International Journal of Data Warehousing and Mining
Volume 19 • Issue 2

Estimating the Number of Clusters in High-Dimensional Large Datasets

Xutong Zhu, Heilongjiang University, China*
Lingli Li, Heilongjiang University, China

ABSTRACT

Clustering is a basic primer of exploratory tasks. In order to obtain valuable results, the parameters in the clustering algorithm, the number of clusters must be set appropriately. Existing methods for determining the number of clusters perform well on low-dimensional small datasets, but how to effectively determine the optimal number of clusters on large high-dimensional datasets is still a challenging problem. In this paper, the authors design a method for effectively estimating the optimal number of clusters on large-scale high-dimensional datasets that can overcome the shortcomings of existing estimation methods and accurately and quickly estimate the optimal number of clusters on large-scale high-dimensional datasets. Extensive experiments show that it (1) outperforms existing estimation methods in accuracy and efficiency, (2) generalizes across different datasets, and (3) is suitable for high-dimensional large datasets.

KEYWORDS

Clustering, Dimensionality Reduction, High-Dimensional Space, Locality-Sensitive Hashing, Sampling

INTRODUCTION

Clustering is the main task of exploratory data mining and a common technique for statistical data analysis. Clustering is widely used, in addition to data mining, pattern recognition, image processing (Bhatia & Deogun, 1998), computer vision (Frigui & Krishnapuram, 1999; Shi & Malik, 2000), and other fields; it is also used in fraud detection, market segmentation, and many other aspects. For example, in fraud detection, outliers in the cluster may predict the existence of fraud. In e-commerce, clustering can help e-commerce enterprises to understand their customers and provide them with more appropriate services by grouping customers with similar browsing behaviors and analyzing their common characteristics (Punj & Stewart, 1983). Clustering has been identified in these and more areas. Clustering methods are used to describe data, measure the similarity between different data points, and classify data points into different clusters.

The *k-means* algorithm is widely used for clustering due to its excellent performance in terms of runtime in practical applications (Chiang & Mirkin, 2010; Dunn, 1974). *k-Means* (Celebi et al., 2013; Jain, 2010; Wang et al., 2020) aims to partition a dataset with *n* entities into *k* ($k < n$) clusters, where each entity is assigned to the closest cluster to the centroid. When performing the *k-means* algorithm, the number of clusters must be specified. However, the optimal number of clusters in a dataset is often unknown and the k-value is difficult to estimate and give in advance. Setting an inappropriate number of clusters when performing clustering algorithms can lead to structural,

DOI: 10.4018/IJDWM.316142 *Corresponding Author

grouping or compression errors. Most of the existing methods for estimating the optimal number of clusters focus on low-dimensional and small datasets.

The elbow method (Thorndike, 1953) is a commonly used and intuitive method to estimate the number of clusters in a dataset. It performs *k-means* algorithm for each *k* in a given search space, then uses the sum of squared errors (SSE) (Arbelaitz et al., 2013) to draw the elbow graph, and determines the optimal number of clusters by judging the inflection point of the elbow graph. As the number of cluster *k* increases, the compactness of each cluster gradually increases, so the SSE gradually decreases. When *k* reaches the optimal number of clusters, the reduction of SSE decreases sharply, and then tends to be flat as the value of *k* continues to increase. The SSE is the sum of the distances of all entities to their cluster centers. The elbow method uses the SSE as the measure of clustering effectiveness because it can be derived directly from the *k-means* algorithm. However, the elbow method also has obvious downsides. First, it has to perform the *k-means* algorithm for each value of *k* in the search space R to get the SSE, which can be very computationally expensive for large datasets or high- dimensional datasets because it takes a lot of time to calculate the SSE before an estimate can be made. Secondly, the optimal number of clusters is not well determined if there is no obvious bend in the elbow graph or if there are multiple bends.

As previously stated, the elbow method has serious drawbacks, which makes it difficult to estimate the optimal number of clusters in a dataset using the elbow method in practical applications. Fritz et al. (2020) proposed LOG-Means as a new estimation method that exploits the valuable properties of the elbow rule to estimate the optimal number of clusters in a dataset by a specialized search strategy. LOG-Means estimates the optimal number of clusters in the dataset by calculating the SSE ratio of nondirectly adjacent *k* values in the search space. Although LOG-Means provides an efficient search strategy, it also has serious drawbacks. First, LOG-Means performs the *k-means* algorithm directly on the data, which suffers from the curse of dimensionality (Giraud, 2021) when the data points are high-dimensional, which, in turn, affects the SSE calculation. Second, the *k-means* algorithm is expensive to execute in a single pass on high-dimensional data, especially for large high-dimensional datasets (Arthur & Vassilvitskii, 2006).

Most of the existing methods for estimating the optimal number of clusters focus on small data sets in low dimensions, and they do not perform well on high-dimensional large-scale data sets (e.g., when calculating the SSE, which is affected by the curse of dimensionality because the data points are high-dimensional). In this paper, the author aims to investigate how to quickly and accurately estimate the optimal number of clusters on a dataset while reducing the computational cost of high-dimensional large-scale data. The author designs a new method, based on locality-sensitive hashing (LSH) and named LSH-Means, to estimate the optimal number of clusters in high-dimensional large datasets, which overcomes the inadequacies of existing methods in high-dimensional space estimation. LSH-Means aims to reduce the dimensionality (Fodor, 2002) and scale of the dataset (Fern, 2003; Meng, 2013; Xie et al., 2017) before estimation while maintaining the similarity in the original dataset..; Then, LSH-Means utilizes the search strategy of LOG-means to effectively estimate the optimal number of clusters in high-dimensional large datasets.

The contributions of this paper are as follows:

- The author proposes their approach LSH-Means and discusses it by comparing it with LOG-Means. They specify how their method estimates the optimal number of clusters in high-dimensional large datasets.
- The author analyzes LSH-Means and proves its effectiveness. It has strong fitting ability for high-dimensional large datasets.
- The authors' experimental results show that LSH-Means outperforms LOG-Means in terms of runtime while maintaining accuracy.

The rest of the paper is structured in five sections as follows: The second section provides the literature review; the third section introduces the preliminaries and problem definition; the fourth section details the author's approach LSH-Means; the fifth section illustrates the author's experimental results and offers a comparison with LOG-Means and the elbow method; finally, the sixth section summarizes the author's work.

RELATED WORK

Existing estimation methods follow three steps: (1) Determine which parameter in the search space R is to be executed next; (2) execute the clustering algorithm using the determined parameters; (3) evaluate the results. Since *k-means* performs well in practical applications, it is usually used in the second step.

Estimation methods can be divided into exhaustive and nonexhaustive methods, where the exhaustive method performs a clustering algorithm for each k in the search space and the nonexhaustive performs a clustering algorithm for nonexhaustive search. The exhaustive algorithm performs a clustering algorithm for each k, evaluates the results based on the validity measure, and finally selects the best result as an estimate of k. The main difference between these methods is that the validity measures evaluated are different, and the validity measure of existing clusters is mainly the tightness of the clusters. Nonexhaustive methods search in the search space according to a certain pattern and stop the search immediately when almost no difference occurs in the evaluation criteria, such as X-Means and G-Means.

Existing methods are mainly used in low-dimensional spaces; in this paper, the author proposes LSH-Means, which reduces the execution of the clustering algorithm and the calculation of the SSE, while avoiding the effect of the curse of dimensionality. As a result, the optimal number of clusters can be estimated quickly and accurately on large high-dimensional datasets.

PRELIMINARIES AND PROBLEM DEFINITION

Before detailing on their approach, the author will cover the preliminaries required for this paper and problem definition.

LOG-Means

The purpose of LOG-Means is to find the k value at the maximum bend in the elbow graph; this value is also the optimal number of clusters. LOG-Means uses the property of elbow graph to find the k value of the elbow graph at the largest bend by binary search strategy. LOG-Means computes the SSE ratio of two k values that are not directly adjacent in the search space R (lines 8 to lines 12); it takes the two k values with the larger SSE ratio as the search space for the next search, iteratively calculates the SSE ratio, and compares the size of the SSE ratio (lines 13 to 18). When two k values are directly adjacent in the search space, the k value on the right is the k value with the maximum bending in the elbow graph, which is also the estimated optimal number of clusters. Algorithm 1 outlines the pseudocode of LOG-Means.

Algorithm 1. LOG-Means

Input: dataset X, $R = \left[k_{low}, k_{high}\right]$

Output: k_{est}.

1: к ¬φ

2: м ¬φ

3: SSE_{low} ← SSE obtained by performing *k-means* with k_{low}
4: K ← $k \cup \{(k_{low}, SSE_{low})\}$
5: SSE_{high} ← SSE obtained by performing *k-means* with k_{high}
6: K ← $k \cup \{(k_{high}, SSE_{high})\}$
7: while (k_{low} and k_{high} are not directly adjacent) do
8: $k_{mid} = \lfloor k_{low} + k_{high} /2 \rfloor$
9: SSE_{mid} ← SSE obtained by performing *k-means* with k_{mid}
10: K ← $k \cup \{(k_{mid}, SSE_{mid})\}$
11: $ratio_{low} = SSE_{low} / SSE_{mid}$
12: $ratio_{high} = SSE_{mid} / SSE_{high}$
13: M ← store or update $\{(k_{mid}, ratio_{low})\}$
14: M ← store or update $\{(k_{high}, ratio_{high})\}$
15: k_{high} ← k at the highest ratio from M
16: k_{low} ← left adjacent value of k_{high} from M
17: SSE_{high} ← SSE for k_{high} from M
18: SSE_{low} ← SSE for k_{low} from M
19: k_{est} ← k with highest ratio in M
20: return k_{est}

LOG-Means is a nonexhaustive method which determines the search space for the next search by comparing SSE_{low} / SSE_{mid} and SSE_{mid} / SSE_{high}, where SSE_{mid} is the SSE at $k_{mid} = k_{low} + k_{high}$. When the former is greater than the latter, the search space becomes $R = \lfloor k_{low}, k_{high} = k_{mid} \rfloor$, and when the latter is greater than the former, the search space becomes $R = \lfloor k_{low} = k_{mid}, k_{high} \rfloor$. The above operations are iteratively performed until two values of *k* are directly adjacent in the search space. LOG-Means denotes the value for *k* with the highest SSE ratio of k_{low} and k_{low} of last iteration as k_{est} and the estimated number of clusters for the dataset as k_{est}.

Locality-Sensitive Hashing

LSH technology is a method Indyk and Motwani (2012) proposed to find high-dimensional similarity patterns based on available data volume linear dependence. It is a probability-based method for searching nearest neighbors in high-dimensional space. The basic idea is to hash data points through a set of hash functions, so that data points that are close to each other in the original data space have higher similarity in the new space. The definition of similarity is determined according to the specific applications, and, for different similarity measurement methods (Datar et al., 2004; Gorisse et al., 2011; Lv et al., 2007; Paulevé et al., 2010; Sun et al., 2014; Slaney et al., 2012), the hash function of LSH is also different. No matter which algorithm, it reduces the high-dimensional data to the low-dimensional data, while keeping the similarity of the original data unchanged to a certain extent. LSH is defined as follows:

A family of hash functions $H = \{h_1, \ldots, h_1\}$ is called (r_1, r_2, p_1, p_2)-sensitive for distance measure function D if it satisfies the following two conditions:

- If $D(v,q) \leq r_1$ then $Pr_H[h(q) = h(v)] \geq p_1$.

- If $D\left(v,q\right)\geq r_2$ then $Pr_H\left[h\left(q\right)=h\left(v\right)\right]\leq p_2$.

To make sure that the set of LSH functions is valid, its parameters must satisfy the inequality $p_1 > p_2$ and $r_1 < r_2$.

Problem Definition

The *k-means* algorithm clustering has been widely studied by extension and applied in various substantive fields. With the development of technology, data collection becomes easier, leading to larger and more complex databases, such as various types of trade transaction data, Web documents, and gene expression data. Their attributes can often reach hundreds, even thousands, of dimensions. However, due to the influence of "dimensional curse," many clustering algorithms that perform well in low-dimensional data space often fail to achieve good clustering effect when applied to high-dimensional space.

k-Means clustering algorithm requires the number of clusters in advance. Generally speaking, the number of clusters is unknown. In this case, validity measure to assess the quality of individual clustering results, such as Bayesian information criterion, Akaike information criterion, Dunn index, silhouette coefficient, and SSE, can be used to find a cluster number. However, the cost and reliability of calculating these effectiveness metrics on high-dimensional data are unacceptable. The aim of this study is to reduce the computational cost of the SSE for high-dimensional data while ensuring the reliability of the SSE, and then to accurately estimate the optimal number of clusters for the original high-dimensional dataset by the value of *k* at the maximum change of the SSE in the elbow graph.

LOG-Means provides valuable properties that the author can exploit, and, since these properties are valid independently of the dataset, the author's approach LSH-Means utilizes these properties to maintain generality. LOG-Means formalizes the decrease of the SSE as $SSE_{ratiok} = SSE_{k-1} \,/\, SSE_k$. This ratio can be exploited to investigate the trend of the SSE variation in the entire search space. The *k* value at the maximum SSE_{ratio} represents the most curved place in the elbow diagram, that is, the optimal *k* value. Table 1 summarizes the symbols the author frequently used in this paper.

Table 1. Symbols

Symbols	Meaning
$X=\left\{x_1,\ldots,x_N\right\}\in\mathbb{R}^d$	d-dimensional dataset with N entities
$D=\left\{v_1,\ldots,v_N\right\}\in\mathbb{R}^l$	l-dimensional dataset with N entities
$S=\left\{s_1,\ldots,s_N\right\}\in\mathbb{R}^l$	Dataset after sampling
φ	Sample rate
$R=\left[k_{low},k_{high}\right]$	Search space
SSE	The sum of squared errors
k_{est}	Estimated number of the cluster

Example

The SSE decreases gradually with increasing *k* values in the elbow diagram, and decreases sharply when *k* reaches the optimal number of clusters, so the author's problem transforms from finding the *k* value at the most curved point of the SSE in the elbow diagram to finding the *k* value at the maximum SSE ratio.

Cluster Number Estimation Problem

In a d-dimensional dataset $X = \left\{x_1, \ldots, x_N\right\} \in$ with *n* entities, the cluster number estimation aims to find the k_{est} value for X, where $k_{est} = \arg\max_k SSE_{k-1} / SSE_k$. Intuitively, k_{est} is the optimal number of clusters for X since it represents the maximum bend in the elbow graph of X.

THE AUTHOR'S APPROACH: LOCALITY-SENSITIVE HASHING MEANS

As the author discussed above, when LOG-Means estimates optimal number of clusters in a dataset, it directly uses the high-dimensional data to perform the *k-means* algorithm to obtain the SSE. When *k-means* processes massive high-dimensional data, calculating the distance from a single point to the central point is very expensive. If there are *n* data points, *k* center points, the data dimension is *d*, the distance calculation cost of *k-means* is $O\left(n * k * d\right)$. Therefore, in order to reduce the complexity of the comparison calculation, the author adopted the method of reducing the data dimension and data scale to obtain the sample dataset. Then, the author used the logarithmic search principle of LOG-Means to find the number of clustering in the sample dataset, which they used as the optimal number of clustering in the original dataset.

The LSH-Means method the author proposed in this paper consists of the following two main steps:

1. The original dataset is dimensionally reduced using a LSH function, and the sample dataset is obtained by sampling the reduced dataset.
2. The sample dataset is obtained as k_{est}' using the search strategy provided by LOG-Means, and k_{est}' is estimated as the optimal number of clusters k_{est} in the original dataset.

Formally, each hash function $h_{\boldsymbol{a},b}\left(x\right): \mathbb{R}^d \to \mathrm{N}$ maps a *d* dimensional vector *x* to an integer, where *a* is, with the same dimensions as the vector *x*, a *d*-dimensional vector with each dimension chosen independently from Gaussian distribution and *b* is a real number chosen independently from the range $\left[0, r\right]$. For a fixed *a*, *b*, the author sets the hash function as $h_{\boldsymbol{a},b}\left(x\right) = \frac{ax + b}{r}$. The function of *r* is to divide a line into segments of equal length and length *r*, assign the same hash value to points mapped to the same segment, and assign different hash values to points mapped to different segments.

Each hash function $h_{a,b}$ can map the eigenvector *x* to a real number, and we use *l* such hash function $h_{a,b}$. We use *l* such hash functions, so that the vector *x* is projected into the *l* -dimensional space. The coordinates of the vector *x* in the *l* -dimensional space are the real numbers mapped by the *l* hash functions.

The above dimensional reduction is performed on each vector in the original dataset $X = \left\{x_1, \ldots, x_N\right\} \in \mathbb{R}^d$ to obtain the *l*-dimensional dataset $D = \left\{v_1, \ldots, v_N\right\} \in \mathbb{R}^l$. Next, dataset D is sampled to reduce the size of the data. The sample dataset $S = \left\{s_1, \ldots, s_N\right\} \in \mathbb{R}^l$ is obtained by simple random sampling of entities in dataset D with proportion φ. Algorithm 2 shows the

pseudocode of sample dataset S obtained by dimensionality reduction. The time complexity of Algorithm 2 is $O\left(N * l\right)$.

Algorithm 2. Dimensionality reduction and sampling to get a sample dataset

```
Input: dataset X, number of hash functions l, sampling rate φ,
search space R
Output: k_est
1: D ¬φ
2: S ¬φ
3: a_i : { a_i ∈ℝ^d , 1≤i≤l } // randomly
generate l d-dimensional vector a
4: b_i : { b_i , 1≤i≤l } // randomly
generate l real number b from [0,r]
5: for x in X do
6:   for a in a_i do
7:       h_{a,b}(x) = (ax+b)/r // b is random
selected in b_i
8:     v_i = (h_1,...,h_l, 1≤i≤l)
9: D ¬ D ∪ v_i
10: S ¬ sample D at scale φ
11: k_est ¬ execute LOG-Means with dataset S
12: return k_est
```

The estimation method involves two steps: (1) Dimensionality reduction of the original dataset, and sampling to obtain a sample dataset; (2) the LOG-Means search strategy is used to estimate the optimal number of clusters in the sample dataset. In high-dimensional data space, due to the influence of the curse of dimensionality, existing methods often cannot accurately estimate the optimal number of clusters in a dataset. Furthermore, the larger the data dimension, the more expensive the evaluation step, which may lead to worse complexity. Therefore, existing methods require a huge overall runtime before an estimate can be made. The aim of the author's method LSH-Means is to find the value of k at the maximum point of bending in the elbow graph; LOG-Means provides an effective strategy to find that point by comparing the SSE ratios of k_{low} and k_{mid}, k_{mid} and k_{high} in the search space to determine the value of k at the maximum point of bending in the elbow graph. LOG-Means finds the value of k_{est} by searching the original high-dimensional large-scale dataset which incurs high overhead of computation. In contrast, the author's method performs the LOG-Means to estimate the value of k_{est} on a low-dimensional small-sized sampled dataset, which outperforms LOG-Means significantly in terms of efficiency while only incurring little accuracy loss. The author guarantees that LSH-Means performs better in running time. The author will analyze the complexity of their method from two aspects:

1. Dimensionality can be reduced in $O\left(l\right)$. Since the dot product of l vectors needs to be calculated, each dot product is just an arithmetic operation, so the complexity of each dot product is $O\left(1\right)$. Simple random sampling is used to sample the dimensionality-reduced dataset, so the complexity is $O\left(1\right)$.

2. The LOG-Means logarithmic search strategy is used to estimate the optimal number of clusters in the sample dataset, so only $log|R|$ times the *k-means* algorithm needs to be performed, with a complexity of $O(log|R|)$. The only evaluation is $log|R|$ times of clustering results, the complexity of the evaluating results is $O(log|R|)$.

   In conclusion, the overall complexity of LSH-Means is $O(l+1+log|R|+log|R|)$.

EVALUATION

The purpose of the author's evaluation is to compare LSH-Means with the LOG-Means method to estimate the optimal number of clusters in high-dimensional large datasets and their running time and accuracy. Since the author's method exploits some properties of the elbow method, they also compare the elbow graphs before and after dimensionality reduction.

This section is structured as follows: First, the author discusses experimental setup; Subsequently, the author shows the changing trend of the elbow graph of the dataset before and after dimensionality reduction on the synthetic dataset; finally, the author shows the running time and accuracy of their method.

Hardware and Software

All experiments were performed on a PC with 2.6GHz 4-core AMD processor and 16GB RAM running the Linux Ubuntu 20.4. The author's algorithms run entirely in main memory. They installed Python 3.6.

The author's approach takes into account larger datasets and higher dimensions. For this purpose, they used synthetic datasets to validate their approach. Table 2 depicts the characteristics of the 10 datasets the author used for evaluation, where *N* is the number of entities in the dataset, *d* is the dimension of dataset, and *c* is the optimal number of clusters. The dataset has value between [-10,10] for each dimension. Each cluster follows a Gaussian distribution with the mean at the center and a standard deviation of 0.5. The *c* centers are chosen randomly and the clusters are nonoverlapping.

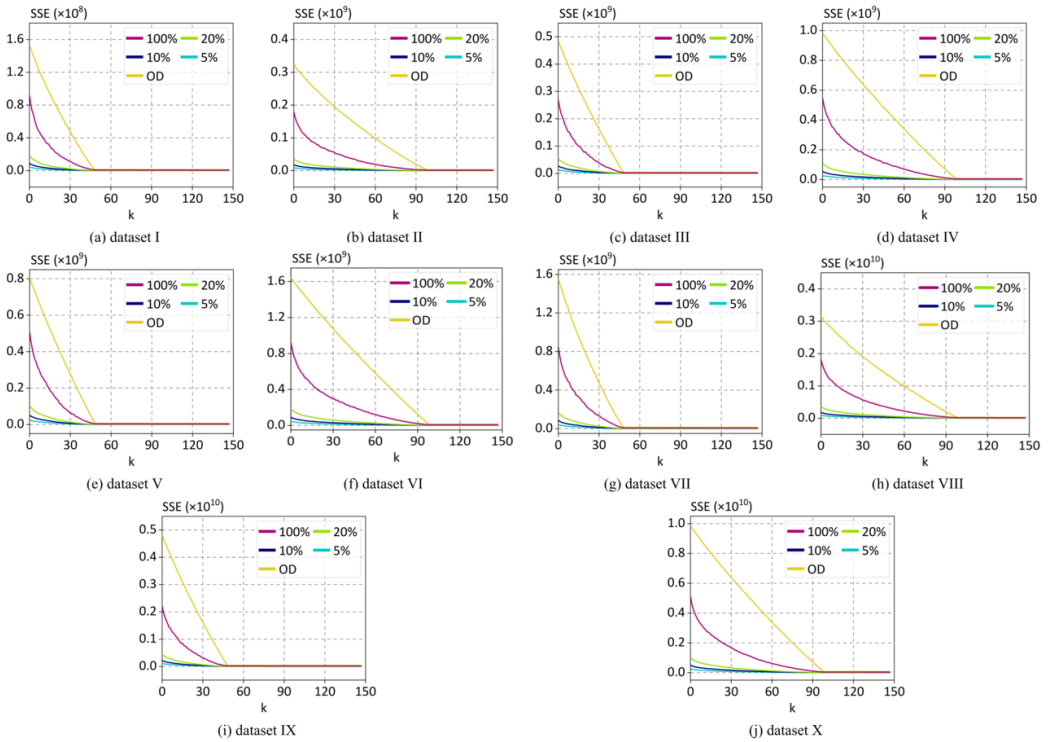
Table 2. Characteristics of the 10 Synthetic Datasets

Dataset	*N*	*d*	*c*
I	50,000	100	50
II	100,000	100	100
III	50,000	300	50
IV	100,000	300	100
V	50,000	500	50
VI	100,000	500	100
VII	500,000	100	50
VIII	1,000,000	100	100
IX	500,000	300	50
X	1,000,000	300	100

Experimental Results

Figure 1 shows the elbow graphs on the synthetic dataset (OD) and the elbow graphs on the dataset sampled at different scales after dimensionality reduction, which allows to clearly observe the trend of the SSE. Indeed, the SSE is gradually decreasing with the increase of the number of clusters k in the horizontal coordinate. After reaching the optimal number of clusters, the SSE does not change much. The curves with different colors in Figure 1 indicate the changes of the SSE with the increase of the number of clusters k at different sampling ratios. The reduced dimensional dataset can find the optimal number of clusters from the change of the SSE, indicating that the author's reduced dimensional dataset can be used to execute the following algorithm instead of the original dataset.

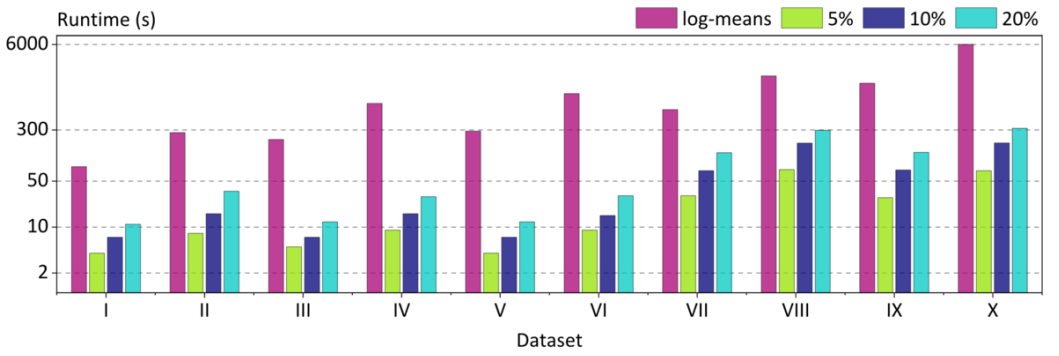
Figure 1. Elbow Graphs Before and After Dimensionality Reduction on Synthetic Datasets



Runtime

Figure 2 shows the runtime for estimation method over all datasets. It evidences that, with the increase in the size and dimension of the dataset, the estimation time of the LOG-Means method also increases, and it reaches 6026s when it is executed on the dataset X. This makes the estimation time even more unacceptable for larger and higher dimensional datasets. The author's method greatly reduces the estimation time. In terms of dataset X, the author's method only takes 318s. Therefore, this method is more efficient on large and high-dimensional data.

Figure 2. Runtime Across All Synthetic

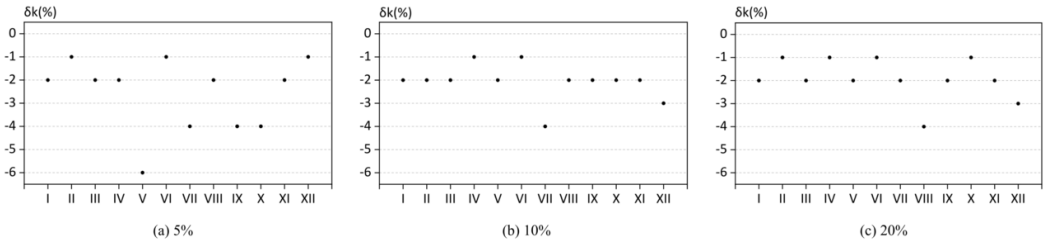


Accuracy

Since the author used synthetic datasets, the researcher is able to know in advance the actual number of clusters for each dataset. They take advantage of this to set relative error $\delta k = \left(k_{est} - c\right) / c * 100$, where k_{est} denotes the *k* value estimated by their estimation method. The author compared the obtained estimates with the actual number of clusters for all datasets and analyzed the error between them. The setting of relative error allows to better judge the accuracy of the author's method.

As Figure 3 shows, the author's estimate of the optimal number of clusters is more stable as the sampling rate increases. In addition, although the error rate of the method estimate is higher at a sampling rate of 5% than the error rate at a sampling rate of 10% on dataset II and XII, large fluctuations in estimation error are due to low sampling rate. Nevertheless, Figure 3 shows that the average relative error of the LSH-Means method is about -2%, thus providing a very accurate overall estimate.

Figure 3. Accuracy at Different Sampling Ratios



Evaluation on Real-World Data

Although the author conducted previous experiments on synthetic datasets, they also investigated the effect of LSH-means on real-world datasets. The author used three real-world classification datasets from the UCI machine learning library. When using these data, the author removes all nonnumeric and symbolic values, ids, timestamps, empty labels, and null values. Table 3 summarizes the characteristics of these datasets.

Table 3. Real-World Datasets and Their Characteristics

Abbr.	Dataset	N	d	c
A	MNIST	60,000	784	10
B	KDD Cup 1999 Data	4,898,431	34	23
C	KITSUNE	21,017,597	115	10

The results in Table 4 show that LSH-means is the fastest and most accurate method compared with elbow method and LOG-Means.

Table 4. Relative Error δk and Overall Runtime

Est. Method	δk (%)			Runtime(s)		
	A	B	C	A	B	C
ELM	-50	-13	-40	1995	1076	6026
LOG-Means	-40	0	-25	210	139	783
LSH-Means	-20*	0	-10*	72*	35*	137*

Note: * indicates best results per dataset.

Concluding, the author's experiments reveal that LSH-means can also achieve accurate and fast estimation on real-world datasets, and often outperforms existing estimation methods in estimating the number of clusters in a dataset.

Effect of Parameters

The author evaluated the performance of their method when varying the values of three different parameters. The researcher only reports the results on dataset II because similar results are observed on other datasets.

Sampling Ratio φ

The author investigated the estimation effect of sampling the dimensionality-reduced dataset at different scales. It is clear that the larger the sampling scale is, the more accurate the author's estimation is. Balancing accuracy and running time, they found that a sampling ratio of $\varphi = 10\%$ works well for valid results.

Setting of Segment Length r in Hash Function

When dimensionality reduction is performed, the difference of the author's segmentation will affect their hash result, resulting in the change of the vector coordinates after dimensionality reduction. Without loss of generality, the author set their r to $r = 1$.

The Number of Hash Functions l

In order to check the effect of data dimension on k estimation after dimensionality reduction, the author first evaluated the impact of the number of hash functions, that is, the dimension after dimensionality reduction, on the accuracy. To this end, the researcher experimented with variations

of $l = 5,10,20$ on dataset II. Table 5 shows the estimation accuracy of different hash number under different sampling ratios. The author found that the larger l, the higher the accuracy can be achieved. For example, at $l = 20$, our error δk is -1% at the sampling ratio $\varphi = 10\%$, and at $l = 10$, $\delta k = -2\%$. On the other hand, when l is larger, although higher accuracy can be obtained, the improvement of accuracy is limited. Therefore, the author set $l = 10$ by default.

Table 5. Accuracy of Different Hash Numbers Under Different Sampling Ratios

l	δk (%)		
	φ		
	5	10	20
5%	-4	-2	-2
10%	-2	-2	-1
20%	-2	-1	-1

CONCLUSION

In this paper, the author proposed LSH-Means, an efficient method based on location-sensitive hashing and sampling, which is well suited for estimating the number of clusters in high-dimensional large-scale datasets, using the properties of elbow methods and the search strategy of LOG-Means logarithmic search. Experiments on several large-scale high-dimensional datasets show that LSH-Means improves the estimation time by a factor of several or even 10 compared with the fastest existing LOG-Means method, and surpasses LOG-Means method in terms of estimation accuracy. Also, the author's method outperforms on real-world datasets, doubling in precision and improving at least three times in runtime compared to the LOG-Means method.

REFERENCES

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., & Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, *46*(1), 243–256. doi:10.1016/j.patcog.2012.07.021

Arthur, D., & Vassilvitskii, S. (2006). How slow is the k -means method? *Proceedings of the twenty-second annual symposium on Computational geometry*. doi:10.1145/1137856.1137880

Bhatia, S. K., & Deogun, J. S. (1998). Conceptual clustering in information retrieval. *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics*, *28*(3), 427–436. doi:10.1109/3477.678640 PMID:18255959

Celebi, M. E., Kingravi, H. A., & Vela, P. A. (2013). A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Systems with Applications*, *40*(1), 200–210. doi:10.1016/j.eswa.2012.07.021

Chiang, M. M. T., & Mirkin, B. (2010). Intelligent choice of the number of clusters in k-means clustering: An experimental study with different cluster spreads. *Journal of Classification*, *27*(1), 3–40. doi:10.1007/s00357-010-9049-5

Datar, M. (2004). Locality-sensitive hashing scheme based on p-stable distributions. *Proc. of the 20th ACM Symposium on Computational Geometry*. doi:10.1145/997817.997857

Dunn, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, *4*(1), 95–104. doi:10.1080/01969727408546059

Fern, X. Z., & Brodley, C. E. (2003). Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach. In *Proceedings of the Twentieth International Conference (ICML 2003)*. AAAI Press.

Fodor, I. K. (2002). *A survey of dimension reduction techniques*. Lawrence Livermore National Lab. doi:10.2172/15002155

Frigui, H., & Krishnapuram, R. (1999). A robust competitive clustering algorithm with applications in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *21*(5), 450–465. doi:10.1109/34.765656

Fritz, M., Behringer, M., & Schwarz, H. (2020). *LOG-Means: efficiently estimating the number of clusters in large datasets*. Very Large Data Bases. VLDB Endowment. doi:10.14778/3407790.3407813

Giraud, C. (2021). *Introduction to high-dimensional statistics*. Chapman and Hall/CRC. doi:10.1201/9781003158745

Gorisse, D., Cord, M., & Precioso, F. (2011). Locality-sensitive hashing for chi2 distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(2), 402–409. doi:10.1109/TPAMI.2011.193 PMID:21968915

Indyk, P. (1998). Approximate Nearest Neighbor: Towards Removing the Curse of Dimensionality. *Proc Symposium on Theory of Computing*. doi:10.1145/276698.276876

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

Meng, X. (2013). Scalable Simple Random Sampling and Stratified Sampling. In *International Conference on Machine Learning*. JMLR.org.

Paulevé, L., Jégou, H., & Amsaleg, L. (2010). Locality sensitive hashing: A comparison of hash function types and querying mechanisms. *Pattern Recognition Letters*, *31*(11), 1348–1358. doi:10.1016/j.patrec.2010.04.004

Punj, G., & Stewart, D. W. (1983). Cluster analysis in marketing research: Review and suggestions for application. *JMR, Journal of Marketing Research*, *20*(2), 134–148. doi:10.1177/002224378302000204

Qin, L., Josephson, W., Zhe, W., Charikar, M., & Kai, L. (2007). Multi-probe LSH: Efficient indexing for high-dimensional similarity search. In *International Conference on Very Large Data Bases*. VLDB Endowment.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(8), 888–905. doi:10.1109/34.868688

Slaney, M., Lifshits, J., & He, J. (2012). Optimal parameters for locality-sensitive hashing. *Proceedings of the IEEE*, *100*(9), 2604–2623. doi:10.1109/JPROC.2012.2193849

Sun, Y., Wei, W., Qin, J., Ying, Z., & Lin, X. (2014). SRS: Solving c-Approximate Nearest Neighbor Queries in High Dimensional Euclidean Space with a Tiny Index. In *Proceedings of the VLDB Endowment*. VLDB Endowment.

Thorndike, R. (1953). Who belongs in the family? *Psychometrika*, *18*(4), 267–276. doi:10.1007/BF02289263

Wang, S., Sun, Y., & Bao, Z. (2020). *On the Efficiency of K-Means Clustering: Evaluation*. Optimization, and Algorithm Selection. doi:10.14778/3425879.3425887

Xie, H., Li, J., & Xue, H. (2017). *A survey of dimensionality reduction techniques based on random projection*. Academic Press.

Lingli Li received the B.S., M.S., and Ph.D. degrees from the Harbin Institute of Technology, China, in 2008, 2010, and 2015, respectively. She is currently an associate professor with the School of Computer Science and Technology, Heilongjiang University, China. She has published over 20 research papers in refereed international journals and conference proceedings, such as TKDE, Information Sciences, CIKM, DASFAA, in the areas of databases and data mining. Her research interests include data quality and big data management.