Lecture Node for K-means

Xuehai Liu
17341111

Abstract

Data clustering is a widely used model for understanding and learning. Clustering analysis clusters objects according to the similarity of measurable feature attributes of objects, which is widely used in many fields. K-means algorithm, which was first published in 1955, is one of the most popular simple clustering algorithms. K-means is still widely used after over 60 years, which shows that it is a difficult challenge to design a clustering algorithm with wide applicability. However, more and more methods are proposed to improve the performance of K-means, and it's still a hot topic now to study cluster algorithms and k-means. K-means algorithm is a kind of unsupervised learning clustering algorithm. Its purpose is to cluster the data according to the characteristics of the data itself without knowing the category and number of categories. In this paper, I will review the k-means algorithm, summarize the main algorithm and main challenges of K-means, and discusses other related improvements. Finally, an experiment on a small data set is conducted and the results are analyzed.

1. Introduction

1.1. Clustering

In pattern recognition, the task that we want to predict the behavior of unknown test data is called ”Learning”. Generally, there are two kinds of learning: (1) supervised learning (classification) and (2) unsupervised learning (clustering). Among them, supervised learning only involves labeled data, while unsupervised learning only involves unlabeled data.[5] Clustering is more difficult than classification. One of its mathematical definitions can be: given a certain representation of n objects, find K groups based on similarity measurement, so that the similarity in the same group is high while that in different groups is low.

However, the definition is somewhat vague. It has several key problems: 1. What is the definition of similarity? 2. What is the definition of cluster? How to determine how many clusters the data should be divided into? [3]In order to realize clustering algorithm by computer, these problems need to be defined strictly. These problems lead to the emergence of more and more improved clustering algorithms since the release of k-means.

1.2. Algorithms of Clustering

Clustering algorithms can be roughly divided into two types: hierarchical and partition based. Hierarchical clustering algorithm uses agglomerative pattern (starting from each data point as a cluster to fuse the most similar cluster) or split pattern (starting from dividing all data points as a whole cluster recursively into smaller clusters) to find clusters recursively. In contrast, the partition based approach does not use a hierarchical structure and will find a partition of the dataset. The input of hierarchical algorithm is n * n similarity matrix, and N is the number of clustering data. The partition based algorithm can accept n * D pattern matrix, where n represents samples with n d-dimensional features. The most popular partition based algorithm is k-means.

1.3. k-means

K-means clustering algorithm (K-means clustering algorithm) is an iterative clustering analysis algorithm. Its step is to divide the data into k groups, then randomly select k objects as the initial clustering centers, and then calculate the distance between each object and each seed cluster center, and assign each object to the nearest cluster center. Cluster centers and the objects assigned to them represent a cluster. Each time a sample is allocated, the cluster center of the cluster is recalculated according to the existing objects in the cluster. This process will be repeated until a termination condition is met. The termination condition can be that no (or the minimum number) objects are reassigned to different clusters, no (or the minimum number) cluster centers change again, and the sum of squares of errors is local minimum.

2. Application Scenarios

Clustering analysis is very common in any field involving multivariate data analysis. As one of the most widely used simple clustering algorithms, K-means algorithm has excellent performance in various fields.

2.1. business

Clustering analysis is used to find different customer groups and characterize different customer groups by purchasing patterns.

Cluster analysis is an effective tool for market segmentation. It can also be used to study consumer behavior, find new potential markets, select experimental markets, and as a pretreatment of multivariate analysis.

2.2. biology

Cluster analysis is used to classify plants and animals, and to classify genes, so as to obtain an understanding of the inherent structure of the population

2.3. Internet

Clustering analysis is used to classify documents on the Internet to repair information.

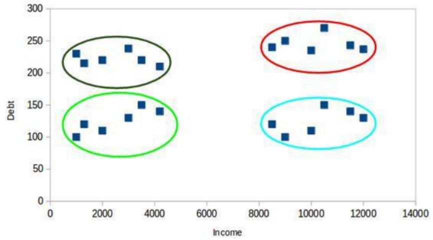


Figure 1. A tiny case of K-means used in bank service to divide customers into different groups by their income and debt

2.4. E-commerce

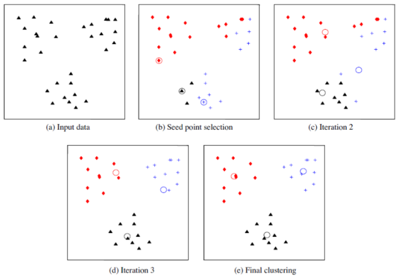
Clustering analysis is also a very important aspect in the data mining of website construction in e-commerce. By grouping and clustering customers with similar browsing behavior, and analyzing the common characteristics of customers, it can better help e-commerce users understand their customers and provide more appropriate services to customers.

3. Method and Mathematical derivation

3.1. K-means Algorithm

3.1.1 Problem Definition

Let x = xi, i = 1,.. n be the d-dimensional point set to be divided into K clusters C = Ck, k = 1..K . K-means algorithm finds one split that minimizes the square error between the empirical mean and the points in a cluster. Let k be the mean value of Ck, the square error between k and points in class C is defined as $J\left(c_k\right)=\sum_{x_i\in c_k}(x_i-u_k)^2$. The goal of k-means is to minimize the sum of square errors in all K clusters $J(C)=\sum_{k=1}^{K}\sum_{x\in c_k}(x_i-u_k)^2$. Minimizing this objective function is a NP hard problem (even if k = 2) . Therefore, K-means is a greedy algorithm, which can only converge to the local optimal solution, even though some studies have shown that K-means has a great probability of converging to the global optimal solution when the clusters are well separated. K-means starts from the initial K clusters and allocates patterns to the



(a) Input data (b) Seed point selection (c) Iteration 2

(d) Iteration 3 (e) Final clustering

Figure 2. An example of K-means algorithm. Figure (a) shows the two-dimensional input data with three clusters; figure (b) shows the three seed points used as the center of the cluster and the initial cluster allocation of the data points. Graphs (c) and (d) represent iterations of cluster labels and cluster centers. Graph (E) represents the final clustering result of K-means convergence

clusters in order to reduce the square error. Because when the number of clusters K increases, the square error always decreases (when k = n, J (c) = 0), only when the number of clusters is a fixed number can J (c) be minimized.

3.1.2 Algorithm Discription

k-means algorithm takes the distance between data as the standard of similarity measurement of data objects. Therefore, the calculation method of distance between data has a significant impact on the final clustering effect. The commonly used methods of calculating distance are: cosine distance, Euclidean distance, Manhattan distance, etc. The following is an example of Euclidean distance:

$$\operatorname{dist}\left(x_i, x_j\right)=\sqrt{\sum_{d=1}^D\left(x_{i,d}-x_{j,d}\right)^2}$$

Through the formula, the distance between each pair of data objects can be calculated, and the number of classes K can be clustered according to the distance. There are many ways to select the centroid of data in each category, such as:

- The mean value of all the data;
- Randomly take k data as centroid;
- Select the farthest k points as the centroid.

All of the above methods need to iterate the initial centroid. When the centroid changes slowly, it can be considered as convergence, and this point is the final centroid.

$$\text{Center}\ _k=\frac{1}{\left|C_k\right|} \sum_{x_i \in C_k} x_i$$

Here, C_k Represents the Kth class and $|C_k|$represents the number of data objects in the K class.

3.1.3 End Situation

Usually, there are three situations that the algorithm will stop.

- The centroid of the newly formed cluster will not change
- Data points remain in the same cluster
- Maximum number of iterations is reached

If the centroid of the new cluster does not change, we can stop the algorithm. Even after many iterations, all clusters still have the same centroid. We can say that the algorithm does not learn any new patterns, and it is a sign to stop training.

Another obvious sign is that after many iterations of training, if the data points are still in the same cluster, we should stop the training process.

Finally, if the maximum number of iterations is reached, we can stop training. Suppose we set the number of iterations to 100. The process repeats 100 iterations before stopping.

3.1.4 Algorithm Steps

Above all,the steps of K-means algorithm can be discribed as follow:

1. Select the number of clusters K (k-means algorithm only needs to set the maximum K value when passing super parameters)

2. Randomly generate K clusters, and then determine the cluster centers, or directly generate K centers.

3. Determine the cluster center of each point.

4. Calculate the new clustering center.

5. Repeat the above steps until the convergence requirements are met.

3.2. K-means Parameters

K-means algorithm requires users to determine three parameters: the number of classes K, class initialization, distance scale.

3.2.1 K

Among them, the most important parameter and the easiest one to affect the performance is the parameter k. Automatic determination of K is one of the most difficult problems in clustering. Most of the existing methods to determine the number of classes K is to transform it into a model selection problem. By making the clustering algorithm run under different K values, and then select the best K according to the defined scoring criteria.

Unlike the classification and regression problems of supervised learning, unsupervised clustering has no sample output, so there is no direct clustering evaluation method. Here, I will introduce two methods to determine the value of K.[2]

(1) Inertia determination method, inertia is only the sum of the square distance between the sample and its nearest cluster center. Based on Euclidean distance, the problem that k-means algorithm needs to optimize is to minimize the within cluster sum of squared errors (SSE), also known as cluster inertia. As the number of categories increases, the value of SE will become smaller and smaller. However, it doesn't mean that the larger the number of categories is, the performance of algorithm is the better. When selecting, we always need to select "the K value at inflection point".

$$S_E=\sum_{i=1}^{r}\sum_{j=1}^{n_i}\left(X_{ij}-\bar{X}_i\right)^2$$

(2)Another method is to evaluate the clustering effect from the density within the cluster and the dispersion between clusters. The common methods are the contour coefficient silhoutte coefficient and calinski harabasz index. Among them, the calculation of calinski harabasz index is simple and direct. The larger the calinski_harabasz_score is, the better the clustering effect is. The mathematical formula of calculating calinski_harabasz score is as follows:

$$s(k)=\frac{\operatorname{tr}(B_k)}{\operatorname{tr}(W_k)}\frac{m-k}{k-1}$$

Here, m is the number of training samples and K is the number of categories. B_k is the covariance matrix between categories and wk is the covariance matrix of data within categories. TR is the trace of the matrix. In other words, the smaller the covariance of the data within the category ,the better the performance is, and the greater the covariance between the categories, the better performance is. In scikit_learn, the corresponding method of calinski harabasz index is metrics.calinski_ harabsaz_ score.

3.2.2 Class Initialization

Secondly, due to the greedy strategy of K-means, it can only converge to the local optimal solution, which leads to different clustering results with different initialization. To find the global optimal solution, a common method is to fix K, use different class initialization to run k-means algorithm, and then find the partition with the least square error. However, some useful methods including k-means++ are doing better now to help initialize the centroids, and therefore contribute to the performance of k-means algorithm. We will discuss about this later.

3.2.3 Distance Scale

Finally, the k-means algorithm takes the distance between data as the standard of similarity measurement of data objects, so the selection of the scale to calculate the distance between data has a significant impact on the final clustering effect. However, simple K-means algorithm often uses simple Euclidean distance, which limits its practicability in complex space. Later, we will discuss the kernel method to help solve this problem.

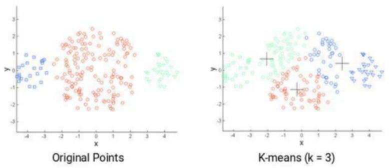


Figure 3. When k-means deal with spherical clusters with not uniform distribution of data objects, the performance would be impacted.

3.3. Difficulties for K-means

K-means has mainly four shortcomings.

- K-means clustering algorithm requires users to specify the number of clusters K value in advance. In many cases, when clustering data sets, users don't know how many classes the data set should be divided into, so it is difficult to estimate the K value

- It is sensitive to the initial cluster center. Choosing different cluster centers will produce different clustering results and different accuracy. The random selection of initial cluster centers will lead to the instability of the algorithm and may fall into the local optimal situation

- It is sensitive to noise and outlier data. K-means algorithm takes the centroid of cluster as the cluster center and adds it to the next round of calculation. Therefore, a small amount of such data can have a great impact on the average value, leading to the instability and even error of the results

- Because k-means algorithm mainly uses Euclidean distance function to measure the similarity between data objects, and uses the sum of squares of errors as the criterion function, it can only find spherical clusters with uniform distribution of data objects. When the size and density of clusters are not uniform, the result would always be impacted, as Figure 3 shows.

4. Related Works and Improvements

Since K-means method is still limited, in recent years, several methods were proposed to improve k-means.Now I will choose 3 representative algorithm to analyze.Here, we can see that these methods are actually optimizing the three parameters of K-means I mentioned before.

(1) K-means and K-means++: the original k-means algorithm randomly selects k points in the data set as cluster centers, and K-means++ selects K cluster centers according to the following idea: assuming n initial clustering centers have been selected (0¡n¡K), then when selecting the N + 1 cluster center, the farther away from the current n cluster centers, the more likely it is to be selected as the N + 1 cluster center. The first cluster center (n = 1) is also selected by random method. It can be said that this is also in line with our intuition: cluster centers are, of course, as far away from each other as possible. This improvement is intuitive and simple, but it is very effective.

(2) K-means and ISODATA: the full name of ISODATA is iterative self-organizing data analysis method. In K-means, the value of K needs to be determined artificially in advance and cannot be changed in the whole process of the algorithm. However, when it comes to high-dimensional and massive data sets, it is often difficult to accurately estimate the size of K. The idea of ISODATA is also very intuitive: when the number of samples belonging to a certain category is too small, the class is removed; when the number of samples belonging to a certain category is too large and the degree of dispersion is large, the category is divided into two subcategories.

(3) K-means and kernel k-means: the traditional K-means uses Euclidean distance to measure the similarity between samples. Obviously, not all data sets are suitable for this measurement method. According to the idea of kernel function in support

vector machine, mapping all samples to another feature space and clustering, it is possible to improve the clustering effect. This article does not introduce kernel k-means in detail.

4.1. K-means++

K-means++ selects the initial cluster centroid for k-means clustering. [4] The original k-means algorithm initially randomly selects k points in the data set as the clustering centers, while K-means++ selects K clustering centers according to the following idea: assuming that n initial clustering centers have been selected ($0 < n < k$), then when selecting the N + 1 cluster centers, the farther away from the current n cluster centers, the more likely the points will be selected as the N + 1 cluster centers. The first cluster center (n = 1) is also selected by random method. It can be said that this is also consistent with our intuition: the center of clustering is, of course, the farther away from each other, the better. This improvement is intuitive and simple, but it is very effective.

4.1.1 Algorithm Steps

- 1.Select the first cluster randomly and evenly from the data points we want to cluster. This is similar to what we did in K-means
- 2.But instead of randomly selecting all centroids, we choose one here
- 3.Next, we calculate the distance (d (x)) between each data point and the centroid of the selected cluster
- 4.Then, a new cluster centroid is selected from the data points, and the point corresponding to the square of the distance from the largest point to its nearest centroid is the new cluster centroid.

Then, repeat steps 2 and 3 until K clusters are selected.

4.2. ISODATA algorithm

As mentioned earlier, the number of clustering centers K for k-means and K-means++ is fixed. The ISODATA algorithm can adjust the number of cluster centers K according to the actual situation of each category. (1) split operation, corresponding to increase the number of cluster centers; (2) merge operation, corresponding to reduce the number of cluster centers

4.2.1 Algorithm Steps

step1: randomly select k0 samples from the data set as the initial clustering center,$C = \{c_1, c_2, \dots, c_{K_0}\}$

step2: for each sample X in the dataset, calculate the distance from it to K0 cluster centers and divide it into classes corresponding to the cluster centers with the smallest distance;

step3: judge whether the number of elements in each class is less than Nmin. If it is less than Nmin, it is necessary to discard the class so that k = k-1, and the samples in this class are redistributed to the class with the smallest distance among the remaining classes; ·

step4: for each class Ci, its cluster center $c_i = \frac{1}{|c_i|}\sum_{x \in c_i}$ (i.e., the centroid of all samples belonging to the class) needs to be recalculated;

step5: if the current $K \leq \frac{K_0}{2}$, it indicates that the current number of categories is too small, and go to the splitting operation;

step6: if the current $K \geq 2K_0$, Indicates that the current number of categories is too many, so go to the merge operation;

step7: if the maximum number of iterations is reached, terminate, otherwise, go back to step 2 to continue;

4.3. Kernel K-means

[1] When the clusters of data points in the data set have great differences in shape and density, the standard k-means algorithm can not play its role.

The kernel method is to transform the data set into a data style that can be received by standard k-means algorithm through a mapping, and then it is processed by clustering algorithm. This is the kernel k-means algorithm.

Main idea: through a nonlinear mapping, the data points in the input space are mapped to a high-dimensional feature space, and the appropriate kernel function is selected to replace the inner product of the nonlinear mapping, and the clustering analysis is carried out in the feature space. This method of mapping data to high-dimensional space can highlight the feature differences between sample categories, making samples linearly separable (or approximately linearly separable) in kernel space.

4.3.1 Algorithm Steps

Input: all data point a, cluster number K
Output: K clustering centers
1: The input data is mapped to high dimensional space by kernel function to get matrix B
2: Standard K-means clustering is used for B

4.4. Conclusion for related works

Now, we can find out how the above related works are doing with the k-mean's parameters: K, class initialization and distance scale.The mean idea of K-means++ is : the farther away from the current N cluster centers, the higher probability will be selected as the N + 1 cluster center. It is a simple and direct optimization on initializing the centroids and have a larger chance to find out the global optimal solution.

Next, the ISODATA algorithm can dynamically adjust the number of cluster centers K according to the actual situation of the samples in each class. If a certain class has a large degree of sample dispersion (measured by variance) and the number of samples is large, split it; if two categories are close (measured by the distance of cluster centers), they are merged.

Finally, Kernel K-means algorithm uses the idea similar to SVM kernel function, the original simple distance scale is transformed into high-dimensional complex representation by kernel function, so as to obtain more effective distance measurement.

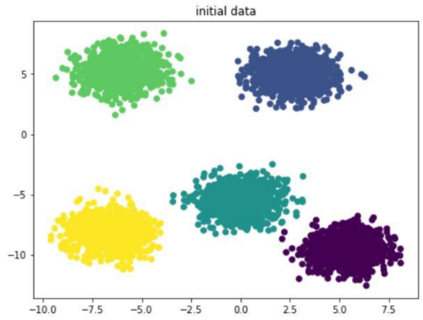


Figure 4. Experiment Data

5. Experiment

5.1. Experiment Discription

I conduct an experiment with k-means on dividing randomly initialized 2-dimension points into several groups(K is unknown), as Figure 4 shows. Here, k-means method is initialized by k-means++method. The value range of n_clusters is 2 - 9, and for each n_clusters, SSE and calinski in clustrs_harabaz score is recorded.By comparing the SSE and calinski in clustrs_harabaz score, I will figure out the best K value to cluster the data.

The codes are written in kmeans.py.

5.2. Result Analysis

Here, I test several possible K to find the best split for the above data.

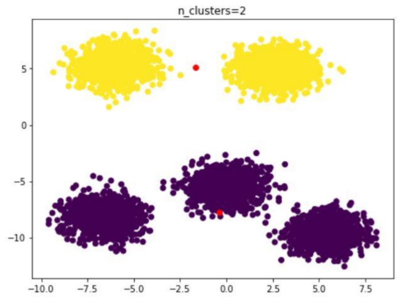


Figure 5. K = 2

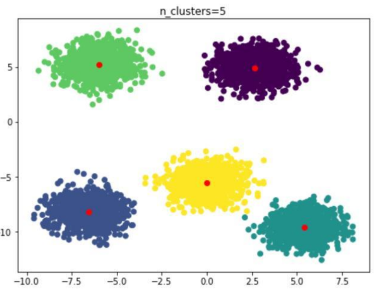


Figure 6. K = 5

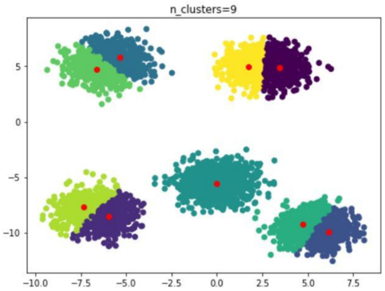


Figure 7. K = 9

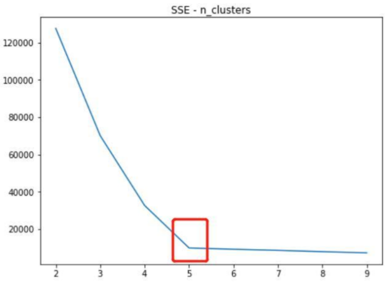


Figure 8. SSE

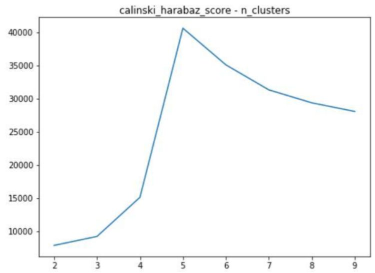


Figure 9. metrics.calinski_harabsaz_score

According to the definition of SSE, SSE changes with n_clusters. With increase of n_clusters, SSE will be smaller. Generally, the value of K will be taken at the inflection point of SSE. At the same time, the value of calinski_harabaz_score should be referenced. In this case, it can be seen that there is a big turning point for SSE when n_clusters = 5. The maximum value of calinski_harabaz_score is obtained also when n_clusters = 5.Therefore, we can draw a conclusion that the k-means algorithm correctly divide the data into 5 parts and get a excellent performance on the experiment.

6. Conclusion

In this paper, I have reviewed k-means clustering method, summarized the main algorithm and main challenges of K-means, and discusses other related improvements.We can see that three parameters: number of classes K, class initialization, and distance scale are playing an important role on the performance of K-means. Therefore, different methods including K-means++ are proposed to optimize the origin K-means algorithm and this topic is still hot discussed nowadays. Finally, through the experiment on a small data set, we can see that using SEE and calinski_harabsaz_score can help us determine the best K for the algorithm and therefore improve the performance of K-means to do clustering jobs.

References

[1] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized

cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556, 2004.

[2] Greg Hamerly and Charles Elkan. Learning the k in k-means. In *Advances in neural information processing systems*, pages 281–288, 2004.

[3] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[4] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003.

[5] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.