An Exploration of the paper "Power k-Means Clustering" by Jason Xu, Kenneth Lange

Kathryn Carbone, Erik Swanke, Cindy Lyu

April 2023

1 Motivation

1.1 An introduction to clustering

K-Means clustering broadly is a method to separate data into categories of similarity. Formally it is defined as partitioning n data points into k clusters, where each data point belongs to the closest cluster, such that the total variance (squared distance between x_i and the mean of all data inside a cluster) in every cluster is minimized. The general case, where the number of dimensions d and clusters k are uncapped, is NP-complete. Because of this, there exist approximation algorithms that will achieve a local minimum in a relatively fast time complexity, such as Lloyd's algorithm. Notably, however, the K-Means objective is a nonconvex optimization problem and thus makes finding a global minimum difficult. Let $\mathbf{X}$ denote the dataset. C_j denotes a cluster, defined as the set of all datapoints x which belong to this cluster. Finally, define Θ as the center matrix, or the matrix containing all central points for which clustering is performed. In other words, for each cluster C_j in the range of 1 to k, minimize the Euclidean distance of each datapoint x_i from the center of C_j. The k-Means objective function is then defined as

$$f_{-\infty}(\Theta) = \sum_{j=1}^{k} \sum_{i \in C_j} ||x_i - \theta_j||^2$$

This can then be re-expressed as

$$f_{-\infty}(\Theta) = \sum_{i=1}^{N} min_{1 \leq j \leq k} ||x_i - \theta_j||^2$$

1.2 Lloyd's algorithm and center-based clustering

Vanilla K-Means clustering famously makes use of Lloyd's algorithm, a simple method for grouping datapoints based on their distance from a pre-selected cluster center. This algorithm is one of several center-based clustering algorithms, which, true to their name, perform clustering around central points within the dataset. Lloyd's algorithm makes use of a two-step update process to converge at a local minimum, making it popular for its simplicity. This gives us a unique partitioning of datapoints into clusters (without repetition), but is highly susceptible to performance issues in higher dimensional data, and given poor selection of initial centroids. This is because Lloyd's algorithm employs a greedy update scheme, making it difficult to overcome local minima within the problem

Algorithm 1 Lloyd's Algorithm

Initialize Θ with k random datapoints
$X \leftarrow x$
$\Theta \leftarrow \theta$ Randomly initialize centroids
while Not Converged **do**
  Empty each C_j
  for $x_i \in X$ **do**
    $j = argmin_j\{||x_i - \theta_j||\}$
    $C_j \leftarrow C_j \cup x_i$ ▷ Reassign each datapoint to its nearest cluster
  end for
  for $j = 1 \cdots k$ **do**
    $\theta_j = \frac{1}{|C_j|}\sum_{x_i \in C_j} x_i$ ▷ Recalculate each centroid as the average of points in its cluster
  end for
end while

space to reach a global minimum. Thus, for a poor selection of initial centroids, one may be able to achieve a local optimum after running Lloyd's algorithm, but this could be significantly less optimal than other solutions with different initial clusterings.

1.3 Generalized and power means

Before discussing the Power K-Means algorithm, the concept of generalized and power means must first be discussed. The generalized means formula is simply a method of describing many different types of means in a single formula, rather than separate, more specific formulas for each. Given a positive integer k and a function $g(y)$ which is continuous and strictly decreasing or increasing, the generalized means formula is defined as such:

$$M_g(\mathbf{y}) = g^{-1}[\frac{g(y_1) + \cdots + g(y_k)}{k}]$$

In words, the generalized mean of a vector $\mathbf{y}$ is the inverse of the function g applied to the mean of $g(\mathbf{y})$.

Suppose we let $g(y) = y^s$ where $s \in \mathbb{R}$ and $\mathbf{y} \in (0, \infty)$. Then we restrict the generalized mean to the subset of power means, also known as Hölder means. This is expressed as

$$M_s(\mathbf{y}) = [\frac{y_1^s + \cdots + y_k^s}{k}]^{\frac{1}{s}}$$

Consider the gradient for the family of power means, $\frac{\partial}{\partial y_j} M_s(\mathbf{y})$:

$$\frac{\partial}{\partial y_j} M_s(\mathbf{y}) = \frac{1}{k}(\sum_{i=1}^{k} y_i^s)^{\frac{1}{s}-1}\frac{1}{k}y_j^{s-1} > 0$$

Here we can easily see that that the derivative is always greater than 0 and thus the Power Means family consists of strictly increasing functions.

Two interesting facts to note are that $s > 1$ yields the family of l_s-norms acting on vector $\mathbf{y}$, $s = 1$ results in the arithmetic mean, and $s = -1$ results in the harmonic mean. Indeed, the harmonic mean has previously been investigated as an improvement to Lloyd's algorithm.

1.4 K-Harmonic Means Algorithm

K-Harmonic Means (KHM) is a center-based, iterative algorithm that refines the clusters defined by K centers. It takes the sum over all data points of the harmonic average of the squared distance from a data point to all the centers as its objective function:

$$f_{-1}(\Theta) = \sum_{i=1}^{N} \frac{k}{\sum_{j=1}^{k}(||x_i - \theta_j||^2)^{-1}}$$

Previous work has shown that the harmonic average (denoted HA) can be used as a proxy for the minimizer function ([5]), motivating the adjustment of the original objective function

$$f_{-\infty}(\Theta) = \sum_{i=1}^{N} min_{1 \leq j \leq k} ||x_i - \theta_j||^2$$

to

$$f_{-1}(\Theta) = \sum_{i=1}^{N} HA_{1 \leq j \leq k} ||x_i - \theta_j||^2 = f_{-1}(\Theta) \quad = \quad \sum_{i=1}^{N} \frac{k}{\sum_{j=1}^{k}(||x_i - \theta_j||^2)^{-1}}$$

Using the harmonic mean as a proxy function improves performance on clustering by smoothing the geometry of the problem space. This is significant in that a smoother problem space reduces local minima, making clustering less susceptible to poor selection of initial centroids. However, previous empirical studies have also shown that the k-Harmonic mean algorithm is most beneficial in low dimensions, making it less robust than desired [4]. Details of the harmonic mean algorithm have been excluded to minimize redundancy in the report and presentation as it is subsumed by the Power k-Means algorithm.

2 Concepts

2.1 Power K-Means objective function

The Power K-means algorithm has the objective function:

$$f_s(\Theta) = \sum_{i=1}^{n} M_s(||x_i - \theta_1||^2, \cdots, ||x_i - \theta_k||^2)$$

which can be expanded to

$$f_s(\Theta) = [\frac{(||x_i - \theta_1||^2)^s + \cdots + (||x_i - \theta_k||^2)^s}{k}]^{-s}$$

as previously mentioned, $f_{-1}(\Theta)$ is KHM objective. $f_{-\infty}(\Theta)$ is the standard k-means because of the end behavior $\lim_{s \to -\infty} M_s(y) = \min\{y_1, \cdots, y_k\}$. Finding the hessian of $f_s(\Theta)$, Minimizing this objective function, and deriving the weight update function in Algorithm 2 will be part of the problem set.

2.2 Majorization minimization

Majorization-minimization (MM) is a general optimization framework for finding the minimum of a non-convex or non-smooth objective function. The principle of MM is to iteratively minimize a sequence of surrogate functions that majorize the original objective function at each step until convergence to a stationary point.

Given a non-convex objective function $f(\theta)$, the MM principle seeks to find a sequence of surrogate functions $g(\theta|\theta_m)$ at the m-th step that satisfy the following two properties:

$$\begin{aligned} f(\theta_m) &= g(\theta_m \mid \theta_m) && \text{tangency at } \theta_m \\ f(\theta) &\leq g(\theta \mid \theta_m) && \text{domination for all } \theta \end{aligned}$$

The constructed function $g(\theta|\theta_m)$ is the majorized version of $f(\theta)$ with respect to θ at θ_m. Then, update θ_{m+1} by minimizing $g(\theta|\theta_m)$ instead of $f(\theta)$:

$$\theta_{m+1} = \arg\min_{\theta} g(\theta \mid \theta_m),$$

which implies the decrease of $f(\theta)$:

$$f(\theta_{m+1}) \leq g(\theta_{m+1}|\theta_m) \leq g(\theta_m|\theta_m) = f(\theta).$$

Note that all Expectation-Maximization (EM) algorithms for maximum likelihood estimation are instances of MM. Since Lloyd's algorithm can be considered EM for Gaussian mixtures with limiting $\sigma^2 \to 0$, it can be improved by the broader MM principle which leads to the Power k-Means Algorithm.

2.3 The Power K-Means algorithm

The Power k-Means Clustering is derived from Lloyd's Algorithm and MM principle. It can be easily proved that $M_s(\mathbf{y})$ is concave whenever $s \leq 1$ and convex otherwise. Since M_s is concave when $s < 1$, it lies under its tangent at y_m where y_m is the estimate of f at the m-th iteration. Thus, we find the linear majorization:

$$M_s(\mathbf{y}) \leq M_s(\mathbf{y}) + \sum_{j=1}^{K} \nabla_{y_j} M_s(y_m)^T (y_j - y_{m,j}) \quad (*).$$

Substituting $||x_i - \theta_j||^2$ for y_j and $||x_i - \theta_{m,j}||^2$ for $y_{m,j}$ and summing over all data points, the inequality $(*)$ now becomes

$$f_s(\Theta) \leq f_s(\Theta_m) + \sum_{i=1}^{n} \sum_{j=1}^{K} \nabla_{y_j} M_s(y_m)^T ||x_i - \theta_j||^2 - \nabla_{y_j} M_s(y_m)^T ||x_i - \theta_{m,j}||^2$$

The right-hand side of the above inequality then serves as a surrogate function majorizing the original ojective function $f_s(\Theta)$. Since the first and the last term does not depend on Θ, only the second term needs to be considered to minimize the surrogate function. Denote the coefficient of the selected term as weights, solve for the gradient of M_s to get:

$$\omega_{m,ij} = \frac{\frac{1}{k}||x_i - \theta_{m,l}||^{2(s-1)}}{\left(\frac{1}{k}\sum_{l=1}^{k} ||x_i - \theta_{m,l}||^{2s}\right)^{(1-\frac{1}{s})}}$$

Thus, the updating rule is constructed to minimize the weights:

$$\theta_{m+1,j} = \frac{\sum_{i=1}^{n} \omega_{m,ij} x_i}{\sum_{i=1}^{n} \omega_{m,ij}}$$

The Power k-Means Clustering successively minimizes the majorization given by weights instead of the Euclidean distance between data points and centers and updates the centroids accordingly as shown in Algorithm 2.

Algorithm 2 Power k-Means Algorithm

Input: Data matrix $X \in \mathbb{R}^{n \times d}$, number of clusters k

1: Initialize $s_0 < 0$, Θ_0, $\eta > 1$ ▷ initial exponent s_0, exponent increment factor η

2: **repeat**

3: $\omega_{m,ij} \leftarrow \left(\sum_{l=1}^{k} ||x_i - \theta_{m,l}||^{2s_m} \right)^{\frac{1}{s_m}-1} ||x_i - \theta_{m,l}||^{2(s_m-1)}$

4: $\theta_{m+1,j} \leftarrow \left(\sum_{i=1}^{n} \omega_{m,ij} \right)^{-1} \sum_{i=1}^{n} \omega_{m,ij} x_i$

5: $s_{m+1} \leftarrow \eta \cdot s_m$ ▷ Optional Annealing

6: **until** convergence

By analogy to the Lloyd's algorithm, this new algorithm updates the weights $\omega_{m,ij}$ (step 3) and the centers $\theta_{m+1,j}$ (step 4) at each iteration with time complexity $O(nkd)$, which retains the efficiency of the original Lloyd's algorithm. Moreover, the additional power parameter s takes the advantage of annealing and makes the Power k-Means Clusering more robust to initialization than the previous Lloyd's Algorithm and KHM. (See proof in section 2.4.1)

2.4 Performance guarantees

Motivation for Power K-means lies in some of the mathematical properties derived from its objective function.

2.4.1 Annealing

Refer to the optional annealing step with respect to s in Algorithm 2.

Geometrically, as s tends to $-\infty$, the power means surfaces $f_s(\Theta)$ provide a family of progressively rougher landscapes, which means more and more local valleys appear on the objective surfaces (bad initialization in Lloyd' Algorithm). In the other extreme when $s = 1$, all optimal centers θ_j collapse to the sample mean $\overline{X}$.

Proposition 2.1 For any decreasing sequence $s_m \leq 1$, the iterates Θ_m produced by the MM updates generates a decreasing sequence of objective values $f_{s_m}(\Theta_m)$ bounded below by 0.

Proof. By the power mean inequality, we have $M_{s_{m+1}}(y) \leq M_{s_m}(y)$ for $s_{m+1} \leq s_m$ where $\lim_{s\to\infty} M_s(y) = \min(y1, ..., y_k)$. By summing over all data points, the descent properties of MM principle holds for chosen surrogate function $g = \omega : 0 \leq f_{s_{m+1}}(\Theta_{m+1} \leq g(\Theta_m) = f_{s_m}(\Theta_m)$. As a consequence, the sequence of objective values converges.

In words, altering the objective function at each MM step in our power k-means algorithm still guarantees the descent property and the advantages that follow from it. This also means we can freely choose a schedule for decreasing s in the initialization of the power k-means algorithm.

With s equals to -1, the algorithm serves as k-harmonic means clustering which attempts to optimize one member $f_{-1}(\Theta)$ of an entire family of objectives. Thus, KHM is more robust to initialization than Lloyd's algorithm by applying the heuristic annealing methods. While moving along a sequence of power mean objectives $f_s(\Theta)$, the power k-means algorithm automatically provides a form of annealing that guides the solution path toward a global minimizer as it threads its way across the landscapes. Therefore, the Power K-Means retains the benefits of annealing as dimension d increases, remaining successful in settings where both KHM and Lloyd's algorithm deteriorate.

3 Experimental Results

Our experimentation focused on comparing the performance of K-Means, K-Harmonic Means, and Power K-Means on real and synthetic datasets. Areas of analysis included performance comparisons based on dimensionality, dataset size, and power hyperparameter initialization. Performance was measured using the VI (variation of information), NMI (normalized mutual information), and ARI (adjusted Rand index) metrics.

Synthetic datasets are generated using a binomial distribution with true centers at coordinates $(10 * d), (20 * d), (40 * d)$ to account for variance in feature count. Each cluster contains the same amount of data points to better compare performance across all clustering methods. Synthetic dataset generation and Power K-Means algorithm implementation makes use of code from [2], an extension of the paper [3] analyzed in this report. Testing has shown that Power k-Means outperforms standard k-Means in nearly every metric. Graphs for each metric can be found below and are also available on the Github repo. CSV files containing the numerical information are also available on the Github repo.

NMI measures the mutual information, or shared datapoints, between each clustering such that a desirable NMI score reveals high similarity within data points in a single cluster and low similarity across different clusters. NMI is calculated using two metrics: mutual information and entropy. Two forms of entropy are considered: marginal and conditional entropy. Given the predicted clustering with k clusters, the marginal entropy is computed as

$$H(Pred) = -\sum_{i=1}^{k} p_{C_i} log(p_{C_i})$$

where p_{C_i} indicates the probability of a data point being in cluster C_i. If n_i points exist in the cluster and n points in the entire dataset, probability is $\frac{n_i}{n}$. Conditional entropy is calculated using the equation

$$H(Pred|True) = -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{i,j} log(\frac{p_{i,j}}{p_{C_i}})$$

where $p_{i,j}$ is the probability that a data point in true cluster $C_i \in Pred$

Mutual information is represented as

$$I(Pred, True) = H(Pred) - H(Pred|True)$$

The equation for NMI is expressed as

$$NMI(Pred, True) = \frac{I(Pred, True)}{\sqrt{H(True)H(Pred)}}$$

The graph of NMI is shown below:

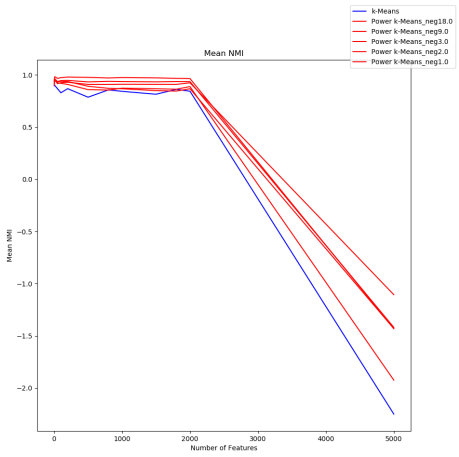

Figure 1: NMI over synthetic datasets with variable features and s_0 values

VI measures the distance between two partitions (clusterings) of the same dataset or, in other words, the amount of "information" gained or lost between the two. If this amount of " lost information," or difference in partitioned datapoints, is large, then the distance is greater and the similarity between the two is small. As clustering converges, a smaller VI denotes less entropy between the two and thus a stabilization to the clustering [1]. VI is calculated with the formula:

$$VI(True, Pred) = H(True|Pred) + H(Pred|True)$$

VI performance is shown below:

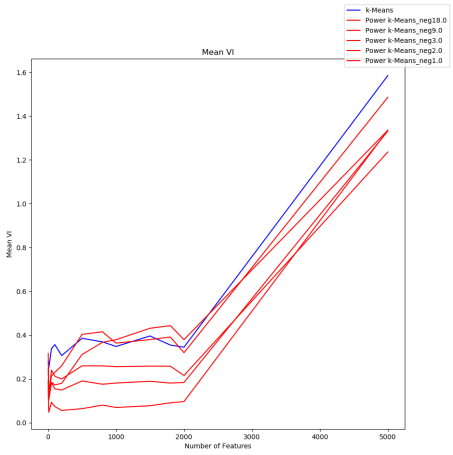

Figure 2: VI over synthetic datasets with variable features and s_0 values

Finally, the ARI measures the similarity of two clusterings, or the number of objects assigned to the same cluster or different clusters in two clusterings (eg, given two proposed clusterings, is datapoint x_i assigned to C_j in both clusterings). ARI is a modified form of the Rand Index metric, in which the labels of a clustering algorithm are compared against the true cluster labels for the dataset. Adjusted Rand Index makes use of a contingency table of size $n \times m$ where n is the number of predicted clusters and m is the number of true clusters (for k-means this is the same number). A contingency table displays frequency distribution of variables. In this case, cells in the contingency table represent the number of points which are common in the true and predicted clusterings for a single cluster and are denoted as $n_{i,j}$. Furthermore, the sums of each row and column are placed in an additional column and row respectively for ARI calculation.

True/Pred	C_1^T	$\cdots$	C_m^T	sum
C_1^P	$n_{1,1}$	$\cdots$	$n_{1,m}$	a_1
$\cdots$	$n_{i,1}$	$\cdots$	$n_{i,m}$	a_i
C_n^P	$n_{n,1}$	$\cdots$	$n_{n,m}$	a_n
sum	b_1	$\cdots$	b_m	

Table 1: Contingency table for ARI

At a high level, the ARI formula is defined as

$$ARI = \frac{RI - ExpectedRI}{max(RI) - ExpectedRI}$$

Using the table, the expected Rand Index is $\frac{\sum_i \binom{a_i}{2} \times \sum_j \binom{b_j}{2}}{\binom{n}{2}}$, or the sum of the number of pairs in the predicted clustering for each cluster times the sum of the number of pairs in the true clustering for each cluster divided by the total number of possible pairs in the dataset. The Rand Index is

calculated as

$$RI = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP represents true positives, TN represents true negatives, FP represents false positives, and FN represents false negatives. RI is thus the ratio of correct predictions to total predictions. Finally, $max(RI) = 1$, since Rand Index is calculated on a scale from 0 to 1.

Using the table, the ARI formula can thus be computed as such:

$$ARI = \frac{\sum_{i,j} \binom{n_{i,j}}{2} - [\sum_i \binom{a_i}{2} \times \sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}]/\binom{n}{2}}$$

The graph of ARI is shown below:

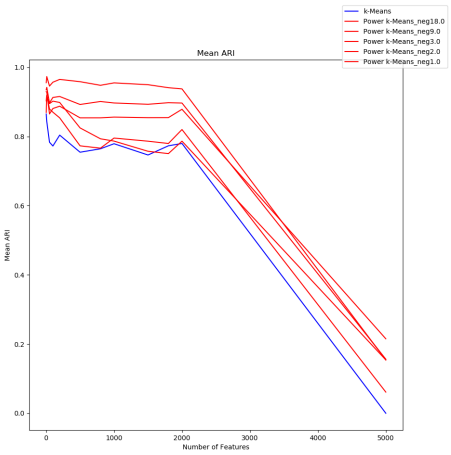

Figure 3: ARI over synthetic datasets with variable features and s_0 values

For the Real data sets, we used a low dimension (3), high cluster (510) set of roads in Denmark, and a high dimension (20532), low cluster (5) set of Cancer gene expressions. We chose these two to cover the extremes of both dimensionality and num of clusters. Unfortunately, being real data, we cannot compare individual variables at once, as each dataset represents something in the real world. The graphs for each metric is shown below:

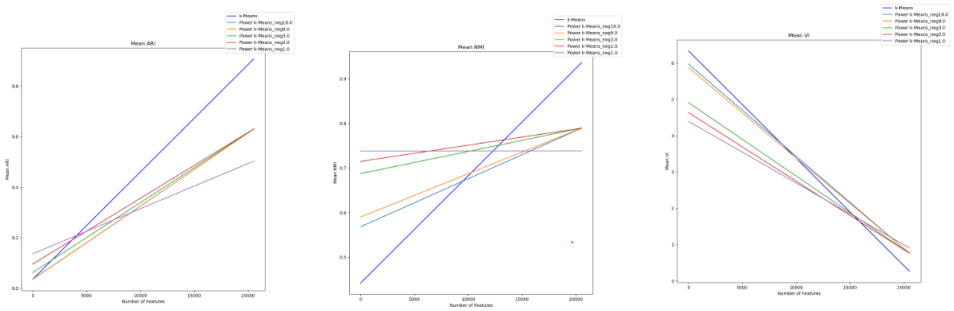

Figure 4: ARI, NMI, VI, for the low dimension dataset on the left, and the high dimension on the right, for both vanilla K means and several s_0 values

While Power K means performs better than K means on the low dimension data set, it performed worse in each metric for the large dimension data set. VI is bounded above by the log of the number of clusters, so the relatively large value above 1 makes sense. The initial value of s_0 also seems to matter much less in the high dimension data set, while initial conditions affect the performance much greater in the low dimension data. These findings suggest that Power K means may not be optimal for extremely high dimensional data in the real world.

Code for this project is available at `https://github.com/carbonkat/Power_K_Means_ML_Opt`

References

[1] Marina Meilă. "Comparing clusterings—an information based distance". In: *Journal of Multivariate Analysis* 98.5 (2007), pp. 873–895. ISSN: 0047-259X. DOI: https://doi.org/10.1016/j.jmva.2006.11.013. URL: https://www.sciencedirect.com/science/article/pii/S0047259X06002016.

[2] Adithya Vellal, Saptarshi Chakraborty, and Jason Q Xu. "Bregman Power k-Means for Clustering Exponential Family Data". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 17–23 Jul 2022, pp. 22103–22119. URL: https://proceedings.mlr.press/v162/vellal22a.html.

[3] Jason Xu and Kenneth Lange. "Power k-Means Clustering". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 6921–6931. URL: https://proceedings.mlr.press/v97/xu19a.html.

[4] Bin Zhang. "Generalized K-Harmonic Means - Dynamic Weighting of Data in Unsupervised Learning". In: *SDM*. 2001.

[5] Bin Zhang, Meichun Hsu, and Umeshwar Dayal. "K-Harmonic Means - A Data Clustering Algorithm". In: 1999.