Seeded Center Initialization In k-Means Clustering

Masum Billal BILLALMASUM93@GMAIL.COM
Farhad Naeem FARHAD.ALAM@SHOHOZ.COM
Data Science Department
Shohoz
Bangladesh

Editor: Not assigned

Abstract

k-Means is a popular clustering algorithm that aims to reduce the sum of minimum squared distances from data points to the centers. Now a days most of the times seeded centers are used instead of choosing all uniformly at random. k-Means++ chooses the centers with a so called D^2 weighting. However, that is only one of the ways. Similar ideas are possible for seeding the centers. In this paper, we improve k-Means++ by choosing the first center with a probability instead of selecting a random point and we analyze another way of initializing the centers. Empirical evidence shows that our proposed algorithms perform better than available methods of center initialization consistently.

1. Introduction

Widely regarded as the most popular clustering techniques, k-Means remains a humble interesting topic in machine learning as well as computational geometry. Roughly the problem is: given a set of points $\mathbf{X}$ in $\mathbb{R}^d$. Find a set of centers $\mathcal{C}$ such that the function *inertia*

$$\mathcal{I} = \sum_{\mathbf{x} \in \mathbf{X}} \min_{\mathbf{c} \in \mathcal{C}} (\|\mathbf{c} - \mathbf{x}\|^2)$$

is minimum where $\|\cdot\|$ is the L_2 norm[1].

Default k-Means algorithm starts with random centers and then converge based on minimum distances of the centers from the data points. New centers are calculated based on the centroid. This is known as Lloyd's algorithm (Lloyd, 1982). We repeat this process until no more change is possible. Ostrovsky et al. (2006) and Arthur et al. (2007) take it one step further by choosing the initial centers with a probability. We intend to introduce other ways of initialization.

2. Related Work

There has been multiple surveys on k-Means in the literature. Probably the most relevant work in this regard is done by Celebi et al. (2013). However, most of the algorithms used in

1. $\|c - x\|$ or L_2 norm of $\mathbf{c} - \mathbf{x}$ is the distance between the center $\mathbf{c}$ and point $\mathbf{x}$ or the magnitude of the vector $\mathbf{c} - \mathbf{x}$.

that paper are not used practically very much. It was also noted by the authors themselves that k-Means++ and its greedy version work better than most. They also mention that probabilistic algorithms perform better than deterministic ones. Moreover, another highly influential center initialization algorithm (Ostrovsky et al., 2006) was not considered in their experiments. There is no mention of Ostrovsky's algorithm in their paper whatsoever. Surprisingly, there is no mention of Ostrovsky's algorithm in k-Means++ paper either even though it was published in 2007 whereas Ostrovsky's algorithm was published in 2006.

We would also like to point out that to our knowledge no surveys were done after removing linear dependency prior to running the experiments. This is a very important step if we are to get a meaningful clustering out of k-Means algorithm. PCA also helps with some drawbacks of k-Means such as k-Means struggles with non-spherical data set. PCA decomposes the existing data points into orthogonal[2] ones. If we use PCA (principal component analysis) before running a clustering algorithm, we can redefine the variables into linearly independent ones. For our experiment, we have used PCA on every data set before running cluster algorithm.

3. Proposed Initialization Methods

For a set of points S and a point x, we use $\min(\|x-S\|)$ to denote the minimum of distances from x to the points of S that is $\min(\|x-S\|) = \min_{a\in S}(\|x-a\|)$. Set $D(\mathbf{x}) = \min(\|\mathbf{x}-\mathcal{C}\|)$ for a point x and a set of centers $\mathcal{C}$. Let us denote the centroid of S by μ_S that is $\mu_S = \frac{1}{|S|}\sum_{x\in S} x$.

3.1 First center for k-Means++

In k-Means++ algorithm, the first center is chosen uniformly at random. However, not all points have the same contribution to inertia. We choose x as a first center in a way that is equivalent to the variance explained by x.

i Choose $\mathbf{x}$ with probability $\frac{\|\mathbf{x}-\mu_{\mathbf{X}}\|^2}{\sum_{\mathbf{x}\in\mathbf{X}}\|\mathbf{x}-\mu_{\mathbf{X}}\|^2}$. Set $\mathcal{C}_1 = \{\mathbf{x}\}$.

ii Repeat the remaining steps in k-Means++ (Arthur et al., 2007, Section 2.2, Page 3).

3.2 Centroid of Centers Based Seeding

We want to choose x with probability proportional to squared distance from centroid of the cluster centers. Our motivation for doing so is the following. In k-Means++, probability is proportional to squared minimum distance from the centers. Therefore, the larger this minimum squared distance is, the higher the probability is for x to be chosen as a center. So, in a sense this can be thought of maximizing the minimum squared distance from the centers to the point in consideration. We intend to check the case where we choose the probability proportional to the total sum of squared distances rather than just the minimum one. AS we will show later, sum of all squared distances from centers to the point in discussion is actually dependent on the distance from centroid of those cluster centers to that point.

2. It is well known that orthogonal vectors are linearly independent.

i Choose a point $\mathbf{x}$ as stated in step (i) of section (3.1). Set $\mathcal{C}_1 = \{\mathbf{x}\}$.

ii For an already existing set of i centers $\mathcal{C}_i = \{c_1, \cdots, c_i\}$, choose a new center $\mathbf{x} \in \mathbf{X}$ with probability proportional to $\|\mathbf{x} - \mu_{\mathcal{C}_i}\|^2$.

iii Repeat step (ii) until $i = k$.

iv For each $1 \leq i \leq k$, set $\mathcal{C}_i = \{\mathbf{x} \in \mathbf{X} : \|\mathbf{x} - c_i\| = \min(x - \mathcal{C})\}$.

v Set $c_i = \mu_{\mathcal{C}_i}$.

vi Repeat (iv) and (v) until convergence or number of iteration is reached.

4. Analysis

First, we will analyze the choosing of first center in k-Means++.

4.1 k-Means++ Improved

Consider a set of n points $\mathbf{X}$ and that the probability of $x \in \mathbf{X}$ being chosen as a center as $p(x)$. From the definition of variance, for a set of points S,

$$\sigma^2(S) = \frac{\sum_{x \in S} \|x - \mu_S\|^2}{|S|}$$

$$\sum_{x \in S} \|x - \mu_S\|^2 = |S|\sigma^2 \tag{1}$$

For any arbitrary point a and μ as the centroid of $\mathbf{X}$,

$$\begin{aligned}
\sum_{\mathbf{x} \in \mathbf{X}} \|\mathbf{x} - a\|^2 &= \sum_{\mathbf{x} \in \mathbf{X}} \|\mathbf{x} - \mu + \mu - a\|^2 \\
&= \sum_{\mathbf{x} \in \mathbf{X}} \left(\|\mathbf{x} - \mu\|^2 + 2\langle \mathbf{x} - \mu, \mu - a \rangle + \|\mu - a\|^2 \right) \\
&= \sum_{\mathbf{x} \in \mathbf{X}} \|\mathbf{x} - \mu\|^2 + 2 \left\langle \sum_{\mathbf{x} \in \mathbf{X}} \mathbf{x} - n\mu, \mu - a \right\rangle + n\|\mu - a\|^2 \\
&= n\sigma^2 + 2\langle n\mu - n\mu, \mu - a \rangle + n\|\mu - a\|^2 \\
&= n(\sigma^2 + \|\mu - a\|^2)
\end{aligned} \tag{2}$$

Here $\langle a, b \rangle$ is the dot product of vectors a and b. Using equation (2), we have the following.

$$\begin{aligned}
\sum_{\mathbf{x} \in \mathbf{X}} \sum_{\mathbf{y} \in \mathbf{X}} \|\mathbf{x} - \mathbf{y}\|^2 &= \sum_{\mathbf{x} \in \mathbf{X}} n(\sigma^2 + \|\mu - \mathbf{x}\|^2) \\
&= n(n\sigma^2 + \sum_{\mathbf{x} \in \mathbf{X}} \|\mu - \mathbf{x}\|^2) \\
&= n(n\sigma^2 + n\sigma^2) \\
&= 2n^2\sigma^2
\end{aligned} \tag{3}$$

Using equation (3), the probability becomes

$$\begin{aligned} p(x) &= \frac{\sum_{\mathbf{y}\in\mathbf{X}} \|\mathbf{x}-\mathbf{y}\|^2}{\sum_{\mathbf{y}\in\mathbf{X}}\sum_{\mathbf{x}'\in\mathbf{X}} \|\mathbf{x}'-\mathbf{y}\|^2} \\ &= \frac{n(\sigma^2 + \|\mu-\mathbf{x}\|^2)}{2n^2\sigma^2} \\ &= \frac{\sigma^2 + \|\mu-\mathbf{x}\|^2}{2n\sigma^2} \\ &= \frac{1}{2} + \frac{\|\mu-\mathbf{x}\|^2}{2n\sigma^2} \end{aligned}$$

However, to make things smoother, one can also choose to use the following as the probability of x being chosen as the first center.

$$p(x) = \frac{\|\mu-\mathbf{x}\|^2}{2n\sigma^2}$$

Notice that the denominator is the variance of S. Therefore, we can consider this as the amount of variance explained by x. Also, notice that computationally this version of $p(x)$ is much cheaper than using $\sum_{\mathbf{y}\in\mathbf{X}} \|\mathbf{x}-\mathbf{y}\|^2$. Therefore, we considered $p(x)$ proportional to $\|x-\mu\|^2$ for choosing x as the first center in our experiments.

4.2 Centroid of Centers Seeding

We will now analyze the center initialization algorithms using inertia value. Since these algorithms are probabilistic, we are going to take a look at the expected value of inertia. For a set of points $\mathbf{X}$ and a set of centers $\mathcal{C}$, we denote the inertia by $\mathcal{I}_{\mathcal{C}}(\mathbf{X})$. For the optimal set of cluster centers $\mathcal{C}_{opt}$, we denote the corresponding inertia by $\mathcal{I}_{opt}(\mathbf{X})$.

$$\mathcal{I}_{\mathcal{C}}(\mathbf{X}) = \sum_{\mathbf{x}\in\mathbf{X}} \min_{\mathbf{c}\in\mathcal{C}}(\|c-x\|^2)$$

If the context is clear, we may omit $\mathcal{C}$ and $\mathbf{X}$. For a fixed set of points $\mathbf{X}$, if the probability of $\mathbf{x}\in\mathbf{X}$ being chosen to be a center is $p(x)$ with respect to a set of centers $\mathcal{C}$, then the expected value of inertia $E[\mathcal{I}(\mathbf{X})]$ is

$$E[\mathcal{I}(\mathbf{X})] = \sum_{\mathbf{x}\in\mathbf{X}} p(x) \sum_{\mathbf{y}\in\mathbf{X}} \min(D(x), \|y-x\|)^2$$

The following lemma was proven in (Arthur et al., 2007, Lemma 3.2).

Lemma 1 *Let A be an arbitrary cluster in $\mathcal{C}_{opt}$, and let $\mathcal{C}$ be the clustering with just one center, which is chosen uniformly at random from A. Then, $E[\mathcal{I}(A)] = 2\mathcal{I}_{opt}(A)$.*

Here, $\mathcal{I}_{opt}(A) = \sum_{x\in A} \|x-\mu_A\|^2$.

Now, we want to find a similar formula for the remaining centers using centroid of centers seeding. If we take $p(x)$ proportional to $f(x) = \|x-\mu_{\mathcal{C}}\|^2$, then using equation (2),

$$\begin{aligned} f(x) &= \|x-\mu_{\mathcal{C}}\|^2 \\ \sum_{x\in S} f(x) &= \sum_{x\in S} \|x-\mu_{\mathcal{C}}\|^2 \\ &= |S|(\sigma^2 + \|\mu_S - \mu_{\mathcal{C}}\|^2) \end{aligned}$$

Then the probability of x being chosen as a center is

$$\begin{aligned} p(x) &= \frac{f(x)}{\sum_{y \in S} f(y)} \\ &= \frac{\|x - \mu_{\mathcal{C}}\|^2}{|S|(\sigma^2 + \|\mu_S - \mu_{\mathcal{C}}\|^2)} \end{aligned}$$

Since $D(y) \leq \|y - \mu_{\mathcal{C}}\|$, we have the following

$$\begin{aligned} \sum_{y \in S} \|y - \mu_{\mathcal{C}}\|^2 &\geq D(y)^2 \\ \frac{1}{\sum_{y \in S} \|y - \mu_{\mathcal{C}}\|^2} &\leq \frac{1}{\sum_{y \in S} D(y)^2} \\ \frac{\sum_{y \in S} D(y)^2}{\sum_{y \in S} \|y - \mu_{\mathcal{C}}\|^2} &\leq 1 \end{aligned}$$

Our expected value of inertia would be

$$\begin{aligned} E[\mathcal{I}(S)] &= \sum_{x \in S} p(x) \sum_{y \in S} \min(D(y), \|x - y\|)^2 \\ &= \sum_{x \in S} \frac{\|x - \mu_{\mathcal{C}}\|^2}{\sum_{y \in S} \|y - \mu_{\mathcal{C}}\|^2} \sum_{y \in S} \min(D(y), \|x - y\|)^2 \\ &\leq \sum_{x \in S} \frac{\|x - \mu_{\mathcal{C}}\|^2}{\sum_{y \in S} \|y - \mu_{\mathcal{C}}\|^2} \sum_{y \in S} D(y)^2 \\ &\leq \sum_{x \in S} \|x - \mu_{\mathcal{C}}\|^2 \end{aligned}$$

Using equation (2), we have

$$E[\mathcal{I}(S)] \leq n(\sigma^2 + \|\mu_S - \mu_{\mathcal{C}}\|^2)$$

Now, using equation (3), we have

$$\begin{aligned} \mathcal{I}_{opt}(S) &= \sum_{x \in S} \|x - \mu_S\|^2 \\ &= n\sigma^2 \end{aligned}$$

Thus, we have the following theorem.

Theorem 2 *Let S be an arbitrary cluster chosen from $\mathcal{C}_{opt}$ and $\mathcal{C}$ be an arbitrary clustering. If a center is chosen from S and added to $\mathcal{C}$ using centroid of centers initialization, then*

$$E[\mathcal{I}(S)] \leq \mathcal{I}_{opt}(S) + n\|\mu_S - \mu_{\mathcal{C}}\|^2$$

For a set of points S, consider the partition into k clusters $\{S_1, S_2, \cdots, S_k\}$ so that $S = S_1 \cup S_2 \cup \cdots \cup S_k$ and $S_i \cap S_j = \{\}$ for $i \neq j$. For brevity, we denote the center of S_i by μ_i instead of μ_{S_i} and the center of S by μ. Also, assume that $|S_i| = n_i$ for each $1 \leq i \leq k$ and that the set of centers of the clusters is $\mathcal{C} = \{\mu_1, \mu_2, \cdots, \mu_k\}$. We have $\sum_{x \in S_i} \|x - \mu_i\|^2 = n(\sigma_i^2 + \mu_i^2)$ where σ_i is the variance of the cluster S_i.

$$\begin{aligned}
\sum_{x \in S} \|x - \mu\|^2 &= \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu\|^2 \\
n\sigma^2 &= \sum_{i=1}^{k} n_i(\sigma_i^2 + \|\mu_i - \mu\|^2) \\
&= \sum_{i=1}^{k} n_i\sigma_i^2 + \sum_{c \in \mathcal{C}} \|c - \mu\|^2 \\
&= \sum_{i=1}^{k} n_i\sigma_i^2 + k(\sigma_{\mathcal{C}}^2 + \|\mu_{\mathcal{C}} - \mu\|^2) \qquad (4)
\end{aligned}$$

Therefore, $k\|\mu - \mu_{\mathcal{C}}\|^2 \leq n\sigma^2$ and we have

$$\begin{aligned}
E[\mathcal{I}(S)] &\leq \mathcal{I}_{opt}(S) + n \cdot \frac{n\sigma^2}{k} \\
&\leq \mathcal{I}_{opt}(S) + n\frac{\mathcal{I}_{opt}(S)}{k}
\end{aligned}$$

This gives us the following result.

Theorem 3 $E[\mathcal{I}(S)] \leq \mathcal{I}_{opt}\left(1 + \frac{n}{k}\right)$.

We believe this estimate is not tight at all and it can be improved by a huge margin. For example, we had ideas of using Abel's summation formula (Apostol, 1976, Theorem 4.2) and Taylor series to approximate the sum on the right side of equation (4) and get a tighter bound. However, we was happy with the experimental results and therefore, we decided to not spend any more time in finding a tighter bound. We may work on this in future, but right now we do not have any plans to improve this bound.

5. Experiment Setup

We ensure that all algorithms are run under the same conditions. All of them share the same environment and no special optimizations were made for any particular algorithm. Only CPU was used to determine the values we are interested in and no parallelism mechanism was in place for speeding up the process. This way, we can get an idea about the raw performances of the algorithms involved.

Python is used as the programming language to write necessary codes. Some common auxiliary packages such as *scipy, scikit-learn, numpy* etc are used to help with the code. The algorithms are simply different methods of the same class, so they share the same fitting and prediction function. Only the initialization differs for different algorithm. It

should be mentioned that even though some packages have native support for k-Means implementation, we did not use them to run the experiments. Not all algorithms we want to test are available in those packages. Therefore, in order to ensure same environment and optimizations for every algorithm, we wrote them all from scratch so that we could be sure they are tested under the same settings.

The data sets used for the experiment are some of the popular ones.

1. Boston housing data set
2. Wine quality testing data set
3. Mall customers data set
4. Airlines cluster data set
5. Iris flower data set
6. Cloud data set

Miligan et al. (1988) shows that using z-score to standardize the data is not favorable for clustering because it loses between-cluster variation. Therefore, we did not use any sort of standardization or normalization lest it should lose variance or become prone to bias. Instead, we have used PCA to remove linear dependency among variables.

While experimenting on such algorithms, it is of utmost importance to run the same experiment more than once under the same parameters and conditions. For example, assume that we want to compare k-Means++ and Forgy's algorithm (Forgy, 1965) for $k = 5$ clusters. We should run this experiment at least m times where $m > 1$ in order to eliminate bias and account for randomness. We run each experiment a total of 20 times and take the average and minimum values of inertia.

We would like to address another important issue regarding k-Means algorithms. Usually convergence speed is tested for k-Means by checking how many iteration it requires until the centers stop changing. However, as we will see, given enough number of iterations all initialization can eventually produce the minimum inertia value. This enabled us to run the experiments a fixed number of times for every initialization instead of checking the convergence. There is another big reason for doing so. Consider some convergence criterion. For example, when the sum of squared distances between the old and new centers is less than some tolerance value, we reach convergence. Using this kind of criterion does not necessarily optimize inertia value as much as possible. In our experiments, we have found multiple instances where convergence is reached in less than 10 iterations but the inertia value is far from the minimum possible value. The speed of convergence is our secondary goal and since that does not actually help us distinguish the initialization procedures, we decided to not use this at all. We have another reason for doing so. If a data set is normalized or standardized for some reason, then the convergence criterion does not work as well either. The reason is that it already has very small distances between old and new centers. It is no wonder convergence would be reached really fast regardless of what initialization is used. And this does not necessarily indicate that the optimum inertia value was achieved. There is another benefit of using this setting. We can understand how consistent an initialization is. As we will see in section (6), every method can reach the minimum inertia at least once.

Algorithm	Average $\mathcal{I}$	Minimum $\mathcal{I}$	Time
k-Means	2586435.45	1442170.41	2.68s
k-Means++	1638992.86	1442170.41	2.66s
ORSS	1620320.77	1442170.41	2.69s
CoC	1562697.24	1442170.41	2.66s

Table 1: Results on Boston housing data set, 5 clusters

Algorithm	Average $\mathcal{I}$	Minimum $\mathcal{I}$	Time
k-Means	1011171.27	916424.19	0.93s
ORSS	1020273.2	916424.19	0.94s
k-Means++	1011171.27	916424.19	0.94s
CoC	994075.55	916424.19	0.94s

Table 2: Results on Wine data set, 5 clusters

Therefore, the method with lower average performed more consistently without any doubt. We will have a similar implication for comparison between k-Means++ and our proposed improvement for k-Means++.

6. Results

We present the results on the initialization procedures mentioned before. We have let each algorithm run exactly the same number of times (300) which is more than enough for convergence. The time registered here is the time taken by each algorithm for those 300 iterations. As expected, default k-Means is usually the fastest algorithm. Also, as mentioned earlier, every method was repeated 20 times and average/minimum was considered for those iterations.

6.1 Comparison of Different Initialization

The comparison of inertia for different initialization is shown in tables 1, 2, 3. We have shown the results here for $k = 5$ clusters.

Algorithm	Average $\mathcal{I}$	Minimum $\mathcal{I}$	Time
k-Means	5788612160669.75	5724491382108.87	20.22s
ORSS	5883085483272.44	5724491382108.87	20.2s
k-Means++	5908999306255.42	5724491382108.87	20.33s
CoC	5782820760840.1	5724491382108.87	20.2s

Table 3: Results on Airlines data set, 5 clusters

Algorithm	Average $\mathcal{I}$	Minimum $\mathcal{I}$	Time
k-Means++	17693133.44	17414699.85	4.95s
k-Means++ Improved	17614961.42	17414699.85	4.93s

Table 4: Results on Cloud data set, 5 clusters

Algorithm	Average $\mathcal{I}$	Minimum $\mathcal{I}$	Time
k-Means++	89866.67	78385.08	0.99s
k-Means++ Improved	88468.3	78392.42	0.99s

Table 5: Results on Mall customer data set, 5 clusters

6.2 k-Means++ vs k-Means++ Improved

The comparison of k-Means++ against k-Means++ improved is shown in tables 4, 5, 6.

Notice that in all cases, we achieved lower minimum inertia for k-Means++ and yet k-Means++ improved version achieved lower inertia on average. Moreover, even though these are randomized algorithms, pretty much every time we ran the experiments, k-Means++ improved version performed better. Given the mathematical intuitive justification and experimental results, we are inclined to say that this version works better than original k-Means++ consistently.

References

D. Arthur and S. Vassilvitskii. `k-means++`: The Advantages of Careful Seeding. Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics Philadelphia, PA, USA. pp. 1027–1035.

E. Forgy. Cluster analysis of multivariate data: Efficiency versus interpretability of classification. Biometrics, $21, 768 - 780, 1965$.

Stuart P. Lloyd. Least squares quantization in pcm. IEEE Transactions on Information Theory, $28(2) : 129–136, 1982$.

M. E. Dyer. A simple heuristic for the p-center problem. Operations Research Letters, Volume 3, February 1985, pp. $285 - 288$.

G. Milligan, M. C. Coope. A Study of Standardization of Variables in Cluster Analysis, Journal of Classification 5(2) (1988) 181–204.

Algorithm	Average $\mathcal{I}$	Minimum $\mathcal{I}$	Time
k-Means++	1007402.92	916424.19	0.91s
k-Means++ Improved	995907.65	916424.19	0.91s

Table 6: Results on Wine data set, 5 clusters

R. Ostrovsky, Y. Rabani, Leonard J. Schulman, C. Swamy. The Effectiveness of Lloyd-Type Methods for the k-Means Problem. Proceedings of the 47th Annual Symposium on Foundations of Computer Science. 2006.

M. Emre Celebi, Hassan A. Kingravi, Patricio A. Vela. A Comparative Study of Efficient Initialization Methods for the K-Means Clustering Algorithm. Expert Systems with Applications, $40(1) : 200–210, 2013$.

Tom M. Apostol. Introduction to Analytic Number Theory. Springer, $10.1007/978 - 3 - 662 - 28579 - 4$, 1976.