A fast k-means algorithm based on multi-granularity

Qing Wen, Junkuan Wang, Yabin Shao, Zizhong Chen

Qing Wen, Junkuan Wang, Yabin Shao, Zizhong Chen, "A fast k-means algorithm based on multi-granularity," Proc. SPIE 12174, International Conference on Internet of Things and Machine Learning (IoTML 2021), 1217410 (22 April 2022); doi: 10.1117/12.2628453

SPIE.

Event: International Conference on Internet of Things and Machine Learning (IoTML 2021), 2021, Shanghai, China

A fast k-means algorithm based on multi-granularity

Qing Wen[1], JunkuanWang[2*], Yabin Shao[3], and Zizhong Chen[4]
[1]Chongqing Key Laboratory of Computational Intelligence, College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China
[2] Chongqing Key Laboratory of Computational Intelligence, College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China
[3] Faculty of Science, Chongqing University of Posts and Telecommunications, Chongqing, China
[4]Department of Computer Science and Engineering, University of California, Riverside, California, USA
*Corresponding author: S190231167@stu.cqupt.edu.cn

ABSTRACT

The k-means algorithm has been widely used since it was proposed, but the standard k-means algorithm does not perform well in terms of efficiency when dealing with large-scale data. To solve this problem, in this paper, we propose a fast k-means algorithm based on multiple granularities. First, from the coarse-grained perspective, we use the clustering distribution information to narrow the search range of sample points, which makes the proposed algorithm very advantageous on large k. Second, from the fine-grained perspective, we use the rules of upper and lower bounds to reduce the number of sample points involved in the distance calculation, thus reducing many unnecessary distance calculations. Finally, we evaluate the proposed k-means algorithm on several real-world datasets, and the experimental results show that the proposed algorithm converges hundreds of times faster than standard k-means on average with the accuracy loss controlled at about three percent, and the speedup of the algorithm is more obvious when the dataset size is larger and the dimensionality of the dataset is higher.

Keywords: k-means, multiple granularities, neighbor cluster, upper bound, lower bound

1. INTRODUCTION

k-means algorithm was proposed by James MacQueen in 1967, which advocates clustering based on the distance between sample points[1]. Nowadays, the k-means algorithm has become one of the top ten classical machine learning algorithms and is used in a wide range of fields[2][3]. The Lloyd algorithm is known as the standard k-means algorithm[4], its core idea is to divide a given sample set into k clusters according to the distance between the sample points, and make the points in the same cluster as close as possible, the distance between the clusters as large as possible， which can be described by the following data expressions. Assuming that there are n sample points and k clusters represented as $(x_1, x_2, \ldots, x_n)$ and $(C_1, C_2, \ldots, C_k)$ respectively, our goal is to minimize the square error E, that is,

$$E = \min \left(\sum_{i=1}^{k} \sum_{x_i \in C_i} \left(\left||x_i - c_i|\right|^2\right)\right) \quad (1)$$

The k-means algorithm is a distance-based calculation. It is performed in two phases: the assignment phase, in which each point is assigned to the nearest cluster, and the update phase, in which the centroids of each cluster are recalculated after the assignment phase. These two phases are repeated until the algorithm converges. In the standard k-means Lloyd algorithm, the initial centroids are chosen randomly, which may lead to different number of iterations and clustering results will vary from one execution to another. Lloyd has been shown to be an NP-hard problem. It has a time complexity of O(nkt), where n is the data size, k is the number of clusters, and t is the total number of iterations. In very in very large scale clustering, k is a value of hundreds or larger. Lloyd's k-means can be inefficient. Therefore, many speedup algorithms have been proposed for Lloyd.

2. RELATED WORK

Since the standard k-means algorithm needs to calculate the distance from all sample points to the center of each cluster, the standard k-means algorithm does not perform well on big data. Therefore, it is a very important research field to improve the efficiency of k-means algorithm, the most commonly used method is to reduce the calculation of the distance

International Conference on Internet of Things and Machine Learning (IoTML 2021),
edited by Pushpendu Kar, Steven Guan, Proc. of SPIE Vol. 12174, 1217410
 · 0277-786X · doi: 10.1117/12.2628453

between the point and the cluster center. Using this method to improve the efficiency of k-means will lead to two results, one is to accelerate the approximate k-means [6-9], which is to lose accuracy to obtain higher efficiency. For example, Fahim AM et al. proposed a criterion that as long as the point x is reached in the current iteration The distance from the center is less than the distance from x to the center in the previous iteration, so the calculation of distance from x to the rest of the centers can be avoided [7]. Perez J et al. proposed a heuristic method, that is, if the distance from point x to the center is less than the sum of the two minimum centroid offsets, then the calculation of distance from x to the rest of the centers can be avoided[8], Daowan Pen et al. proposed to narrow the search range of each sample point through the neighbor information, thereby reducing unnecessary distance calculations[10]; the other result is to accelerate the exact k-means [10-14], Elkan proposed to use triangle inequality to achieve this goal in 2003, and set upper and lower bounds for each point x. As long as the upper bound is smaller than the lower bound or the upper bound is smaller than the minimum distance between the centroids, there is no need to calculate the distance between x and the remaining centroids[11]. The Elkan considerably decreased the times of distance calculation and improve the efficiency. However, too many lower bounds leads to a large time complexity and high space cost. As a most classical exact k-means, Harmerly improved the elkan algorithm by decreasing the number of lower bounds to one, resulting in a considerable improvement in efficiency[12]. Lots of excellent algorithms are developed based on the harmerly algorithm. To further improve its efficiency when the center offset is large, the Annulus algorithm[13] and Exponion algorithm[14] are proposed by drawing out possible areas for each point participating in the distance calculation. Yinyang k-means [15] compared the characteristics of elkan and Hamerly algorithms, and proposed three filters: global filtering, group filtering and local filtering, and combined the three filters to overcome the need to set k lower bounds [11] and the defect of being sensitive to the large offset of the centroid[12]. Ball k-means treats all clusters as a sphere, and the search range of samples in all clusters is based on the relationship between the distance and the radius of the center of the sphere[5][16].

In this paper, we are committed to proposing a more efficient k-means algorithm based on multi-granularity ideas. First of all, from a coarse-grained point of view, we narrow the searching range of points by find neighbor clusters; Secondly, we reduce points that need to participate in distance calculation by setting lower bound and upper bound for each points.

3. NEW ALGORITHM

we propose an efficient k-means algorithm based on the idea of multi-granularity. This section will describe the acceleration standards of the algorithm from the two perspectives of coarse-grained and fine-grained. In order to distinguish between coarse-grained and fine-grained acceleration criteria, this article regards each cluster as a coarse-grained and each sample point as a fine-grained.

3.1 Coarse-grained acceleration criteria

In order to improve the efficiency of the k-means algorithm from a coarse-grained perspective, the distribution relationships between clusters can be used to narrow the search range of sample points. Our algorithm borrows the concept of neighbor clusters [9], which are clusters that have exchanged points with the current cluster during the previous iteration, as defined in Definition 1.

Definition 1. Given a cluster C, the neighbor clusters of C in the t-th iteration can be defined as the set of clusters that exchange points with C in the (t-1)-th iteration, that is,

$$NC_c{}^t = \left\{C_i | \exists x \in C^{t-1} \& x \in C_i{}^t\right\} \cup \{C_j | \exists x \in C_j{}^{t-1} \& x \in C^t\} \quad (2)$$

Where $NC_c{}^t$ represents the set of neighbor clusters of cluster C during the t-th iteration.

For a cluster C, if there is a point x, it belongs to cluster C in the (t-1)-th iteration, and belongs to C_i in the t-th iteration (or the sample point x belongs to the cluster C_i during the (t-1)-th iteration, Belongs to C in the t-th iteration), then C_i is the neighbor cluster of C.

Based on the definition of neighbor clusters, this algorithm further proposes that in the t-th iteration, the points in cluster C can only be allocated to its neighbor clusters. That is, the calculation of the distance between the points in the cluster C and the non-neighbor clusters of the cluster C can be directly avoided, which greatly narrows the search range of points in cluster C.

3.2 Fine-grained acceleration criteria

The new algorithm maintains an upper bound and a lower bound for each sample point for further acceleration. And initializes the upper bound of the sample point as the distance from the sample point to the nearest cluster center, and the lower bound as the distance from the sample point to the second nearest cluster center. The definition and the update method of the upper and lower bounds of each sample point will be given in Definition 2 and Theorem 1.

Definition 2: Given a cluster C with c as its center, in the t-th iteration process, $\forall x \in C$, the upper bound of x is always greater than or equal to the distance from x to c, and the lower bound of x is always less than or equal to the distance from x to all other clusters, that is,

$$u(x)^t \geq \left||x - c^t|\right| \quad (3)$$

$$l(x)^t \leq min||x - {c_i}^t|| \quad (4)$$

Where $u(x)$ and $l(x)$ represents the upper bound and lower bound of x respectively, and c_i represents the center of the second closest cluster to x.

According to the definition of the upper bound and lower bound, it can be concluded that as long as the lower bound of a sample point is greater than the upper bound, then the closest cluster to the sample point is the cluster where the sample point is currently located, that is, the sample point can avoid participating in distance calculation in the current iteration.

In order to ensure that Definition 2 is always valid, Theorem 1 can be used to update the upper bound and lower bound of each sample point.

Theorem 1: Given a cluster C with c as its center, ∀x∈C, the upper bound of the sample point x in the t-th iteration can be expressed as the upper bound in the previous iteration plus the offset of the center c, and the lower bound can be expressed as the lower bound of the sample point x in the previous iteration minus the maximum cluster center offset, that is,

$$u(x)^t = u(x)^{t-1} + p(c) \quad (5)$$

$$l(x)^t = l(x)^{t-1} - \max\left(p\left(c_j\right)\right) \quad (6)$$

Where $p(c)$ and $\max\left(p\left(c_j\right)\right)$ respectively represent the center offset of cluster C and the maximum center offset of all clusters.

Proof: According to definition 1, for a sample point x in a given cluster c, in the t-1 iteration, we have $u(x)^{t-1} \geq ||x - c^{t-1}||$,

Therefore in the t-th iteration, it can be concluded, $\left||x - c^t|\right| \leq \left||x - c^{t-1}|\right| + p(c) \leq u(x)^{t-1} + p(c)$;

In the same way, we have $l(x)^{t-1} \leq min\left|\left|x - {c_j}^{t-1}\right|\right|$,

Therefore in the t-th iteration, it can be concluded, $min\left|\left|x - {c_j}^t\right|\right| \leq min\left|\left|x - {c_j}^{t-1}\right|\right| - p\left(c_j\right) \leq l(x)^{t-1} - p\left(c_j\right) \leq l(x)^{t-1} - \max\left(p\left(c_j\right)\right)$.

In summary, the upper and lower bounds of the sample points updated by Theorem 1 can always satisfy Definition 2.

3.3 Overall description of new algorithm

The flow of the algorithm is shown in Figure 1, which can be divided into the following main steps.

Step 1: use standard k-means to cluster twice, and initialize the upper bound of the sample point as the distance from the sample point to the nearest cluster center, the lower bound as the distance from the sample point to the second nearest cluster center ;

Step 2: calculate the offsets of all cluster centers, and update the upper and lower bounds of sample points through theorem 1.

Step 3: calculate the distance between the points in the cluster that do not meet the upper bound less than the lower bound, assign the sample point to the nearest cluster, and change the upper bound of the sample point into the distance from the

cluster center closest to the sample point, and the lower bound into the distance from the cluster center second closest to the sample point, and find out the neighbor clusters of the current cluster according to definition 1.

Step 3: Update all cluster cnters.

Step 5: judge whether the new cluster center is completely equal to the old cluster center. If it is completely equal, output the clustering result, otherwise jump to step 2.

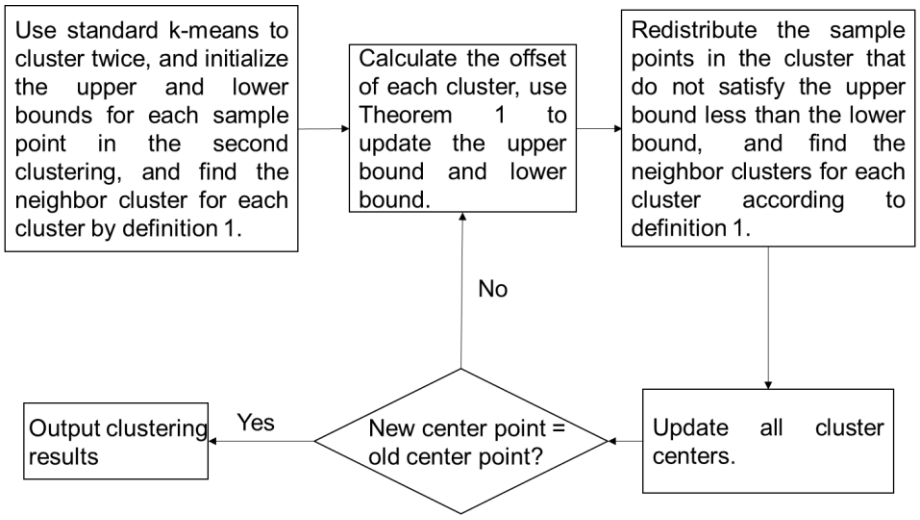

Figure 1. The flow of new algorithm.

4. EXPERIMENT

To verify the efficiency of the algorithm, this paper uses multiple k-means algorithms for comparison, including two exact k-means algorithms (Lloyd and ball k-means, respectively), and an approximate k-means algorithms which is proposed by daowan Pen et al. in 2021 (hereafter app_k-means). In this paper, these four algorithms are applied to different k values on different datasets and the efficiency of the algorithms is compared. A brief description of these datasets is given in Table 1.

Table 1. Dataset information.

Dataset	Number of points	dimension
Fourclass	862	3
ijcnn	141690	22
Birch	100000	2
isolet	6598	167

The performance of each algorithm on each dataset was evaluated and the running time of each algorithm was calculated for each value of k until the algorithm converged. The specific results are shown in Table 2, and it can be seen that the proposed method performs the best on all datasets, and our proposed algorithm is hundreds or even thousands of times faster than the standard k-means algorithm on most datasets, and also twice as fast as another approximate k-means algorithm on average. According to the experimental results, it can be seen that the proposed algorithm accelerates more significantly on the dataset ijcnn than on the fourclass dataset, and the algorithm accelerates thousands of times on all k values of high-dimensional data isolet, indicating that the algorithm can solve the problem of inefficiency of the standard k-mean algorithm on large-scale datasets and high-dimensional data, and this acceleration effect is stronger at larger k values the larger the value of k is, the more obvious this speedup effect is. Table 3 compares the sum of squared errors (SSE) of the proposed algorithm and app k-means. Because the SSE after convergence of Ball k-means is the same as the clustering result of Lloyd's k-means algorithm, the algorithm is not shown in Table 3. Combining Tables 2 and 3 shows

that the average speedup of the proposed algorithm is several hundred or even several thousand times higher, while the average growth rate of SSE is about 3% on all data sets.

Table 2. Running time.

Dataset	k	Lloyd	ballk-means	app_k-means	ours
fourclass	k=50	1.55	0.27(6)	0.08(19)	**0.04(39)**
	k=100	1.53	0.15(10)	0.03(51)	**0.03(51)**
	k=300	2.25	0.09(25)	0.02(113)	**0.01(225)**
ijcnn	k=50	9978.35	320.38(31)	57.96(172)	**49.19(203)**
	k=100	53593.72	547.62(98)	79.66(673)	**43.48(1233)**
	k=300	40864.87	737.74(55)	74.15(551)	**41.12(994)**
birch	k=50	554.98	66.67(8)	35.62(16)	**12.76(43)**
	k=100	1418.16	95.01(15)	49.68(29)	**13.29(107)**
	k=300	6564.42	173.98(38)	38.7(170)	**15.47(424)**
isolet	k=50	1906.68	30.39(63)	1.86(1025)	**1.58(1207)**
	k=100	6338.7	85.76(74)	2.37(2675)	**1.3(4876)**
	K=300	43420.1	719.83(60)	25.84(1680)	**10.01(4235)**

Table 3. Distortion (SSE) compared with Lloyd's algorithm.

Dataset	algorithm	K=50	K=100	K=300	avg
fourclass	app k-means	3.04%	1.47%	1.24%	1.91%
	ours	4.71%	2.50%	1.45%	2.88%
ijcnn	app k-means	0.18%	0.42%	1.05%	0.55%
	ours	1.18%	1.92%	3.03%	2.04%
birch	app k-means	1.23%	0.49%	1.73%	3.45%
	ours	2.10%	2.36%	3.93%	2.79%
isolet	app k-means	0.22%	0.31%	0.27%	0.26%
	ours	0.83%	0.91%	0.53%	2.27%

5. CONCLUSION

This paper proposes a fast approximation k-means algorithm based on multiple granularities. There are two reasons that lead our proposed algorithm to be so efficient, one is that the algorithm uses the cluster distribution information to narrow down the search, and the other is that the algorithm reduce the sample points that need to be involved in the search by using upper bound and lower bound. So our algorithm provides an efficient solution for clustering large scale and high-dimensional data.

ACKNOELWGMENT

This work was supported in part by National Key Research and Development Program of China (2019QY(Y)0301, the N ational Natural Science Foundation of China under Grant Nos. 62176033 and 61936001, and the Natural Science Founda tion of Chongqing No. cstc2019jcyj-cxttX0002.

REFERENCES

[1] Macqueen, J. . (1965). Some Methods for Classification and Analysis of MultiVariate Observations. In: Proc of Berkeley Symposium on Mathematical Statistics and Probability. California. pp. 281–297

[2] Xia, S. , Zhang, Z. , Li, W. , Wang, G. , Giem, E. , & Chen, Z. . (2020). Gbnrs: a novel rough set algorithm for fast adaptive attribute reduction in classification. IEEE Transactions on Knowledge and Data Engineering, 99: 1-1.

[3] Prez-Ortega, J. , Almanza-Ortega, A. , VegaVillalobos, A. , Pazos-Rangel, R. and Martnez-Rebollar, A. . (2019). The K-Means Algorithm Evolution.

[4] Lloyd, S. P. . (1982). Least squares quantization in pcm. IEEE Trans, 28(2): 129-137.

[5] Xia, S. , Liu, Y. , Xin, D. , Wang, G. , & Luo, Y. . Granular ball computing classifiers for efficient, scalable and robust learning. Information Sciences, 483:136–152 (2019).

[6] aul S. Bradley and Usama M. Fayyad. Refining Initial Points for K-Means Clustering. In: Jude W. Shavlik. (Eds.), Proceedings of the Fifteenth International Conference on Machine Learning. Morgan Kaufmann, Madison, Wisconsin. pp. 91-97 (1998).

[7] Fahim, A. M. , Salem, A. M. , Torkey, F. A. , and Ramadan, M. A. . An efficient enhanced k-means clustering algorithm. Journal of Zhejiang University. Science, 7(10), pp.1626-1633 (2006).

[8] Joaquín Pérez, Pires, C. E. , Balby, L. , Mexicano, A. , and Hidalgo, M. N. . Early classification: a new heuristic to improve the classification step of k-means. journal of information and data management, 4(2):94–94 (2013).

[9] Khandelwal, S., Awekar, A. Faster K-Means Cluster Estimation. In: Jose J. et al. (eds) Advances in Information Retrieval. Springer International Publishing, Cham. pp. 520—526 (2017).

[10] Peng, D. , Chen, Z. , Feng, F. , Xia, S. , and Wen, Q. . Fast k-means Clustering Based on the Neighbor Information. In: 2021 International Symposium on Electrical, Electronics and Information Engineering. Association for Computing Machinery, New York. pp: 551–555 (2021).

[11] Elkan, C. . Using the triangle inequality to accelerate k-means. Proceedings of the Twentieth International Conference on International Conference on Machine Learning. Washington. pp: 147–153 (2003).

[12] Hamerly, G. . Making k-means Even Faster. In: Proceedings of the 2010 SIAMInternational Conference on Data Mining. Ohio. pp:130-140 (2010).

[13] Hamerly, G. and Jonathan, D. Accelerating lloyds algorithm for k-means clustering. In: Celebi, M.(eds), Partitional clustering algorithms. Springer International Publishing, Cham. pp: 41–78 (2015).

[14] Newling, J. , et al. Fast k-means with accurate bounds. In: International Conference on Machine Learning. New York. pp: 936–944 (2016).

[15] Ding, Y. , Zhao, Y. , Shen, X. , Musuvathi, M. , and Mytkowicz, T. . Yinyang K-means: a drop-in replacement of the classic K-means with consistent speedup. In: International Conference on International Conference on Machine Learning. Lille, France. pp:579–587 (2015).

[16] Xia, S. , Peng, D. , Meng, D. , Zhang, C. , and Chen, Z. . A fast adaptive k-means with no bounds. IEEE Transactions on Pattern Analysis and Machine Intelligence, 99: 1-1 (2020).

[17] Xia S, Xiong Z, Luo Y, et al. Location difference of multiple distances based k-nearest neighbors algorithm[J]. Knowledge-Based Systems, 90: 99-110 (2015).