Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 3159-0040
Vol. 2 Issue 5, May - 2015

BS-means: Automatic Selection of k for k-means

Ahad Azarian
Lecturer in Taha Institute of Higher Education
Tehran, Iran
Ahad.azarian@gmail.com

Saeed Shahrivari
Freelancer
Tehran, Iran
saeed.shahrivari@gmail.com

***Abstract*—k-means is one of the most well-known and effective algorithms for data clustering. However, it has some shortcomings, and a challenging problem is determining the right number of clusters. Determining the number of clusters is very complex and is usually done by an expert. We have presented an algorithm called *BS-means* which can automatically determine the proper number of clusters of a dataset. Our approach is based on binary search, and can find the proper number of clusters in $O(n.\sqrt{n}.\log n)$ considering *n* as the number of items in the dataset. We have performed several experiments on various datasets and the results show that our algorithm is more accurate than competing solutions like X-means.**

Keywords—k-means++, X-means, automatic clustering

I. INTRODUCTION

One of the most popular techniques for data analysis is data clustering, also known as unsupervised learning. According to Jain's definition, "*The goal of data clustering, also known as cluster analysis, is to discover the natural grouping(s) of a set of patterns, points, or objects.*"[1]. Informally, we can define clustering as grouping of unlabeled objects into a set of groups, given a similarity metric. Data clustering has many applications. In Computer Vision, Image Segmentation can be defined as a clustering problem [2]. In Information Retrieval, document clustering is a very important method that can provide hierarchical retrieval and improvements in flat retrieval performance [3]. In Bioinformatics, clustering is used for improving multiple sequence alignment [4]. Many other applications also exist in other fields like: Online Shopping, Medicine, Online Social Networks, Recommender Systems, and etc.

A valid clustering should have two characteristics: i) *Cohesion*: the objects in one cluster should be as similar to each other as possible, and ii) *Separation*: clusters should be well separated i.e. the distance among the cluster centers must be large enough. Many different approaches have been proposed for clustering problem, such as Multi Variant Analysis, Graph Theory, Expectation Maximization, and Evolutionary Computing. Amongst available clustering algorithms, maybe the most well-known one is the k-means algorithm. Although different clustering algorithms have shown good performance while applied to specific problems, but k-means has proven to be efficient and fast if applied to various domains [1].

Despite of simplicity and effectiveness of k-means, it has several disadvantages, too. The quality of clustering highly depends on the initial seeds. Choosing bad seeds can result very bad clusters. Another challenge of the k-means algorithm is the requirement of giving the number of clusters as an input parameter. However, determining the correct number of clusters is very complex and usually needs an expert. Some mechanisms have been proposed for automatic selection of number of clusters, like the X-means algorithm. However, they are not accurate enough and usually result inaccurate cluster numbers.

In this paper, we proposed an algorithm for automatic selection of number of clusters without user interference. Our algorithm is called *BS-means*. BS-means uses a binary search based method combined with an elbow diagram to guess the number of clusters from the original dataset. We tested BS-means on several datasets and the experimental results show that the accuracy of BS-means is much better than competing solutions like X-means.

II. RELATED WORK

Data clustering is a very old problem and many solutions have been proposed since its inception. For a comprehensive survey on available methods, the interested reader can refer to [5]–[7][1]. Since our focus is on the k-means algorithm here we just concentrate on it. k-means was first described by Macqueen in 1967 [8]. Due to its simplicity and speed it was highly used in many problems. Some extensions and modifications have been proposed to overcome the problems of standard k-means.

X-means is a variant of k-means that uses the Bayesian Information Criterion (BIC) and the Akaike Information Criterion (AIC) measures for finding the optimal number of clusters [9]. Several initialization techniques have been proposed for finding proper initial seeds. Most of available methods try to make an approximation of the optimal solution [10]. One of the mostly cited works is *k-means++* that is Θ(*log n*)-competitive with the optimal solution [10]. Bahmani et al have also presented a parallel and scalable version of k-means++ called k-means|| [11]. Hadian and Shahrivari have also proposed a parallel k-means variant that is optimized to handle very large disk resilient datasets [12]. Hamerly has also discussed a

set of techniques that makes k-means even faster [13].

There are several methods available for determining the number of clusters. Two of the most well-known methods are the Bayesian Information Criterion (BIC) and the Akaike Information Criterion (AIC) measures that are used by the X-means algorithm [9]. The average silhouette of the data is another useful criterion for determining the number of clusters [14]. Rate distortion theory is another method that can be used to determine the number of clusters [15]. However, most of the mentioned works are both complex and usually inaccurate.

III. PRELIMINARIES

Let *D* be a set of *n* data items (also known as dataset), each having *d* features. Each data item can be represented by a D_i vector. A partitioned clustering algorithm should find a set $C=\{C_1,C_2,\ldots,C_k\}$ of clusters (partitions) as the output clustering. Each cluster should preserve these two conditions:

1) Each cluster should have at least one data item assigned, i.e., $\forall C_i \in C : C_i \neq \varnothing$

2) Every two different clusters should have no data item in common, i.e., $\forall C_i, C_j \in C : C_i \cap C_j = \varnothing$

These constraints imply a hard and partitioned clustering. Data items that are assigned to a cluster should be as similar as possible. Therefore, a similarity metric should be defined too. The most widely used similarity metric is the Euclidean distance. Euclidean distance between any two *d*-dimensional data items can be computed using:

$$d(\vec{D}_i,\vec{D}_j) = \left\|\vec{D}_i - \vec{D}_j\right\| = \sqrt{\sum_{p=1}^{d}(D_{i,p} - D_{j,p})^2} \quad (1)$$

Another property that should be defined here is the center of a cluster. Center (centroid) of a cluster is actually the mean of all data items in that cluster. Usually, for running the k-means algorithm on a set of data items, it is just sufficient to be able to compute the distance between each two pairs of data items and to compute the center of a set of data items. Center (centroid) of a cluster can be calculated using Equation 2.

$$Center(C_i) = \frac{1}{|C_i|}\sum_{D_j \in C_i} D_j \quad (2)$$

The error metric denoted by *E* that shows the quality a clustering is the standard Sum of Squared Error (SSE) function that is defined as:

$$E = \sum_{C_i \in clusters}\sum_{D_j \in C_i} D(D_j, Center(C_i))^2 \quad (3)$$

A. k-means and k-means++ algorithms

For better illumination of our proposed algorithm, it is necessary to know the principles of k-means and k-means++ algorithms. k-means is a simple and fast algorithm and its details are given in Algorithm 1.

Usually repeating steps 2 and 3 is bound to a fixed number in order to avoid infinite loop. The k-means++ algorithm is identical to k-means except the first step in which initialization of centers is done. k-means++ tries to choose a set of carefully selected initial centers instead of random initialization. If we assume *D(x)* to be the distance of a data item to its nearest center that is already chosen, then k-means++'s initialization step can be defined as Algorithm 2. In the rest of the paper these variables are used: *n* as the number of data items in dataset, *d* as the size (dimension count) of each data item, and *k* as the number of clusters.

Algorithm 1: k-means clustering algorithm

Input: Dataset *X*, and *k* as the number of clusters
Output: *k* centers

1. Randomly select a set of *k* initial centers $C=\{c_1, c_2, \ldots, c_k\}$.
2. Build a set of *k* clusters by assigning each data item in dataset to the nearest center to that data item.
3. Update each c_i to be the center of all points assigned to the cluster related to c_i using equation 2.
4. Repeat steps 2 and 3 until there are no changes in the set of centers.
5. Return the set of centers.

Algorithm 2**:** k-means++ centers initialization

Input: Dataset *X*, and *k* as the number of clusters
Output: *k* initial centers

1.1. Choose a center c_1 randomly from the dataset.
1.2. While the number of selected centers is less than *k* do:
1.3. Take a new center c_i, choosing an item *x* with probability of $\frac{D(x)^2}{\sum_{x \in dataset} D(x)^2}$

I. PROPOSED METHOD: BS-MEANS

In this section, we describe the BS-means algorithm. BS-means finds the correct number of clusters in a dataset using the k-means++ algorithm. For determining the right number of clusters, we cluster the dataset using the k-means++ algorithm, assuming different values as k in range of 1, 2, 4, 8, ., $\sqrt{n}$. We choose $\sqrt{n}$ as the upper bound for the number of clusters because considering information theory concepts the maximum number of clusters for a dataset is at most $\sqrt{n}$[16].

For each *k* we compute the error for that number of clusters. If we assume $E_{km}(k)$ as the error function of k-means algorithm with *k* clusters, we have $E_{km}(1), E_{km}(2), E_{km}(4), E_{km}(8), \ldots, E_{km}(\sqrt{n})$. Naturally the error should decrease as the number of the clusters increases. For getting a more stable error value we run k-means++ *t* times for each value of *k*

and assume the average value as $E_{km}(k)$. The correct number of clusters is a point where there is a very little decrease in the error value.

Note that, it is not necessary to compute E_{km} for all values between 1 and $\sqrt{n}$ and we just consider exponential values. Using the binary search algorithm, the correct point can be identified by just $O(\log n)$ steps. The details of BS-means algorithm are given in Algorithm 3.

In lines 1-5 of algorithm, we compute E_{km} values for different cluster numbers. In line 6, we find the elbow point which can be found using a linear search between each adjacent cluster numbers. In lines 7-13, we use binary search and test all of the values between the two consecutive points found in line 6. At last, the best cluster number is returned as the final number.

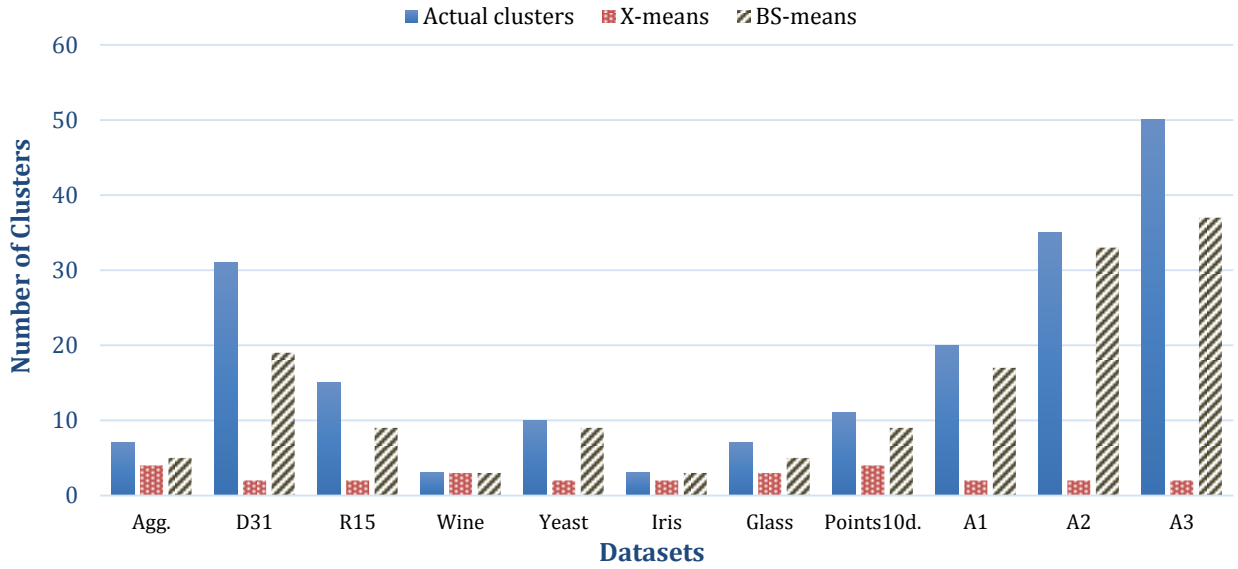

Figure 1. Comparison of the number of clusters

The complexity of BS-means is $O(t.d.n.\sqrt{n}.\log n)$, in which n is the number of the objects, k is the number of the clusters, d is the number of the dimension of each object, and t is the number of the iterations. Normally, wehave $k, t, d \ll n$. Since we set *t* equal to 3 in our setting and if we assume the distance between two items can be computed in *O*(1), we can state that the complexity of BS-means is $O(n.\sqrt{n}.\log n)$.

II. EVALUATIONS

We used a 2.2 GHz Intel core-i3 machine with 6 GB of RAM during the experiments. The BS-means algorithm is coded in Java. For X-means algorithm, we used the implementation available in Weka 3.6 toolkit.

A. Dataset Description

We use several datasets to evaluate the accuracy of BS-means. All of the datasets can be found in the UCI machine learning repository and the University of Eastern Finland's repository. The details of each dataset are given in Table I.

B. Results

We use two measures to evaluate BS-means algorithm. The first measure is accuracy which defined as the average ratio of the number of clusters determined by the algorithm to the number of actual clusters. The second criterion is speed, which is the average amount of time that an algorithm requires to solve a clustering problem regardless of the accuracy. We run each algorithm ten times for each dataset and reported the mode value.

Algorithm 3: BS-means algorithm

Input: A dataset D
Output: An Integer *k* as the number of clusters
1. **Let** *temp* be an empty list
2. **for** (*k*=1;*k*<=$\sqrt{D}$;*k*=*k**2)
3. calculateE_{km}(cluster)
4. addE_{km}(cluster) to *temp*
5. **endfor**
6. **find** two values $E_{km}(j)$ and $E_{km}(2j)$ in *temp* where there is very little difference between them.
7. **Repeat**
8. *i=j/2*
9. *z=(i+j)/2*
10. calculate $E_{km}(z)$
11. *g*=greatest value in $\{E_{km}(\mathrm{i}), E_{km}(\mathrm{z}), E_{km}(\mathrm{j})\}$points
12. **if** *g* =E_{km}(j/2)**then**j=z**else***i=z*
13. **until**$|i-j|=1$
14. **if** j==z **return** *j* **else return** i

Fig. 1 shows the actual number of clusters and the number of clusters produced by X-means and BS-means algorithms for all of the eleven datasets. The actual numbers of clusters are officially reported by dataset providers. We compare BS-means with X-

means, which is one of the most popular algorithms in this area.

As Fig. 1 shows, the number of clusters that are produced by BS-means algorithm is very close to the actual number of clusters than the number of clusters that are generated by X-means. In some cases such as Wine, and Iris datasets, the number of clusters reported by BS-mean equal to the actual number of clusters. The average accuracy criterion, which is calculated for both BS-means and X-means are shown in Table II. The accuracy of BS-means is very higher than X-means. When the actual number of clusters is smaller or equal than four, the output of X-means is not bad. But, X-means fails when *k* is large.

An important weakness of X-means is its accuracy. Considering the reported results in Fig. 1,we can state that X-means is designed for detection of small number of clusters. If the number of actual clusters is large, X-means usually fails to detect the correct number. Most of the time, x-mean reports '2' as the number of clusters and it seems that it is a local minimum for this algorithm. In contrast, BS-means show a more acceptable accuracy. For both large and small cluster numbers, BS-means performs better and it has no bias towards small number of clusters.

TABLE II. AVERAGE ACCURACY OF BS-MEANS AND X-MEANS

Algorithm	*Accuracy*
BS-means	0.808
X-means	0.329

In many data mining applications, speed is very important. Many of applications are real time, online and stream-based. Naturally, the speed of execution drops with increase of the input dataset. Hence, an efficient algorithm must be able to process large datasets effectively. We evaluated the performance of BS-means and X-means algorithms using speed criterion. Table III shows the execution time of both algorithms on all of the input datasets. There is no absolute winner between X-means and BS-means. For some datasets, BS-means is faster and for some datasets, X-means is faster.

TABLE I. DATASET PROPERTIES

Datasets	*Instances*	*Dimension*	*Clusters*
Aggregation	788	2	7
D31	3,100	2	31
r15	600	2	15
Wine	178	13	3
Yeast	1,484	8	10
Iris	150	4	3
Glass	214	9	7
points10dCCNorm	100,000	10	11
A1	3,000	2	20
A2	5,250	2	35
A3	7,500	2	50

TABLE III. EXECUTION TIME FOR BS-MEANS AND X-MEANS

Datasets	*X-means(ms)*	*BS-means(ms)*
Aggregation	110	102
D31	11	628
R15	54	56
Wine	61	31
Yeast	34	340
Iris	15	31
Glass	55	31
Points10dCCNorm	273,564	168,932
A1	88	606
A2	270	1,492
A3	450	2,258

An interesting point is the performance of X-means and BS-means on the *Points10dCCNorm* dataset which is the largest input dataset. For this dataset, BS-means executes much faster than X-means. We also tested other large datasets such as the *KDDCUP04Bio* dataset and in all cases, X-means was slower than BS-means. Hence, we can state that BS-means is more suitable for larger datasets.

As a conclusion for this section, we can say that considering the experiments done for evaluation of accuracy and speed of BS-means and X-means, BS-means is a better solution. Since, there is no significant difference between speed of X-means and BS-means. On the other hand, the accuracy of BS-means is far better than X-means and in all of the evaluated datasets, X-means was biased in favor of small cluster numbers, while BS-means had not such a shortcoming.

III. CONCLUSION AND FURTHER WORK

Determining the number of clusters of a dataset is a very challenging problem which is usually decided by an expert. However, most of the available clustering algorithm like k-means, cannot determine the number of clusters automatically and the number must be given by the user. To overcome this problem, we presented a novel algorithm called BS-means which is able to automatically determine the number of clusters of a dataset without any user given information. BS-means uses a binary search based approach to detremine the number of clusters. To show the effectiveness of BS-means we made several experiments of various datasets and our experiments showed that BS-means is more effective and accurate than competiting solutions like X-means. For furthur works we plan to make BS-means more scalable and enable it to process very large datasets. Our goal is to fit BS-means into the MapReduce programming model and the prelimiray results are encouraging. Another opportunity is to tailor BS-means according to datasets with specific charactristics like 2-dimentional datasets, or image datasets.

REFERENCES

[1]A. K. Jain, “Data clustering: 50 years beyond K-means,” *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.

[2]H. Zhang, J. E. Fritts, and S. A. Goldman, “Image segmentation evaluation: A survey of unsupervised methods,” *Comput. Vis. Image Underst.*, vol. 110, no. 2, pp. 260–280, 2008.

[3]R. Baeza-Yates, B. Ribeiro-Neto, and others, *Modern information retrieval*, vol. 82. Addison-Wesley New York, 1999.

[4]D. J. Miller, Y. Wang, and G. Kesidis, “Emergent unsupervised clustering paradigms with potential application to bioinformatics.,” *Front. Biosci.*, vol. 13, pp. 677–690, 2008.

[5]R. Xu, D. Wunsch, and others, “Survey of Clustering Algorithms,” *IEEE Trans. Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[6]A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: a review,” *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999.

[7]G. Gan, C. Ma, and J. Wu, *Data Clustering: Theory, Algorithms, and Applications*. Society for Industrial & Applied Mathematics Publishing, 2007.

[8]J. MacQueen and others, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 1967, vol. 1, no. 14, pp. 281–297.

[9]D. Pelleg, A. Moore, and others, “X-means: Extending k-means with efficient estimation of the number of clusters,” in *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000, vol. 1, pp. 727–734.

[10]D. Arthur and S. Vassilvitskii, “k-means++: the advantages of careful seeding,” in *Proceedings of the 18th annual ACM-SIAM symposium on Discrete algorithms*, 2007, pp. 1027–1035.

[11]B. Bahmani, B. Moseley, A. Vattani, R. Kumar, and S. Vassilvitskii, “Scalable k-means++,” *Proc. VLDB Endow.*, vol. 5, no. 7, pp. 622–633, Mar. 2012.

[12]A. Hadian and S. Shahrivari, “High performance parallel k-means clustering for disk-resident datasets on multi-core CPUs,” *J. Supercomput.*, pp. 1–19, 2014.

[13]G. Hamerly, “Making k-means even faster,” in *SIAM International Conference on Data Mining (SDM)*, 2010.

[14]P. J. Rousseeuw, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,” *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, 1987.

[15]C. A. Sugar and G. M. James, “Finding the number of clusters in a dataset,” *J. Am. Stat. Assoc.*, vol. 98, no. 463, 2003.

[16]K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate analysis*. Academic press, 1979.