as Cluster No 1 and 2.

Since there are only 4 clusters in the aforementioned simulated datasets, we specify the estimated range of K_{opt} as [1, 10]. The execution time of two methods based on Dataset1, Dataset2 and Dataset2 are listed in Table3.

Experimental results of the simulated dataset followed uniform distribution are plotted in Fig.2. From the subplot of Silhouette in the Fig.2, we known that the estimated potential optimal cluster number obtained by the silhouette method is 4, which is consistent with the real cluster number and corresponds to the maximum of the silhouette score. In addition, as is evident from the subplot of Proposed Method in the Fig.2, the obtained optimal cluster number corresponding to the minimum of the angle is 4, which is also consistent with the real cluster number. Therefore, for the uniform distribution, the proposed method and silhouette method can both obtain the optimal cluster number consistent with the real cluster number.

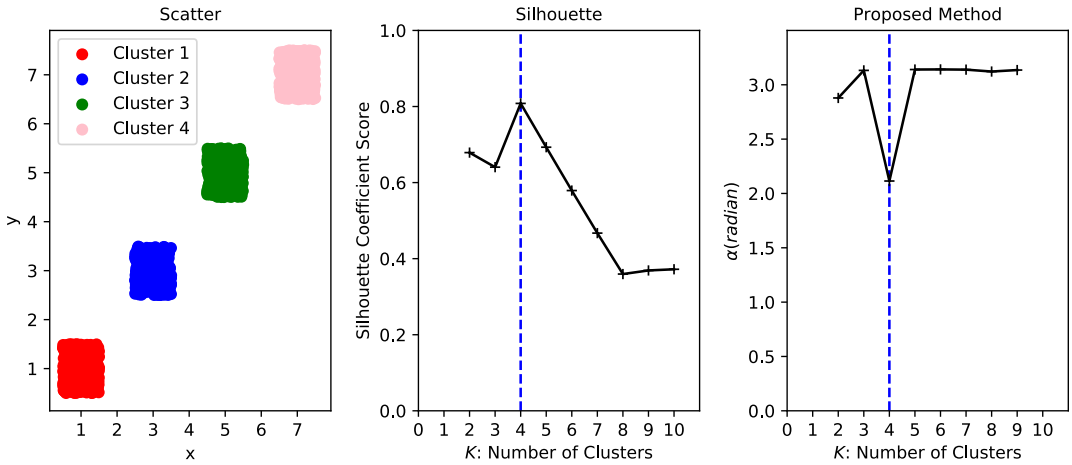

Fig.2 Experimental results of the simulated Dataset1 obeyed uniform distribution with four clusters.

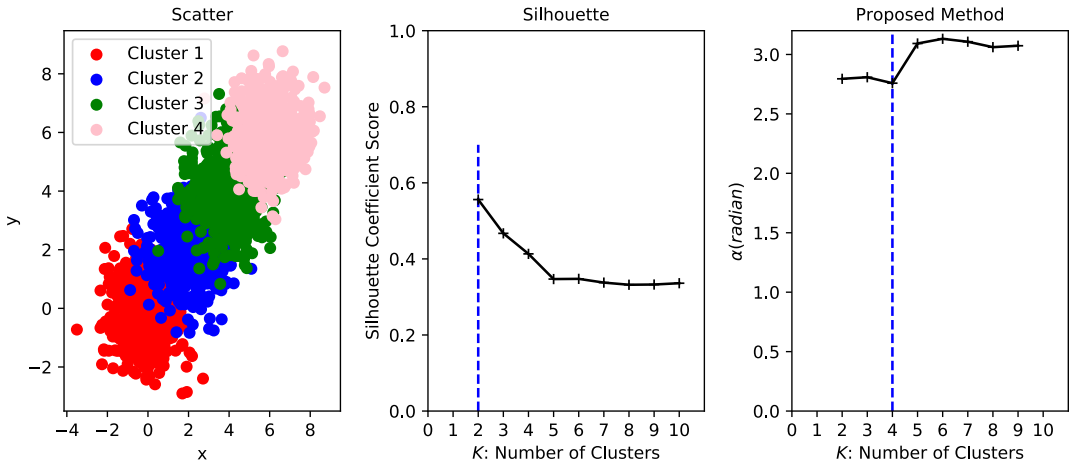

Fig.3 Experimental results of the simulated Dataset2 obeyed multivariate normal distribution with four clusters.

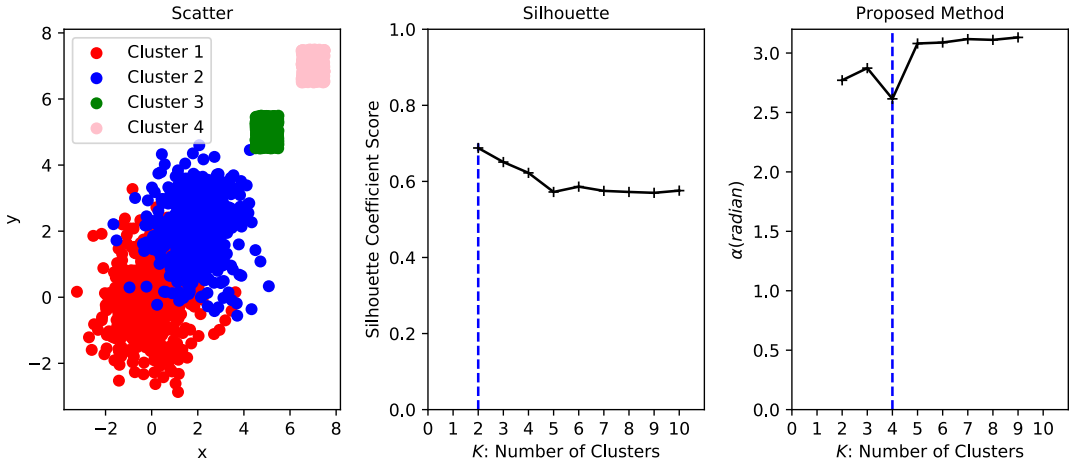

Fig.4 Experimental results of the simulated Dataset3 obeyed uniform and multivariate normal distribution with four clusters.

In aspect of multivariate normal distribution, Proposed Method also work out the same optimal cluster number. Nevertheless, Silhouette does not give the correct cluster number as shown in Fig.3. Furthermore, In aspect of dataset mixed uniform distribution and multivariate normal distribution, Proposed Method also work out the same optimal cluster number. Nevertheless, Silhouette does not give the correct cluster number as shown in Fig.4.

Table 3 The execution time of two methods based on different datasets (unit:second).

Dataset	Execution time of Silhouette	Execution time of Proposed Method
Dataset1	0.974	0.408
Dataset2	1.518	0.950
Dataset3	1.361	0.705
Dataset4	0.327	0.308
Dataset5	0.348	0.318

4.2 Common Benchmark Datasets

For the field of machine learning, Iris[32] is perhaps the best known dataset as common benchmark dataset. It is a multivariate dataset and contains 150 instances with 4 features (sepal length, sepal width, petal length, and petal width) and 3 species (Iris Setosa, Iris Versicolour, and Iris Virginia). In this evaluation, we split Iris dataset into two datasets, namely sepal dataset and petal dataset. The sepal dataset (Dataset4) is comprised of sepal length and sepal width, and the petal dataset (Dataset5) is comprised of petal length and petal width. There are only 3 clusters in Iris. Therefore, we initialize the search range of K_{opt} as [1, 10].

On the two subsets of Iris, as we can see in Fig.5 and Fig.6, Silhouette does not outline the maximal silhouette coefficient score indicated the optimal cluster number. Rather, Proposed Method figures out a clear minimum of the angle at index 3, which is consistent with the real cluster number in Iris. The execution time of two methods based on Dataset4 and Dataset5 are listed in Table3.

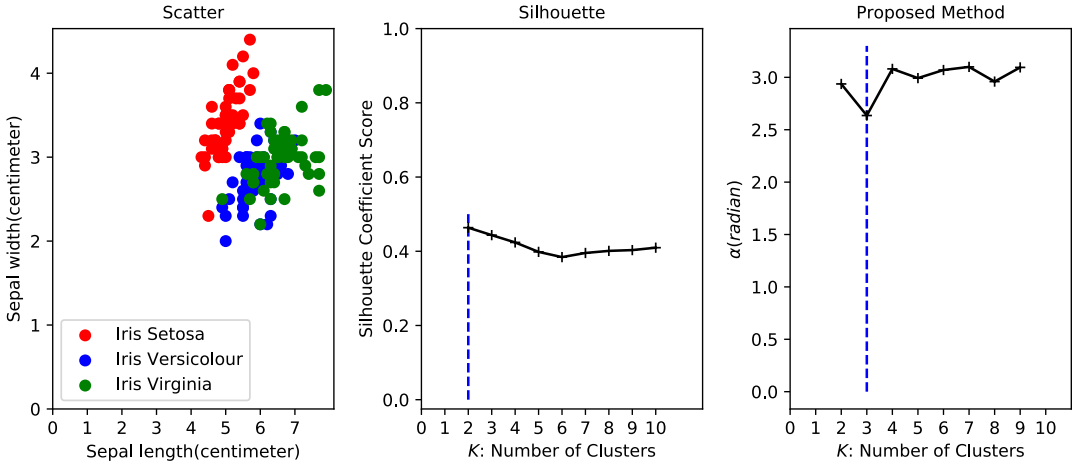

Fig.5 Experimental results on three Iris species only with sepal length and sepal width.

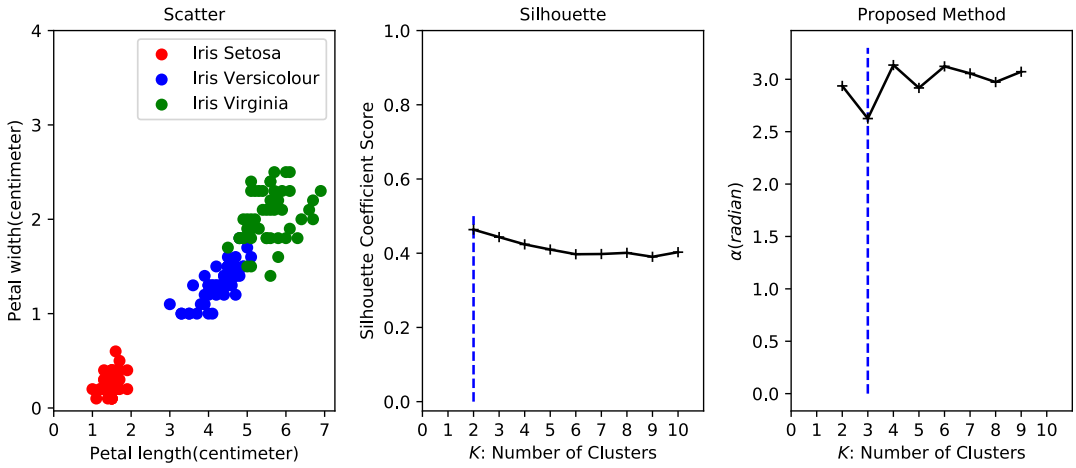

Fig.6 Experimental results on three Iris species only with petal length and petal width.

Therefore, just as shown in the above case, the optimal cluster number obtained by the proposed method is consistent with the real cluster number contained in the

dataset and shows better performance than Silhouette method.

From the table3, we known that the execution time of the proposed method is faster than the Silhouette method for the experimental dataset. Meanwhile, for the figures based on the experimental results, we known that the optimal cluster number obtained by the proposed method is more consistent with the real cluster number contained in the dataset than Silhouette method.

5. Conclusions and Discussion

The elbow method can be as one of the oldest methods to distinguish the potential optimal cluster number for the dataset to be analyzed, which is a visual method. Using the elbow method, the estimated potential optimal cluster number for the analyzed dataset is somewhat subjective. This is because if there exists a clear elbow in the line chart, then, the elbow point corresponds to the estimated optimal cluster number with high probability, whereas, if there is no clear elbow in the line chart, then, the elbow method does not work well.

A new method for distinguished the potential optimal or most appropriate cluster number used in clustering algorithm is proposed in this paper. We exploited interaction angle of adjacent elbow point as criteria to work out a discriminant elbow point. Experimental results demonstrate that the estimated potential optimal cluster number output by our newly proposed method is consistent with the real cluster number and better than Silhouette method on the same experimental datasets.

The proposed method depends on an estimated range of cluster number. For each estimated number of clusters, the entire dataset needs to be trained, and then it will definitely increase the computational cost. The clustering algorithm used in this paper is *K*-means++ which is a centroid-based clustering algorithm. What seems certain is that our method can also be applied to other centroid-based clustering algorithms, e.g., *K*-means and *K*-medoids. Nevertheless, for non-centroid-based clustering algorithms, that may not be the case. How to improve performances and suitability for the proposed method to estimate the potential optimal cluster number will be the focus of our future work.

List of Abbreviations

(1) PAM: Partitioning Around Medoids

(2) BIRCH: Balanced Iterative Reducing and Clustering using Hierarchies

(3) CURE: Clustering Using REpresentatives

(4) ROCK: A Robust Clustering Algorithm for Categorical Attributes

(5) FCM: A Fuzzy C-means Clustering Algorithm

(6) FCS: Fuzzy c-shells

(7) MM: Mountain Method

(8) SSE: Sum of Squared Errors

(9) HAC: Hierarchical Agglomerative Clustering

(10) EM: Expectation Maximum

(11) AIC: Akaike Information Criterion

(12) BIC: Bayesian Information Criterion

(13) MD: Mean Distortion

Availability of data and material

Two simulated datasets are respectively followed uniform distribution and the multivariate normal distribution. The simulated dataset followed uniform distribution is generated by the function of *Numpy, numpy.random.uniform(low, high, size)* with parameters as listed in Table 1. The simulated dataset followed multivariate normal distribution is generated by the function of *numpy.random.multivariate_normal(mean,cov,size)* with parameters as listed in Table 2. The common benchmark dataset is available at https://archive.ics.uci.edu/ml/machine-learning-databases/iris/.

Competing interests

We declare that there is no conflict of interest regarding this submission.

Funding

This work is supported by the National Key Research and Development Program of China (2018YFA0404603), Yunnan Key Research and Development Program(2018IA054), Yunnan Applied Basic Research Project (2017FB001), the Key Science and Technology Program of Henan Province (202102210152), Henan Province Higher Education Teaching Reform Research and Practice Project (2019SJGLX386), and the Research and Cultivation Fund Project of Anyang Normal University (AYNUKPY-2019-24).

Authors' contributions

Congming Shi introduced the idea of this work and finished partial English writing. Shoulin Wei designed the algorithms and finished partial English writing. Bingtao Wei, Wen Wang, and Hai Liu coded to implemented the algorithms and finished partial English writing. Jialei Liu completed the experiments and finished partial English writing.

Acknowledgements

Not applicable.

References

[1] D. Xu, Y. Tian, A Comprehensive Survey of Clustering Algorithms. Annals of Data Science 2 (2):165-193 (2015). doi:https://link.springer.com/article/10.1007/s40745-015-0040-1.

[2] A. K. Jain, Data clustering: 50 years beyond K-means. Pattern Recognition Letters 31 (8):651-666 (2010). doi:https://doi.org/10.1016/j.patrec.2009.09.011.

[3] H.-S. Park, C.-H. Jun, A simple and fast algorithm for K-medoids clustering. Expert Systems with Applications 36 (2, Part 2):3336-3341 (2009). doi:https://doi.org/10.1016/j.eswa.2008.01.039.

[4] C. A. Sugar, G. M. James, Finding the number of clusters in a dataset: An information-theoretic approach. Journal of the American Statistical Association 98 (463):750-763 (2003). doi:https://doi.org/10.1198/016214503000000666.

[5] T. Zhang, R. Ramakrishnan, M. Livny, BIRCH: an efficient data clustering method for very large databases. Paper presented at the Proceedings of the 1996 ACM SIGMOD international conference on Management of data, Montreal, Quebec, Canada, pp 103–114 (1996), doi:https://doi.org/10.1145/235968.233324.

[6] S. Guha, R. Rastogi, K. Shim, Cure: an efficient clustering algorithm for large databases. Information Systems 26 (1):35-58 (2001). doi:https://doi.org/10.1016/S0306-4379(01)00008-4.

[7] S. Guha, R. Rastogi, K. Shim, Rock: A robust clustering algorithm for categorical attributes. Information Systems 25 (5):345-366 (2000). doi:https://doi.org/10.1016/S0306-4379(00)00022-3.

[8] J. C. Bezdek, R. Ehrlich, W. Full, FCM: The fuzzy c-means clustering algorithm. Computers & Geosciences 10 (2-3):191-203 (1984). doi:https://staff.fmi.uvt.ro/~daniela.zaharie/dm2019/EN/projects/biblio/FuzzyCMeans/FCM%20-%20The%20Fuzzy%20c-Means%20Clustering%20Algorithm.pdf.

[9] R. N. Dave, K. Bhaswan, Adaptive fuzzy c-shells clustering and detection of ellipses. IEEE Transactions on Neural Networks 3 (5):643-662 (1992). doi:https://doi.org/10.1109/72.159055.

[10] R. R. Yager, D. P. Filev, Approximate clustering via the mountain method. IEEE Transactions on Systems, Man, and Cybernetics 24 (8):1279-1284 (1994). doi:https://doi.org/10.1109/21.299710.

[11] R. L. Thorndike, Who belongs in the family? Psychometrika 18 (4):267-276 (1953). doi:https://doi.org/10.1007/BF02289263.

[12] M. A. Syakur, B. K. Khotimah, E. M. S. Rochman, B. D. Satoto Integration k-means clustering method and elbow method for identification of the best customer profile cluster. In, 2018. IOP Publishing, p 012017

[13] D. J. Ketchen, C. L. Shook, The application of cluster analysis in strategic management research: an analysis and critique. Strategic Management Journal 17 (6):441-458 (1996). doi: https://doi.org/10.1002/(SICI)1097-0266(199606)17:6<441::AID-SMJ819>3.0.CO;2-G.

[14] T. M. Kodinariya, P. R. Makwana, Review on determining number of Cluster in K-Means Clustering. International Journal 1 (6):90-95 (2013). doi:https://www.researchgate.net/profile/Trupti_Kodinariya/publication/313554124_Review_on_Determining_of_Cluster_in_K-means_Clustering/links/5789fda408ae59aa667931d2/Review-on-Determining-of-Cluster-in-K-means-Clustering.pdf.

[15] D. Marutho, S. H. Handaka, E. Wijaya, Muljono The Determination of Cluster Number at k-Mean Using Elbow Method and Purity Evaluation on Headline News. In: 2018 International Seminar on Application for Technology of Information and Communication, 21-22 Sept. 2018 2018. pp 533-538. doi:https://doi.org/10.1109/ISEMANTIC.2018.8549751.

[16] D. Arthur, S. Vassilvitskii, k-means++: the advantages of careful seeding. Paper presented at the Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, New Orleans, Louisiana, pp 1027–1035 (2007), doi:http://ilpubs.stanford.edu:8090/778/1/2006-13.pdf.

[17] B. Komer, J. Bergstra, C. Eliasmith Hyperopt-sklearn: automatic hyperparameter configuration for scikit-learn. In, 2014. Citeseer, p 50. doi:http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.924.9697&rep=rep1&type=pdf.

[18] E. Hancer, D. Karaboga, A comprehensive survey of traditional, merge-split and evolutionary approaches proposed for determination of cluster number. Swarm and Evolutionary Computation 32:49-67 (2017). doi:https://doi.org/10.1016/j.swevo.2016.06.004.

[19] M. A. Masud, J. Z. Huang, C. Wei, J. Wang, I. Khan, M. Zhong, I-nice: A new approach for identifying the number of clusters and initial cluster centres. Information Sciences 466:129-151 (2018). doi:https://doi.org/10.1016/j.ins.2018.07.034.

[20] M. Z. Rodriguez, C. H. Comin, D. Casanova, O. M. Bruno, D. R. Amancio, L. d. F. Costa, F. A. Rodrigues, Clustering algorithms: A comparative approach. PloS one 14 (1):e0210236 (2019). doi:https://doi.org/10.1371/journal.pone.0210236.

[21] P. J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics 20:53-65 (1987). doi:https://doi.org/10.1016/0377-0427(87)90125-7.

[22] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, I. Perona, An extensive comparative study of cluster validity indices. Pattern Recognition 46 (1):243-256 (2013). doi:https://doi.org/10.1016/j.patcog.2012.07.021.

[23] R. Tibshirani, G. Walther, T. Hastie, Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63 (2):411-423 (2001). doi:https://doi.org/10.1111/1467-9868.00293.

[24] C. Yuan, H. Yang, Research on K-value selection method of K-means clustering algorithm. J—Multidisciplinary Scientific Journal 2 (2):226-235 (2019). doi:https://doi.org/10.3390/j2020016.

[25] H. Liu, L. Fen, J. Jian, L. Chen, Overlapping community discovery algorithm based on hierarchical agglomerative clustering. International Journal of Pattern Recognition and Artificial Intelligence 32 (03):1850008 (2018). doi:

[26] M. Dash, H. Liu, P. Scheuermann, K. L. Tan, Fast hierarchical clustering and its validation. Data & Knowledge Engineering 44 (1):109-138 (2003). doi:https://doi.org/10.1016/S0169-023X(02)00138-6.

[27] P. Burman, A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. Biometrika 76 (3):503-514 (1989). doi:https://doi.org/10.1093/biomet/76.3.503.

[28] S.-S. Yu, S.-W. Chu, C.-M. Wang, Y.-K. Chan, T.-C. Chang, Two improved k-means algorithms. Applied Soft Computing 68:747-755 (2018). doi:https://doi.org/10.1016/j.asoc.2017.08.032.

[29] D. Posada, T. R. Buckley, Model selection and model averaging in phylogenetics: advantages of Akaike information criterion and Bayesian approaches over likelihood ratio tests. Systematic biology 53 (5):793-808 (2004). doi:http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.576.3563&rep=rep1&type=pdf.

[30] D. Pelleg, A. W. Moore X-means: Extending k-means with efficient estimation of the

number of clusters. In, 2000. pp 727-734. doi:http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.19.3377&rep=rep1&type=pdf.

[31] P. Smyth Clustering Using Monte Carlo Cross-Validation. In: KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 1996. AAAI Press, pp 26-133. doi:https://www.aaai.org/Papers/KDD/1996/KDD96-021.pdf.

[32] K. B. a. M. Lichman (2013) UCI machine learning repository. doi:https://archive.ics.uci.edu/ml/datasets/Iris.

[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, Scikit-learn: Machine learning in Python. the Journal of machine Learning research 12:2825-2830 (2011). doi:https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf.