A survey on k-means initialization methods

Huang Xiangyuan, Liu Siyuan, Wang Hao

April 2020

Abstract

Randomly selecting all k centers in one batch, k-means algorithm is destined to hit only local optimas and yield inconsistent results from run to run. In recent years efforts were made to create modified versions of k-means, focusing on selecting a better set of initial cluster centers, including k-means++ that samples centers iteratively in k rounds; and k-means$||$ which samples a few more centers in lesser rounds of iterations. In these variations, k-means family of algorithms benefited from randomized process in clustering performance, however, the very first center still remains being randomly selected. We propose a simple modification in selecting the very first center from dataset, with experiments on synthetic data and real data, we show the effectiveness comparisons in different randomization methods in clustering.

1 Introduction

Clustering is one of the most important problems in data mining. The objective of clustering is to partition a population of unlabeled data points in Euclidean space into several groups (called clusters), where points within the same clusters are more similar to each other than those in different clusters. Over a century, many algorithms have been proposed to address the problem. However, one simple and classic algorithm, k-means, remains the most popular clustering algorithm due to its simplicity and efficiency. K-means starts with a set of randomly chosen initial cluster centers (called centroid), and then repeatly assign each point to its nearest centroid, and finally update the centroid position based on new assignment. The assignment and update process, called Lloyd's iteration, is repeated until convergence.

Given its simplicity, k-means algorithm suffers from several drawbacks. One of them is the result sensitivity to centroid initialization. A bad initialization could lead to exponential running time in worst case, and low quality final assignments which are far away from global optimum. Hence, many researcher have dedicated their effort to improve the initialization procedure for better clustering results and faster convergence time.

In this report, we review two papers discussing the improvement of the initialization procedure. The first paper by Arthur and Vassilvitskii[1]'s paper proposed a method named k-means++, which applied a more careful seeding in the choosing the initial centroids. Their experiments showed such augmentations helped outperform traditional k-means in both speed and accuracy, and they also proved that k-means++ is $O(\log k)$-competitive with respect to global optimum theoretically. Based on their results, Bahmani et al. [2] introduced k-means$||$, which allowed for parallelization in selecting starting centroids, and only required a logarithmic number of passes of the whole dataset.

2 Preliminaries

Before digging deep into the summary of three papers, we first introduce the definition of the clustering problem and mathematical notions that will be used in this report.

Let $X = \{x_1, ..., x_n\}$ be a set of data points in the d-dimensional Euclidean space and let k be a positive integer specifying the number of clusters. Let $||x_i - x_j||$ denote the Euclidean distance between x_i and x_j. For a point x and a subset $Y \subseteq X$ of points, the distance is defined as $d(x, Y) = min_{y \in Y} ||x - y||$. For a subset $Y \subseteq X$ of points, the centroid is given by

$$C(Y) = \frac{1}{|Y|} \sum_{y \in Y} y \tag{1}$$

Let $C = \{c_1, ..., c_k\}$ be a set of points and let $Y \subseteq X$. We define the cost of Y with respect to C as

$$\phi_Y(C) = \sum_{y \in Y} d^2(y, C) = \sum_{y \in Y} \min_{i=1,\dots,k} ||y - c_i||^2 \tag{2}$$

The objective of k-means clustering is to find a subset C of k centroids that minimizes $\phi_X(C)$

$$C = \{c_1, ..., c_k\} = \arg\min_{C} \phi_X(C) \tag{3}$$

Let C^* denote the optimal k-means clustering and ϕ^* denote the cost of that. It is known that finding C^* is NP-hard[3]. And we call a set C of centroids to be an *α-approximation* to C^* if $\phi_X(C) \leq \alpha\phi^*$. Note that the centroids automatically define the clustering result of X, as the i-th cluster includes all $x_j \in X$ such that x_j is closer to c_i than any other centroids c_i'. In the following sections, we will use c_i as the centroid of the i-th cluster, and use C_i to denote the i-th cluster where $x_j \in C_i$ iff $c_i = \arg\min_{c_i \in C} d(x_j, C)$.

3 Main Content

The original k-means algorithm is simple as discussed in the introduction section.

Algorithm 1 k-means

```
1: uniformly sample $k$ points $c_1, ..., c_k$ from $X$, let $C = \{c_1, ..., c_k\}$
2: repeat
3:   for each $i \in \{1, ..., k\}$ do
4:     $C_i = \{x_j | c_i = \arg\min_{c_i \in C} d(x_j, C)\}$
5:   end for
6:   for each $i \in \{1, ..., k\}$ do
7:     $c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$
8:   end for
9: until convergence
```

As we focus on the improvement on the initialization procedure (step 1), the step 2 to 9 in k-means algorithm, a.k.a the Lloyd's process, will be skipped in the following algorithm flow.

3.1 k-means++

Since the clustering result of k-means is vulnerable to bad initialization of centroids, Arthur and Vassilvitskii proposed a randomized version of centroid initialization, called k-means++, where the centroids are picked sequentially and new centroids are far away from centroids already chosen. To be specific, the probability of sampling x_j as a new centroid is proportional to its squared distance from the closest centroid that are already picked, and this is referred as D^2 weighting.

Algorithm 2 k-means++ initialization

```
uniformly sample one points $c_1$ from $X$, let $C = \{c_1\}$
while $|C| < k$ do
  sample $x \in X$ with probability $\frac{d^2(x,C)}{\sum_{x \in X} d^2(x,C)}$
  $C = C \cup \{x\}$
end while
```

The paper further proved that if the centroids set C is constructed with k-means++, the expected cost $E[\phi_X(C)] \leq 8(\ln k + 2)\phi^*$. We provide a brief proof based on two lemma (proof skipped).

Lemma 1 Let A be an arbitrary cluster in C^*, and $C = \{p\}$ to be a point uniformly sampled from A. Then $E[\phi_A(C)] = 2\phi_A^*$
Lemma 2 Let A be an arbitrary cluster in C^*, and let C be arbitrary set of centroids. If we add one random point p chosen with D^2 weighting, from A to C, then $E[\phi_A(C \cup \{p\})] \leq 8\phi_A^*$

Note that Lemma 2 suggests that if we could choose centroids from each cluster in C^*, the overall cost is constant approximation of ϕ^*. The next step is bound the total cost to $O(\log k)$ in general case by induction.

Lemma 3 Let C be an arbitrary clustering. Choose $u > 0$ "uncovered" clusters from C^*, and let X_u denote the set of points in these clusters, and $X_c = X - X_u$. Suppose we add $t \leq u$ random centroids to C, chosen with D^2 weighting, let C' be the resulting clustering and $\phi_X(C')$ be the cost, H_t be the harmonic sum.

$$E[\phi_X(C')] \leq (\phi_{X_c}(C) + 8\phi_{X_u}^*) \cdot (1 + H_t) + \frac{u-t}{u} \cdot \phi_{X_u}(C) \tag{4}$$

We prove by induction, showing that if the result holds for $(t-1, u)$ and $(t-1, u-1)$, then it holds for (t, u). The base cases are $t = u = 1$ and $t = 0, u > 0$. If $t = 0$ and $u > 0$, the result follows from the fact that $1 + H_t = \frac{u-t}{u} = 1$. If $t = u = 1$, we choose a new center from the one uncovered cluster with probability exactly $\frac{\phi_{X_u}(C)}{\phi_X(C)}$. In this case, Lemma 2 guarantees that $E[\phi_X(C')] \leq \phi_{X_c}(C) + 8\phi_{X_u}^*$. Since $\phi_{C'}(X) \leq \phi_C(X)$ even if we choose a centroid from a covered cluster, we have

$$E[\phi_X(C')] \leq \frac{\phi_{X_u}(C)}{\phi_X(C)} \cdot (\phi_{X_c}(C) + 8\phi_{X_u}^*) + \frac{\phi_{X_c}(C)}{\phi_X(C)} \cdot \phi_X(C) \leq 2\phi_{X_c}(C) + 8\phi_{X_u}^* \tag{5}$$

Since $1 + H_t = 2$ here, we have shown the lemma holds for both base cases. For the inductive step, we consider two cases.

Firstly, suppose our first centroid comes from a covered cluster, which happens with probability $\frac{\phi_{X_c}(C)}{\phi_X(C)}$. Note that the new centroid can only decrease ϕ. Applying the inductive hypothesis with the same choice of covered cluster, but with t decreased by one, we have

$$E[\phi_X(C')] \leq \frac{\phi_{X_c}(C)}{\phi_X(C)} \cdot \Big((\phi_{X_c}(C) + 8\phi^*_{X_u}) \cdot (1 + H_t) + \frac{u-t+1}{u} \cdot \phi_{X_u}(C) \Big) \tag{6}$$

On the other hand, if the new centroid comes from some uncovered cluster A, which happens with probability $\frac{\phi_A(C)}{\phi_X(C)}$. Let p_a denote the probability that we choose $a \in A$ as the centroid, and let $\phi_A(a)$ denote the cost. Applying the inductive hypothesis, after adding A to the covered clusters as well as decreasing both t and u by 1, then we have,

$$\begin{aligned} E[\phi_X(C')] &\leq \frac{\phi_A(C)}{\phi_X(C)} \cdot \sum_{a \in A} p_a \Big((\phi_{X_c}(C) + \phi_A(a) + 8\phi^*_{X_u} - 8\phi^*_A) \cdot (1 + H_{t-1}) + \frac{u-t}{u-1} \cdot (\phi_{X_u}(C) - \phi_A(a)) \Big) \\ &\leq \frac{\phi_A(C)}{\phi_X(C)} \cdot \Big((\phi_{X_c}(C) + 8\phi^*_{X_u}) \cdot (1 + H_{t-1}) + \frac{u-t}{u-1} (\phi_{X_u}(C) - \phi_A(a)) \Big) \end{aligned} \tag{7}$$

The last inequality comes from the fact that $\sum_{a \in A} p_a \phi_A(a) \leq 8\phi^*_A$, which is implied by Lemma 2. Note that the power mean inequality states that $\sum_{A \subset X_u} \phi_A(a)^2 \geq \frac{1}{u} \cdot \phi_{X_u}(C)^2$. Therefore, if we sum over all uncovered clusters A, we could bound the cost to be

$$\begin{aligned} &\frac{\phi_{X_u}(C)}{\phi_X(C)} \cdot \Big(\phi_{X_c}(C) + 8\phi^*_{X_u} \Big) \cdot (1 + H_{t-1}) + \frac{1}{\phi_X(C)} \cdot \frac{u-t}{u-1} \cdot \Big(\phi_{X_u}(C)^2 - \frac{1}{u} \cdot \phi_{X_u}(C)^2 \Big) \\ &= \frac{\phi_{X_u}(C)}{\phi_X(C)} \cdot \bigg(\Big(\phi_{X_c}(C) + 8\phi^*_{X_u} \Big) \cdot (1 + H_{t-1}) + \frac{u-t}{u} \cdot \phi_{X_u}(C) \bigg) \end{aligned} \tag{8}$$

Combining two cases, we have

$$\begin{aligned} E[\phi_X(C')] &\leq \Big(\phi_{X_c}(C) + 8\phi^*_{X_u} \Big) \cdot (1 + H_{t-1}) + \frac{u-t}{u} \cdot \phi_{X_u}(C) + \frac{\phi_{X_c}(C)}{\phi_X(C)} \cdot \frac{\phi_{X_u}(C)}{u} \\ &\leq \Big(\phi_{X_c}(C) + 8\phi^*_{X_u} \Big) \cdot (1 + H_{t-1} + \frac{1}{u}) + \frac{u-t}{u} \cdot \phi_{X_u}(C) \end{aligned} \tag{9}$$

The inductive step follows from the fact that $\frac{1}{u} \leq \frac{1}{t}$
Now consider the clustering C after we have completed Step 1. Let A denote the C^* cluster where we choose the first centroid. Applying Lemma 3 with $t = u = k - 1$, with A being the only covered cluster, we have

$$E[\phi_X(C')] \leq \Big(\phi_A(a) + 8\phi^* - 8\phi^*(A) \Big) \cdot (1 + H_{k-1}) \leq 8(\ln k + 2)\phi^* \tag{10}$$

The last step follows from Lemma 1 and the fact that $H_{k-1} \leq 1 + \ln k$

3.2 k-means||

Even though k-means++ improves the performance of k-means clustering by initializing "sparse" centroids, the sequential sampling process requires $O(k)$ pass scan of the full data and thus prohibits parallelization. Hence, it becomes a great concern when dealing with large volume of data, especially when it is costly or infeasible to hold all data in one single machine. To tackle the challenge, in stead of sampling one point in each pass, Bahmani et al. modified the initialization to sample $O(k)$ points in each round and repeated the the process for approximately $O(\log n)$ rounds. Then they reclustered these $O(k \log n)$ points into k centroids as the initial center for Lloyd's process. The paper named the algorithm as k-means|| and showed that their method allowed parallelization of the sample process and faster convergence.

The random initialization of k-means++ and the uniform initialization of k-means are like two ends of a spectrum. k-means selects k centroids uniformly at one single iteration while k-means++ uses k iteration to choose the centroids and each one is picked according to a non-uniform distribution. K-means|| takes advantage of both, using a small number of iteration and select more than one points in each iteration non-uniformly. An oversampling factor $l = \Theta(k)$ is used to control the expected number of points sampled in each iteration. Lastly, a weighted clustering method is adopt to generate the k centroids.

Algorithm 3 k-means|| initialization

```
1: uniformly sample one points $c_1$ from $X$, let $C = \{c_1\}$
2: $\phi = \phi_X(C)$
3: for $O(\log \phi)$ times do
4:     $C' \leftarrow$ sample each point $x \in X$ independently with probability $\frac{l \cdot d^2(x,C)}{\sum_{x \in X} d^2(x,C)}$
5:     $C = C \cup C'$
6: end for
7: for $x \in C, \omega_x = |\{x_j \in X | d^2(x_j, x) < d^2(x, C)\}|$
8: recluster the weighted points in $C$ into $k$ clusters
```

The paper also provided a proof that k-means|| can obtain a solution that is $O(\alpha)$-approximation to C^* if an α-approximation algorithm is used in Step 8. The skeleton of the proof is summarized below.

Consider a cluster A in the optimal k-means clustering, denote $|A| = T$ and sort the points in an increasing order according to their distance to the centroid $C(A)$. Let the ordering be $a_1, ..., a_T$. Let q_t be the probability that a_t is the first point in the ordering chosen by k-means|| and let q_{T+1} be the probability that no point is sampled from cluster A. Furthermore, let p_t denote the probability of selecting a_t, by definition, $p_t = l \cdot d^2(a_t, C)/\phi_X(C)$. Since each point is chosen independently, for any $1 \leq t \leq T, q_t = p_t \prod_{j=1}^{t-1}(1 - p_j)$ and $q_{T+1} = 1 - \sum_{j=1}^{T} q_j$.

If a_t is the first point in A sampled as a new center, we can assign all the points in A to a_t, or just stick with the current clustering of A. Let $s_t = min\{\phi_A, \sum_{a \in A} d^2(a_t, a)\}$, we have

$$E[\phi_A(C \cup C')] \leq \sum_{t=1}^{T} q_t s_t + q_{T+1}\phi_A(C) \tag{11}$$

For simplicity, here we adopt an assumption that all $p_t(1 \leq t \leq T)$ are the same and equal to some value p, but note the conclusion still holds though requires more work in the proof if p_t differs with each other. If all p_t equal, $q_t = p(1-p)^{t-1}$. Hence the sequence $\{q_t\}_{1 \leq t \leq T}$ is a monotonic decreasing sequence. Further let $s'_t = \sum_{a \in A} d^2(a, a_t)$, by the ordering of a_t, $\{s'_t\}_{1 \leq t \leq T}$ is a monotonic increasing sequence. Therefore,

$$\sum_{t=1}^{T} q_t s_t \leq \sum_{t=1}^{T} q_t s'_t \leq \frac{1}{T}(\sum_{t=1}^{T} q_t \cdot \sum_{t=1}^{T} s'_t) \tag{12}$$

The last inequality follows from applying Chebyshev's sum inequality on the two monotonic sequence $\{q_t\}_{1 \leq t \leq T}$ and $\{s'_t\}_{1 \leq t \leq T}$. Following Lemma 1, $\frac{1}{T}\sum_{t=1}^{T} s'_t = 2\phi^*_A$. Hence,

$$E[\phi_A(C \cup C')] \leq (1 - q_{T+1})2\phi^*_A + q_{T+1}\phi_A(C) \tag{13}$$

The above inequality shows that a fraction of ϕ_A is replaced with a constant factor of ϕ^*_A for each optimal cluster A, and in each iteration of k-means|| initialization. Thus, step 1-6 can obtain a constant approximation to C^*. If an α-approximation algorithm is used in Step 8, k-means|| can obtain a $O(\alpha)$-approximation to C^*.

4 Open Problems

We have identified that, in both k-means++ and k-means||, the very first point chosen as center is still from uniform random distribution. Imagining a point cloud where certain regions are more densely distributed than other regions, by intuition the probability of forming a better clustering is higher if we choose the first point from this denser region. Similar ideas were implemented by Karteeka Pavan[4], who proposed a modified algorithm called SPSS. The first center is chosen to be close to more number of other points in the data set. Different from k-means++ in which each run would still yield different results, since the first center is selected with a specific criteria, SPSS yields unique solutions on each dataset, The first chosen point is the highest density point in the dataset and this solution is claimed to be insensitive to outliers in initialization step due to its first point is selected in a dense region in the point cloud.

Inspired by the work of Chris Ding[5]: K-means Clustering via Principal Component Analysis, we have the idea of preprocessing the data using PCA and try to get a better process in selecting the first center, on top of the original k-means++ algorithm.

We first propose a modified algorithm based on k-means++, focusing on the first point initialization, then we generalize it to a modified version on top of k-means||. Suppose we have a high dimensional gaussian ball dataset, we first use PCA to reduce the dataset to only 1 dimension, then we choose the point from original dataset whose projected center is the median of the the whole set. This method is supported by the intuition that projected dimension reflect the most variance explained, by segmenting along the single axis, we can tell apart some cluster information.

Algorithm 4 Modified k-means++ initialization

```
project original scaled dataset of $N$ dimensions down to 2D subspace
randomly generate $m$ windows of predefined size (a constant factor of 2D canvas size)
for each 2D point, compute the frequency of inclusion by $m$ windows, take the point with the highest frequency and find its corresponding $c_1$ from $X$, let $C = \{c_1\}$
while $|C| < k$ do
    sample $x \in X$ with probability $\frac{d^2(x,C)}{\sum_{x \in X} d^2(x,C)}$
    $C = C \cup \{x\}$
end while
```

We have summarized the proposed method in modified k-means++ above, when having projected the dataset onto low dimensional subspace, we can use month carlo sampling to find the points in densely distributed regions. As a 2D example below, two random windows were generated, with their spawn locations sampled from a uniform distribution along 2 axis. We fix the window size to be a constant factor of the total canvas size to adapt to each dataset. After each window is generated, each encircled point will receive 1 vote. After repeating m times we compute the accumulated votes for all points and choose the point with the highest vote as the first centroid. Since it is computationally expensive for pure euclidean based method to find the point in the densest region, we designed this sampling method to relieve the computation cost.

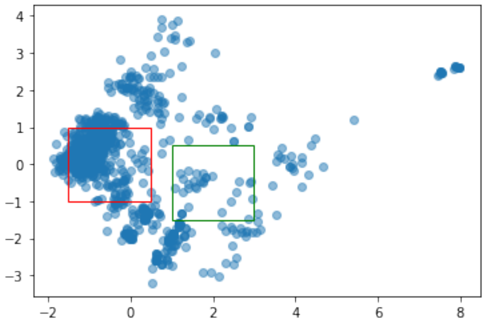

Figure 1: Monte Carlo computation of dense points

We have also attached out experimental results using this method in the following section. But now we wish to generalize this idea to k-means$||$. Considering the process of doing monte carlo sampling in the above process, apparently it is convertible to a parallel computation frame since each window-voting is independent to each other. Therefore we only need to spawn up a few windows together and we can finish computing the votes per point in fewer rounds. Due to time constraint we only implemented the modified version of k-means++ to a simplified context.

5 Experimental Results

5.1 Datasets

We used 3 types of datasets in evaluating the clustering algorithms: the first set is synthetic data from 15 dimensions of gaussian distributions, each attribute with a random standard deviation picked in a given range. This first dataset is similar to commonly used synthetic clustering datasets like norm25 and since it corresponds well with theoretical assumption from kmeans, this will be the key dataset used in evaluation. The second dataset is a synthetic set by sampling 10 attributes from gaussian distribution and 4 features from predefined gamma distributions, with the purpose to test the applicability of kmeans family algorithms on datasets with non-gaussian distribution. The third dataset is UCI cloud data with 10 attributes to perform clustering, with the purpose of apply clustering algorithms onto real use cases and compare performances between algorithms.

5.2 Results

In this section, we will show our experimental results on the 3 datasets aforementioned. Since the performance difference between k-means and k-means++, k-means$||$ has been theoretically and practically proved (we also provide these experimental results in the appendix), we focus our analysis on our proposed modification in choosing the first centroid and compare k-means++ and our modified algorithm of k-means++, the results are shown below.

From the results we have two immediate observations: First thing to note is that the 3 datasets used in our experiment have different level of clusterability, the family of k-means algorithms are designed to be optimally

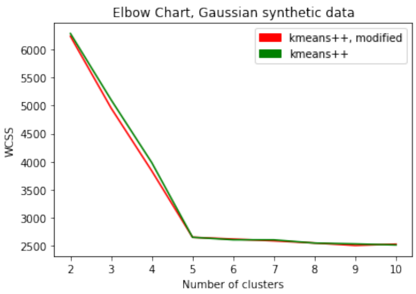

Figure 2: Gaussian data

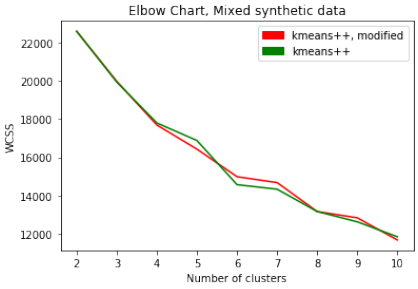

Figure 3: Gaussian with Gamma data

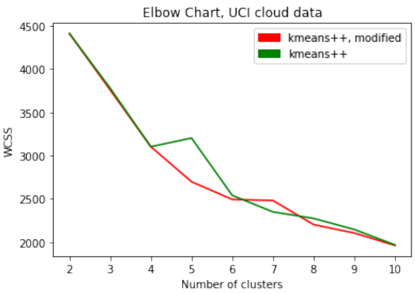

Figure 4: UCI cloud data

operated on gaussian distributed dataset and is validated in Figure 2, where our input synthetic data contains 5 clusters. For the mixed dataset where some attributes are sampled from gamma distribution, the family of k-means algorithm does not give an obvious clustering. We also observe that our proposed modification on the choice of first centroid will have a slight improvement in WCSS (within-cluster sum of squares) performance. We applied 1D subspace manipulation on the projected dataset to get the first centroid instead of the 2D example in last section's description.

We also did a basic test to compare k-means and k-means++ to show the performance difference, the results are in the appendix. In order to better view the result of k-means++ and kmeans on dataset, principle component analysis(PCA) is implemented on the original datasets and the cluster centroids (Figure 5-10). Each algorithm runs 20 times with pre-determined random states. It is shown that after 10 iterations, kmeans++ nearly converges on both a real-life dataset and a gaussian distributed dataset, but k-means takes more than 100 iterations to converge.

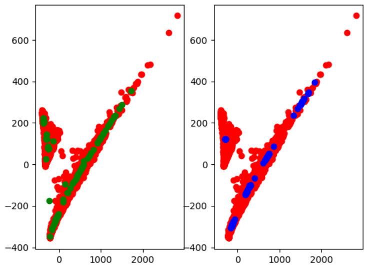

Figure 5: UCI data - Centroids initialized

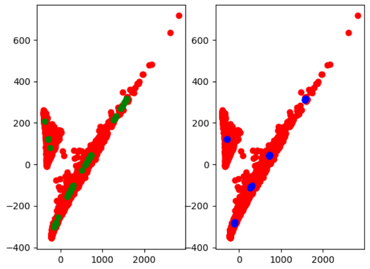

Figure 6: UCI data - Centroids after 10 iterations

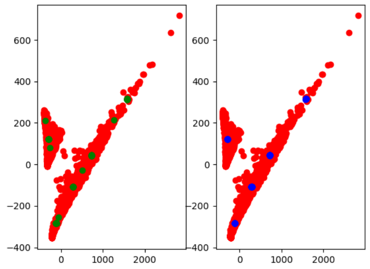

Figure 7: UCI data - Centroids after 100 iterations

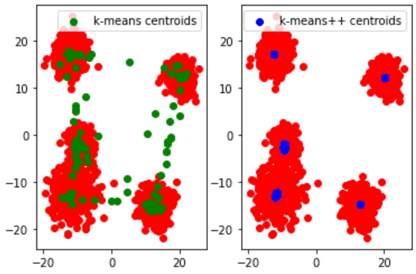

Figure 8: Gaussian data - Centroids initialized

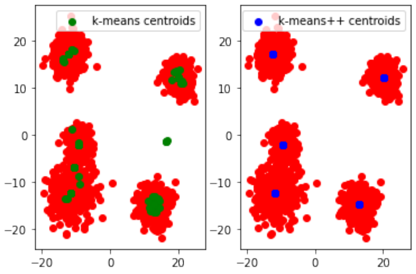

Figure 9: Gaussian data - Centroids after 10 iterations

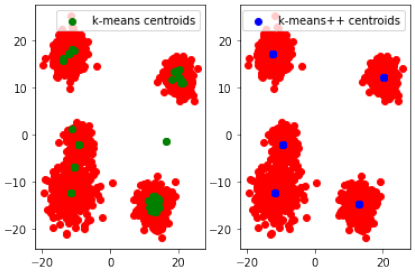

Figure 10: Gaussian data - Centroids after 100 iterations

6 Conclusion

In this short review, we have investigated theoretically and practically on the variants of k-means algorithms. We see the benefits of having a careful selection on the initial set of centroids and proposed an add on step to choose a better first centroid. We have also practically proved its slight improvement over k-means++ algorithm in the experiments on 3 different datasets.

References

[1] D. Arthur, S. Vassilvitskii, k-means++: The advantages of careful seeding, Stanford (2006).

[2] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, S. Vassilvitskii, Scalable k-means++, arXiv preprint arXiv:1203.6402. (2012).

[3] D. Aloise, A. Deshpande, P. Hansen, P. Popat, Np-hardness of euclidean sum-of-squares clustering, Machine learning 75.2 (2009): 245-248. (2009).

[4] K. Pavan, A. A. Rao, A. D. Rao, G. Sridhar, Robust seed selection algorithm for k-means type algorithms, arXiv preprint arXiv:1202.1585 (2012).

[5] C. Ding, X. He, K-means clustering via principal component analysis, Proceedings of the twenty-first international conference on Machine learning(p. 29) (2004).

Appendix

WCSS comparison between k-means and k-means++ on 3 datasets.

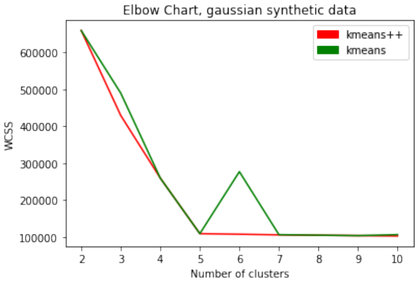

Figure 11: Gaussian data

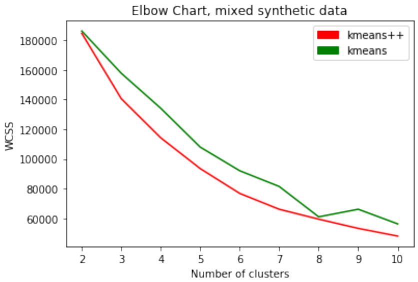

Figure 12: Mixed data

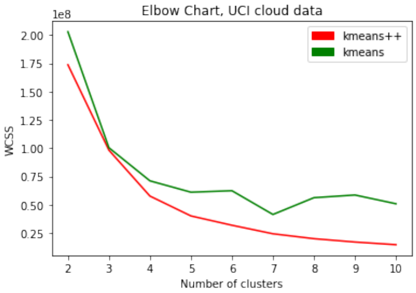

Figure 13: UCI cloud data