An Iterative Algorithm for Optimal Variable Weighting in K-means Clustering

Shaonan Zhang, Shanshang Li, Jiaqiao Hu, Haipeng Xing, Wei Zhu

Department of Applied Mathematics and Statistics

Stony Brook University, NY, 11794-3600

The K-means clustering method is a widely adopted clustering algorithm in data mining and pattern recognition, where the partitions are made by minimizing the total within group sum of squares based on a given set of variables. Weighted K-means clustering is an extension of the K-means method by assigning nonnegative weights to the set of variables. In this paper, we aim to obtain more meaningful and interpretable clusters by deriving the optimal variable weights for weighted K-means clustering. Specifically, we improve the weighted k-means clustering method of Huh and Lim (2009) [1] by introducing a new algorithm to obtain the globally optimal variable weights based on the Karush-Kuhn-Tucker conditions. We present the mathematical formulation for the clustering problem, derive the structural properties of the optimal weights, and implement an recursive algorithm to calculate the optimal weights. Numerical examples on simulated and real data indicate that our method is superior in both clustering accuracy and computational efficiency.

Keywords: KKT conditions; K-means clustering; Lagrange multiplier; Optimization; Variable weights

1 Introduction

The K-means clustering method is a classical centroid-based clustering algorithm widely used in machine learning, data mining, pattern recognition, image analysis, and bioinformatics [2,3]. Assuming that we have n objects with m variables: $x_i = (x_{i1}, x_{i2}, \ldots, x_{im}), i = 1, 2, \ldots, n$, the objective of K-means clustering is to find the k cluster centroids that minimize the total within-group sum of squares (WGSS) as follows:

$$\sum_{g=1}^{k} \sum_{i \in I_g} \sum_{j=1}^{m} (x_{ij} - c_{gj})^2, \tag{1}$$

where I_g is the set of objects belonging to cluster $g = 1, 2, \ldots, k$ and $c_g = (c_{g1}, c_{g2}, \ldots, c_{gm})$ is the centroid of x_i. Note that in K-means clustering, the cluster number k often needs to be predetermined. This is usually carried out in practice either by experimenting with a set of values of k and then picking the best one or through additional information such as prior knowledge of problem structure or expert experience [4].

In standard K-means algorithm, the data are usually column-wise standardized and then iteratively partitioned into k clusters. The most commonly used standardization approach [5,6] is to scale each variable by the variable mean $\overline{x}_{\cdot j}$ and standard deviation s_j:

$$z_{ij} = \frac{x_{ij} - \overline{x}_{\cdot j}}{s_j}. \tag{2}$$

However, this standardization approach is not unique and needs to be chosen carefully based on the underlying data structure [7]. Intuitively, the variables may have different degrees of influence on the data structure. Thus, the clustering results may rely heavily on some variables while others may only enter the optimization problem in a superficial way. In 1984, Desarbo and colleagues [8] proposed a weighted K-means clustering algorithm that assigns each variable a non-negative weight to reflect its contribution to the WGSS. Since then, a varieties of weighted K-means clustering analysis algorithms have been proposed by Bradley and Usama [9], Kanungo, Tapas [10],Huang, Joshua Zhexue [11], and Modha, Dharmendra S. [12]. However, one common issue of these approaches is that the underlying algorithm may suffer from unstable behavior, because the optimal variable weighting can be very sensitive to the underlying data and the choice of parameters. In particular, it has been observed that the variable weights may subject to large fluctuation with even a small change in the parameter setting or the underlying dataset by removing some observations.

To address this instability issue, Huh and Lim proposed a revised weighted clustering objective function by finding the variable weights $w = (w_1, w_2, \ldots, w_m)$ that minimize the sum of the WGSS with a penalty term as follows:

$$\sum_{g=1}^{k}\sum_{i\in I_g}\sum_{j=1}^{m}\frac{w_j(z_{ij}-c_{gj})^2}{n-1} + \alpha\sum_{j=1}^{m}\frac{(w_j-1)^2}{m-1}, \tag{3}$$

where α is an additional parameter that penalizes the increased discrepancy among variable weights. When the penalty parameter α is chosen large, the minimization of equation equation (3) forces all weights to be close to 1, leading to small differences among variable weights, whereas small values of α generally allow large differences among variable weights. The idea is to stabilize the variable weights by carefully choosing the penalty parameter α. Huh and Lim proposed to use the process optimization in response surface methodology to estimate the optimal variable weights. However, the performance of their method is not very satisfactory on large data sets and the Nelder-Mead optimization algorithm [13] they used can be time consuming and may not guarantee a global optimal solution.

In this paper, we aim to improve the work of Huh and Lim by introducing a new algorithm to find the globally optimal variable weights in weighted K-means clustering. In Section 2, we analyze the structural properties of the optimal variable weights based on the well-known Karush-Kuhn-Tucker conditions [14,15] and propose an iterative procedure that exploits these properties to efficiently calculate the optimal variable weights. We carry out computational experiments in Section 3 to illustrate the performance of the algorithm and conclude the paper in Section 4. Our experimental results on both simulated and real data sets indicate that our algorithm is promising and may yield superior performance over existing approaches, especially on large data sets.

2 The Proposed Method

In this section, we formulate equation (3) as an optimization problem with inequality constraints and show that the optimal variable weights have a closed-form representation. We then develop an iterative algorithm for estimating the optimal variable weights in Section 2.1, calculate initial β (within-cluster mean squares on each variable) in Section 2.2, and propose a new data-driven approach to select the penalty parameter α in Section 2.3.

Note that by switching the order of the summations, the objective function (3) can be equivalently written as a function of variable weights $(w_1, w_2, \ldots, w_m)$:

$$\sum_{j=1}^{m} w_j \sum_{g=1}^{k}\sum_{i\in I_g}\frac{(z_{ij}-c_{gj})^2}{n-1} + \alpha\sum_{j=1}^{m}\frac{(w_j-1)^2}{m-1}. \tag{4}$$

Assuming that the true cluster centroids $c_g\,(g = 1, 2, \ldots, k)$ are known, we denote the coefficient of w_j as β_j, i.e., $\beta_j = \sum_{g=1}^{k}\sum_{i\in I_g}\frac{(z_{ij}-c_{gj})^2}{n-1}$. Without loss of generality, we assume that $\beta_1 \le \beta_2 \le \ldots \le \beta_m$. For a given α, the optimal variable weights can be obtained as the solution to the following quadratic optimization problem:

$$\begin{gathered} Minimize: f(w;\alpha) = \textstyle\sum_{j=1}^{m}\beta_j w_j + \alpha\sum_{j=1}^{m}\frac{(w_j-1)^2}{m-1} \\ Subject\ to: \textstyle\sum_{j=1}^{m} w_j = m; \\ w_j > 0, j = 1, 2, \ldots, m. \end{gathered} \tag{5}$$

Thus, by applying the method of Lagrange multiplier, equation (5) can be expressed as:

$$L(w,\lambda,\mu;\alpha) = \sum_{j=1}^{m}\beta_j w_j + \alpha\sum_{j=1}^{m}\frac{(w_j-1)^2}{m-1} + \lambda(\sum_{j=1}^{m} w_j - m) + \sum_{j=1}^{m}\mu_j w_j, \tag{6}$$

where λ and μ_j are the corresponding Lagrange multipliers. It is well-known that the optimal weights $(w_1, w_2, \ldots, w_m)$ satisfy the Karush-Kuhn-Tucker (KKT) conditions:

$$\frac{\partial L}{\partial w_j} = \beta_j + \frac{2\alpha}{m-1}(w_j - 1) + \lambda + \mu_j = 0, \quad j = 1, 2, \ldots, m \tag{7}$$

$$\sum_{j=1}^{m} w_j - m = 0 \tag{8}$$

$$\mu_j w_j = 0, \quad j = 1, 2, \ldots, m \tag{9}$$

$$w_j > 0, \quad j = 1, 2, \ldots, m. \tag{10}$$

The optimal variable weights can be shown to satisfy the following equations:

$$\begin{cases} w_j(\alpha, t_{opt}) = \frac{m}{t_{opt}} + \frac{(\overline{\beta}_{t_{opt}} - \beta_j)(m-1)}{2\alpha} & j \leq t_{opt} \\ w_j(\alpha, t_{opt}) = 0 & j > t_{opt}, \end{cases} \tag{11}$$

where t_{opt} is the optimal number of non-zero variable weights and $\overline{\beta_t} = \frac{\sum_{i=1}^{t} \beta_i}{t}$; see Appendix 1 for detailed derivation steps.

2.1 An Iterative Algorithm

Equation (11) provides the closed-form expression for optimal variable weights in k-means clustering when β (within-cluster mean squares on each variable) is known for all variables. However, the actual value of β is unknown unless the clustering partition is given. To address this issue, we propose a recursive procedure to iteratively estimate β, α, and subsequently the optimal variable weights:

Step 1. Standardize data matrix using equation (2). Specify an initial β and the corresponding penalty parameter α.

Step 2. Given β and α, calculate the optimal variable weights $(w_1, w_2, \ldots, w_m)$ according to equation (11).

Step 3. Run k-means clustering on weighted variable $Z^* = Z * D$, where D is a diagonal matrix with diagonal entries $Diag(D) = (\sqrt{w_1}, \sqrt{w_2}, \ldots, \sqrt{w_m})$. Calculate within-cluster mean squares on each variable β_j and update penalty parameter α accordingly.

Step 4. Repeat steps 2 and 3 until the parameter β converges.

The choices of initial β values and the determination of the penalty parameter α are discussed in detail in the following subsections.

2.2 Initial β Estimation

From equation (4), we have the following linear relationship between the overall within cluster sums of squares on weighted variables Z^* and β

$$\begin{aligned} \sum_{j=1}^{m} \beta_j w_j &= \sum_{j=1}^{m} w_j \sum_{g=1}^{k} \sum_{i \in I_g} \frac{(z_{ij} - c_{gj})^2}{n-1} \\ &= \frac{1}{n-1} \sum_{g=1}^{k} \sum_{i \in I_g} \sum_{j=1}^{m} (z^*_{ij} - c^*_{gj})^2, \end{aligned} \tag{12}$$

where $z^*_{ij} = z_{ij}\sqrt{w_j}$, $c^*_{gj} = c_{gj}\sqrt{w_j}$ are the re-scaled variables and the corresponding cluster centroids.

Given the constraint that all the variable weights w^j sum up to m, we formulate the following canonical mixture linear model with β as coefficients and y being the within cluster mean squares on weighted variable Z^*. Here ε is the white noise error term.

$$y = \sum_{j=1}^{m} \beta_j w_j + \varepsilon. \tag{13}$$

To estimate the initial β, we apply a $\{m, 2\}$ simplex lattice design [16] with center point to generate initial variable weights and estimate β afterward. Generally, a $\{m, p\}$ simplex lattice design generates a set of m-dimensional points $(x_1, x_2, \ldots, x_m)$ such that each component can take $p+1$ equally spaced values from 0 to 1, that is, $x_i = 0, 1/p, 2/p, \ldots, 1;\ for\ i = 1, 2, \ldots, m$ and the sum of all the components equal to 1. Graphically, it consists of all m vertices and p-equal-division-points on $\begin{pmatrix} m \\ 2 \end{pmatrix}$ edges of $(m-1)$ dimensional simplex. For example (See Figure 1), a {3,2} simplex lattice design consists of 6 points, which are the 3 vertices, and the midpoints of 3 edges.

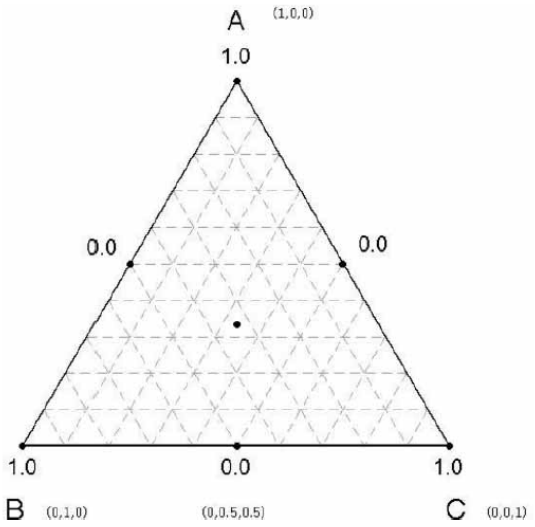

Figure 1: An example of {3,2} simplex design with the center point

In our algorithm, we generate an $\{m, 2\}$ simplex lattice design to obtain a set of vectors $p = (p_1, p_2, \ldots, p_m)$. Then for each design point $\boldsymbol{p}$, we run the k-means clustering on weighted variable with weight $w = m * p$ and calculate the overall within cluster sum of squares and subsequently the response variable y in model (13). After this, we fit the linear model and calculate the least square estimators as the initial β.

2.3 Selection of Penalty Parameter α

In the optimal solution, equation (11), the nonnegative parameter α is the penalty for heterogeneity in variable weighting, and also a tuning parameter to stabilize the optimal weights.

It can be shown that when α stays in a certain range, the clustering partition remains the same. In fact, we prove in Appendix 2 that there is a unique vector $g = \left\{ g_i := \frac{t(\beta_i - \overline{\beta}_i)(m-1)}{2m},\ i = 1, 2, \ldots, m \right\}$ that splits α into $(m+1)$ ranges and the optimal clustering partition remains the same when α changes within each range. Therefore, the determination of α is essentially equivalent to determining the value of t so that $\alpha \in (g_t, g_{t+1}]$ is chosen. For simplicity, we will choose $\alpha = \frac{g_t + g_{t+1}}{2}$ after t is determined.

Here we introduce an efficient measurement, Reduced Variation (RV), to determine t. The RV of the i^{th} variable is defined as follows:

$$RV_i = \frac{1-\beta_i}{\sum_{i=1}^{m}(1-\beta_i)};\ \sum_{i=1}^{m} RV_i = 1. \tag{14}$$

Then we will select

$$t^* = min\left\{ t \mid \sum_{i=1}^{t} RV_i > \frac{m-1}{m} \right\} \tag{15}$$

and therefore

$$\alpha = \frac{g_{t^*} + g_{t^*+1}}{2} \tag{16}$$

Similar to the variable selection problem, there is always an argument about the balance between removing noise and losing information. From our experience on various datasets, the threshold of $\frac{m-1}{m}$ on cumulative RVs always shows stable performance in terms of removing noisy variables without losing too much information. In practice, such a threshold value could also be determined based on prior knowledge of data structure or by experimenting with different threshold values.

3 Numerical Results

To illustrate the performance of our proposed method, we consider some computational experiments on different simulated datasets (Section 3.1) and four real datasets in different fields (Section 3.2). In Section 3.3, the performance of our method is compared with that of Huh and Lim using the same datasets.

3.1 Simulation Data

Simulated Case 1: Five 3-dimensional Gaussian groups with 100 observations in each group contain two informative variables and one noisy variable. Each group follows 3-dimensional multivariate normal, $N(\mu, I_3)$. Five group means are (5,0,0), (-5,0,0), (0,5,0), (0,-5,0), (0,0,0). With the 3D pictures of the simulated data, we can easily see that two components are signals while the third one is noise. (First group dataset, see Figure 2)

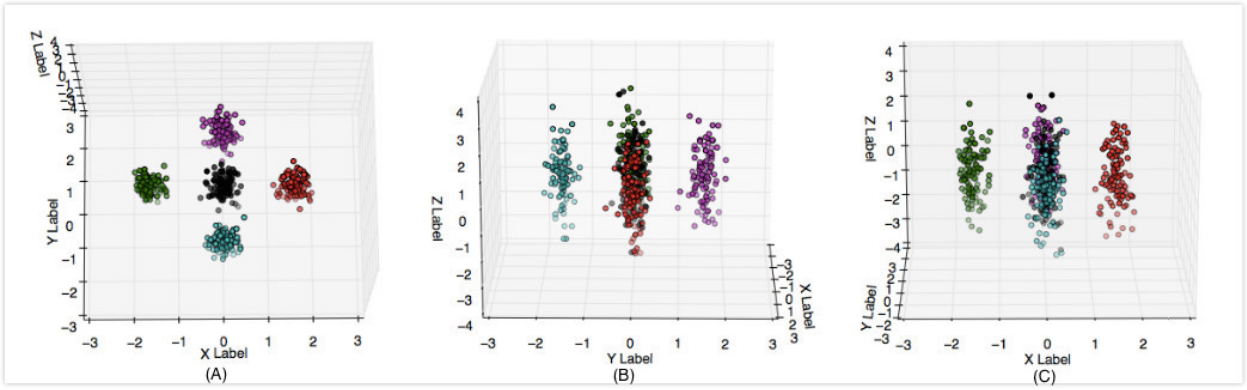

Figure 2: Scatterplots of First group dataset from different perspectives

For the first dataset, our algorithm takes only two iterations in less than 1 second. The β of the first two informative variables are very small while the β of the third noisy variable is almost 1 −− as expected (See Table 1). Also the algorithm correctly indicates $t_{selected} = 2$, which is the true number of informative variables. With the refined estimation on β and α, we calculated the optimal variable weighting derived in equation (11), and the cluster partition is shown in classification table (See Table 2). We can see that all objects are correctly classified.

Table 1: Parameters of the First group dataset

	β	$t_{selected}$	$\alpha_{selected}$	variable weights
Initial	0.0871, 0.0875, 0.2222	2	0.0450	1.5056, 1.4944, 0
1st iteration	0.0272, 0.0275, 0.9963	2	0.3230	1.5056, 1.4944, 0
2nd iteration	0.0272, 0.0275, 0.9963	2	0.3230	1.5004, 1.4996, 0

Table 2: Partition results of the First group dataset

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Group 1	100	0	0	0	0
Group 2	0	100	0	0	0
Group 3	0	0	100	0	0
Group 4	0	0	0	100	0
Group 5	0	0	0	0	100

Simulated Case 2: From Figure 2(A), we notice that the centroids of observations in each group are quite scattered. So we reset the group means to (1,0,0), (-1,0,0), (0,1,0), (0,-1,0), (0,0,0) (Second group dataset, see Figure 3(B)) and (3,0,0), (-3,0,0), (0,3,0), (0,-3,0), (0,0,0) (Third group dataset; Figure 3(C)).

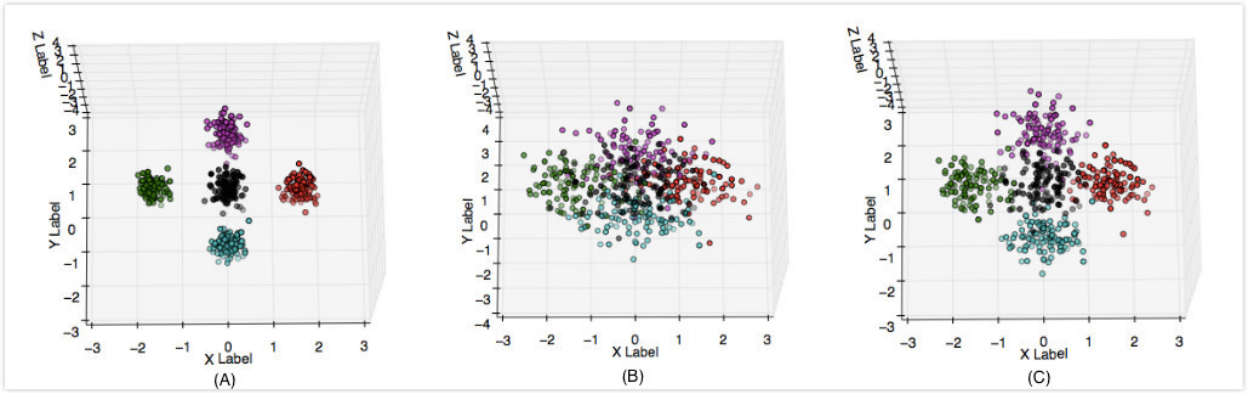

Figure 3: (A) Scatterplots of First group dataset; (B) Scatterplots of Second group dataset; (C) Scatterplots of Third group dataset

For the second and third group datasets, the estimated β and α obtained in each iteration and the resulting partitions are reported in Tables 3 and 4 and Tables 5 and 6, respectively. From the tables, we see that more than half of the observations are correctly clustered in all cases. In addition, the number of mis-clustered points increases as the centroids of observations get close to each other.

Table 3: Parameters of the Second group dataset

	β	$t_{selected}$	$\alpha_{selected}$	variable weights
Initial	0.1928,0.2000,0.2227	2	0.0099	1.8593,1.1407,0
1st iteration	0.2122,0.2701,0.9958	2	0.2612	1.8593,1.1407,0
2nd iteration	0.2240,0.2530,0.9952	2	0.2571	1.6108,1.3891,0
3rd iteration	0.2265,0.2503,0.9950	2	0.2561	1.5565,1.4435,0
4th iteration	0.2341,0.2421,0.9945	2	0.2535	1.5464,1.4536,0
5th iteration	0.2286,0.2472,0.9944	2	0.2553	1.5159,1.4841,0
6th iteration	0.2285,0.2473,0.9944	2	0.2553	1.5364,1.4636,0
7th iteration	0.2285,0.2473,0.9945	2	0.2553	1.5368,1.4632,0
8th iteration	0.2285,0.2473,0.9945	2	0.2553	1.5368,1.4632,0

Table 4: Partition results of the Second group dataset

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Group 1	69	0	3	4	24
Group 2	0	78	4	8	10
Group 3	6	4	66	2	22
Group 4	4	3	0	89	4
Group 5	12	11	10	19	48

Table 5: Parameters of the Third group dataset

	β	$t_{selected}$	$\alpha_{selected}$	variable weights
Initial	0.1483,0.1492,0.2231	2	0.0249	1.5174,1.4826, 0
1st iteration	0.1349,0.1351,0.9969	2	0.2873	1.5174,1.4826, 0
2nd iteration	0.1350,0.1351,0.9969	2	0.2873	1.5003,1.4997, 0
3rd iteration	0.1350,0.1351,0.9969	2	0.2873	1.5002,1.4998, 0

Table 6: Partition results of the Third group dataset

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Group 1	94	0	0	0	6
Group 2	0	95	1	1	3
Group 3	0	0	96	0	4
Group 4	1	0	0	97	2
Group 5	3	5	2	5	85

Simulated Case 3: Note that in Case 1, the numbers of observations are the same across different groups. So we considered two other group datasets. In the fourth group dataset (shown in Figure 4(B)), we changed the number of observations in each respective group to (100, 80, 60, 40, 20); whereas in the fifth dataset (Figure 4(C)), we increased the sample sizes of different groups to (100, 200, 300, 400, 500).

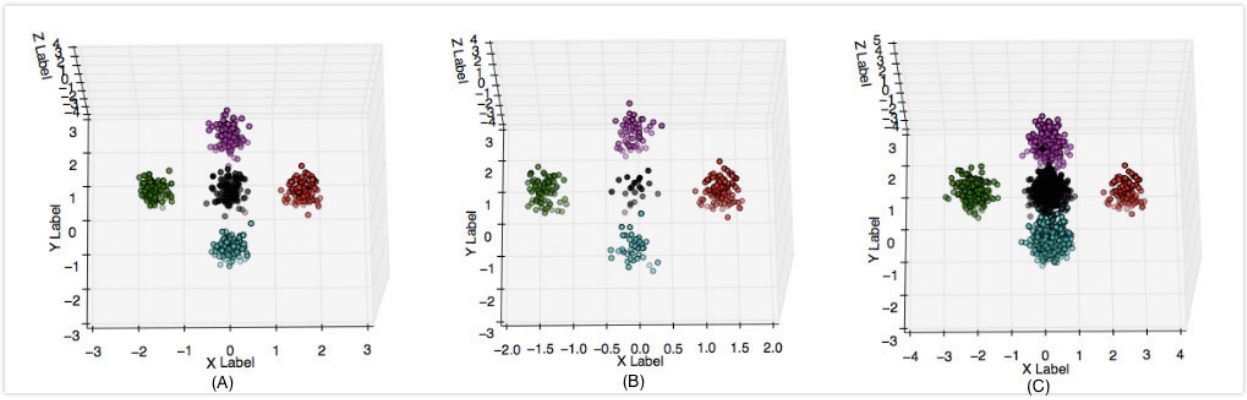

Figure 4: (A) Scatterplots of First group dataset; (B) Scatterplots of Fourth group dataset; (C) Scatterplots of Fifth group dataset

For the fourth and fifth group dataset, the estimated parameters in each iteration and partition results are listed in Tables 7, 8 and Tables 9, 10. We observe that the partition results are not influence by the number of observations in each group since all objects are correctly classified.

Table 7: Parameters of the Fourth group dataset

	β	$t_{selected}$	$\alpha_{selected}$	variable weights
Initial	0.0766, 0.0813, 0.2149	2	0.0461	1.5517, 1.4483, 0
1st iteration	0.0184, 0.0340, 0.9882	2	0.3233	1.5517,1.4483, 0
2nd iteration	0.0184, 0.0340, 0.9882	2	0.3233	1.5240, 1.4760, 0

Table 8: Partition results of the Fourth group dataset

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Group 1	100	0	0	0	0
Group 2	0	80	0	0	0
Group 3	0	0	60	0	0
Group 4	0	0	0	40	0
Group 5	0	0	0	0	20

Table 9: Parameters of the Fifth group dataset

	β	$t_{selected}$	$\alpha_{selected}$	variable weights
Initial	0.0878, 0.0973, 0.2148	2	0.0423	1.6116, 1.3884, 0
1st iteration	0.0248, 0.0582, 0.9966	2	0.3239	1.6116, 1.3884, 0
2nd iteration	0.0248, 0.0582, 0.9966	2	0.3239	1.5516, 1.4484, 0

Table 10: Partition results of the Fifth group dataset

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Group 1	100	0	0	0	0
Group 2	0	200	0	0	0
Group 3	0	0	300	0	0
Group 4	0	0	0	400	0
Group 5	0	0	0	0	500

Simulated Case 4: To test the performance of our method on high-dimensional large dataset, we created seven 8-dimensional Gaussian groups with 100 observations in each group. The 8 variables include three informative variables and five noisy variables. Each group follows an 8-dimensional multivariate normal distribution, $N(\mu, I_8)$. The seven group means are (-10, 0, 0, 0, 1, 1, 0, 0), (-10, 0, 0, 0, 1, 1, 0, 0), (-10, 0, 0, 0, 1, 1, 0, 0), (-10, 0, 0, 0, 1, 1, 0, 0), (0, 0, -10, 0, 1, 1, 0, 0), (0, 0, -10, 0, 1, 1, 0, 0), (0, 0, -10, 0, 1, 1, 0, 0) (the scatter plot of the sixth dataset is shown in Figure 5(A)). We also considered another dataset with smaller distances between centroids, where each group contains 200 observations and the group means are given by (-5, 0, 0, 0, 1, 1, 0, 0), (-5, 0, 0, 0, 1, 1, 0, 0), (-5, 0, 0, 0, 1, 1, 0, 0), (-5, 0, 0, 0, 1, 1, 0, 0), (0, 0, -5, 0, 1, 1, 0, 0), (0, 0, -5, 0, 1, 1, 0, 0), (0, 0, -5, 0, 1, 1, 0, 0)(the seventh group dataset is shown in Figure 5(B)).

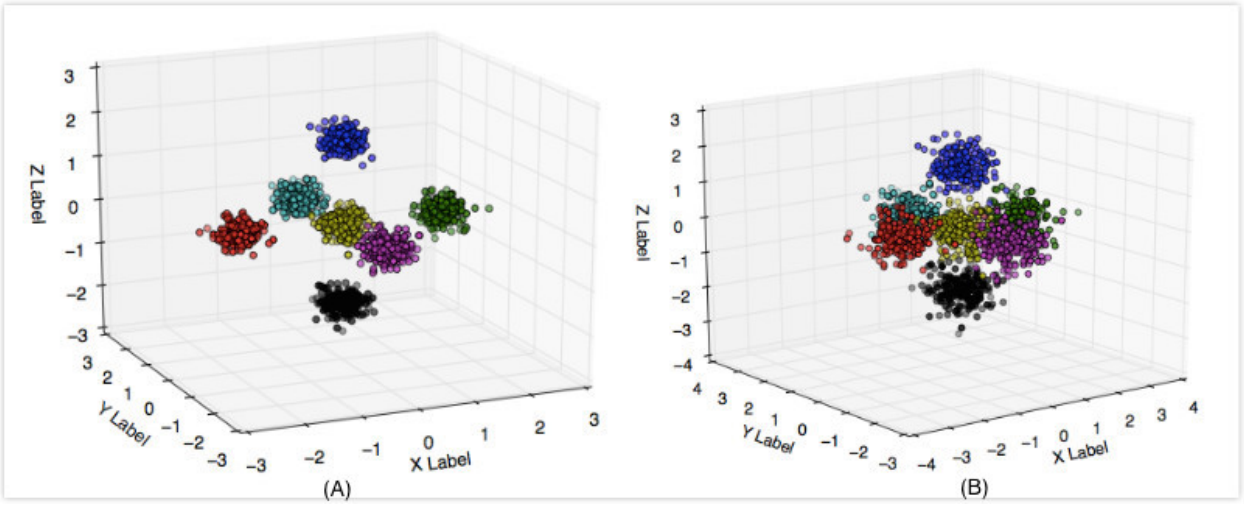

Figure 5: (A) Scatterplots of Sixth group dataset; (B) Scatterplots of Seventh group dataset

For the sixth and seventh datasets, our algorithm correctly indicates $t_{selected} = 3$, which is exactly the number of informative variables we create. The estimated parameters in each iteration and partition results are given in Tables 11, 12 and Tables 13, 14.

Table 11: Parameters of the Sixth group dataset

	β	$t_{selected}$	$\alpha_{selected}$	variable weights
Initial	0.0614,0.0638,0.064, 0.1842,0.1848,0.1866, 0.1871, 0.1923	7	0.1732	2.5922,2.5438,2.5393, 0.1108,0.0981,0.0632, 0.0526,0
1st iteration	0.0327,0.0336,0.0358, 0.9950,0.9950,0.9956, 0.9965, 0.9991	3	0.6318	2.5922,2.5438,2.5393, 0.1108,0.0981.0.0632, 0.0526,0
2nd iteration	0.0327,0.0336, 0.0358, 0.9950,0.9950,0.9956, 0.9965,0.9991	3	0.6318	2.6740,2.6690,2.6570, 0,0,0, 0,0

Table 12: Partition results of the Sixth group dataset

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	
Group 1	100	0	0	0	0	0	0
Group 2	0	100	0	0	0	0	0
Group 3	0	0	100	0	0	0	0
Group 4	0	0	0	100	0	0	0
Group 5	0	0	0	0	100	0	0
Group 6	0	0	0	0	0	100	0
Group 7	0	0	0	0	0	0	100

Table 13: Parameters of the Seventh group dataset

	β	$t_{selected}$	$\alpha_{selected}$	variable weights
Initial	0.1178,0.1206,0.1221, 0.1900,0.1903,0.1923, 0.1928,0.1980	7	0.1059	2.5663,2.4724,2.4241, 0.1800,0.1703,0.1017, 0.0853,0
1st iteration	0.1188,0.1216,0.1275, 0.9953,0.9956,0.9956, 0.9963,0.9993	3	0.5758	2.5663,2.4724,2.4241, 0.1800,0.1703,0.10166, 0.0853,0
2nd iteration	0.1188,0.1216,0.1275, 0.9953,0.9956,0.9956, 0.9963,0.9993	3	0.5758	2.6900,2.6727,2.6372, 0,0,0, 0,0

Table 14: Partition results of the Seventh group dataset

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 7	Cluster 7
Group 1	199	0	1	0	0	0	0
Group 2	0	200	0	0	0	0	0
Group 3	0	0	196	0	0	0	4
Group 4	0	0	0	199	0	1	0
Group 5	0	0	0	0	200	0	0
Group 6	0	0	0	0	0	199	1
Group 7	2	0	1	1	2	0	194

3.2 Real Datasets

Iris Data: This is a well-known dataset in the pattern recognition literature. The dataset contains 150 instances of three types of iris, 50 instances each. For each instance, sepal length, sepal width, petal length and petal width were measured in cm as 4 variables. For this dataset, the estimated parameters in each iteration and partition results are listed in Tables 15 and 16. We see that only 6 instances are mistakenly clustered.

Table 15: Parameters of the Eighth group dataset

	β	$t_{selected}$	$\alpha_{selected}$	variable weights
Initial	0.0818,0.0977,0.2176,0.3096	3	0.1477	1.8464,1.6860,0.4676,0
1st iteration	0.0602,0.0620,0.3468,0.5848	3	0.3482	1.8464,1.6860,0.4676,0
2nd iteration	0.0602,0.0620,0.3468,0.5848	3	0.3482	1.7475,1.7400,0.5126,0

Table 16: Partition results of the Eighth group dataset

	Cluster 1	Cluster 2	Cluster 1
Group 1	50	0	0
Group 2	0	49	1
Group 3	0	5	45

Wholesale Customers data: This marketing and management dataset contains 440 instances of three regions, 77, 47 and 316 respectively. For each instance, fresh products, milk, grocery, frozen products, paper products, delicatessen products were measured as 6 variables. Using our algorithm, we get the partition results and count the misclassified cluster members in all clusters and divide it by the total number of observations as the misclassification rate (MR). The optimal weights and corresponding results are shown in Table 17.

Yeast Data: In this data, there are 1484 observations in a total of 10 classes of yeasts. Then different measurements were transformed into 8 variable parameters from which the yeast features can be computed. The optimal weights and partition results are given in Table 17.

Contraceptive Method Choice Dataset: This dataset including 1473 samples is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The problem is to predict the current contraceptive method choice (3 clusters: no use, long-term methods, or short-term methods) of a woman based on her demographic and socio-economic characteristics, which includes 9 features. The optimal weights and partition results are listed in Table 17.

Table 17: Comparison results of the real datasets

	Instances/ Obsevations	Variables/ Features	Iteration	Optimal Variable Weights	Misclassification Rate
Iris Data	150	4	2	1.7475,1.7400, 0.5126,0	6/150
Wholesale Data	440	6	4	2.4242,2.2391,1.3367,0,0,0,0	109/440
Yeast Data	1484	8	8	2.342,2.3423,1.3733,1.0040,0.9382,0,0,0	602/1484
CMC Data	1473	9	2	4.4570,4.2607,0.2637,0.0187,0,0,0,0,0	321/1473

3.3 Numerical Comparison

In this section, we compare the performance of our method to that of Huh and Lim. To allow for a fair comparison of both algorithms, we use the first and sixth group dataset in Section 3.1 and iris data in Section 3.2 for both two methods. We adopt the same mean-variance standardization used in Huh and Lim paper and use multiple initial points to initialize both algorithms. In particular, we note that the Nelder-Mead simplex method they used is primarily designed for unconstrained optimization and is not known to be globally convergent. Consequently, their approach may likely lead to non-optimal clustering partitions in practice. Therefore, we focus on the following two aspects in our comparison: Algorithm Stability and Clustering Accuracy.

3.3.1 Algorithm Stability

We compare the algorithm stability by plotting the weighting curves on a set of penalty parameters α ranging from 2^{-5} to 2^5 with an increment of 0.01. This is very critical, especially for the method of Huh and Lim. Because in their method, this graph is used to locate a feasible range of α with stable variable weighting. If the algorithm is not stable, it will be very difficult to find the feasible range.

Recall from equation (3), when α increases from 0 to 1, the penalty part is emphasized and therefore all weights in the objective function will gradually move towards 1. In the following figures (See Figure 6, 7, 8), we see that our method captures this movement very nicely in all three datasets with all weights moving slowly towards 1. However, for the method in Huh and Lim, this behaviour is observed only in the First group of simulated data and the iris data with some outliers. In the Sixth group of simulated data, their method failed to capture this movement. We see that the plot looks more like a collection of random points rather than an expected weighting curve. This is because the Nelder-Mead method is only designed for unconstrained optimization problems. In our problem, each variable weight is required to be bounded between 0 and m. Thus the Nelder-Mead method fails to find the global optimal solution, and gets trapped at local optima, especially when the dimensionality increases and the data structure becomes more complex. In this case, their α selection approach, which based on the weighting curve, is not reliable.

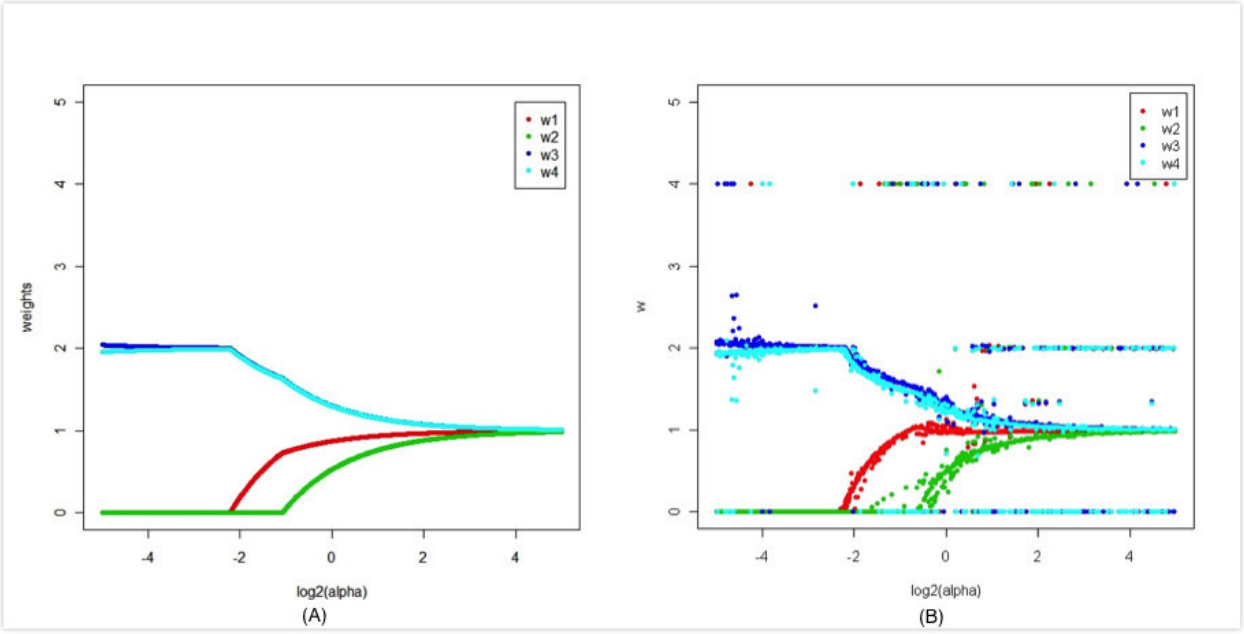

Figure 6: (A) Weighting curve of the First group dataset using our method; (B) Weighting curve of the First group dataset using Huh and Lim's method

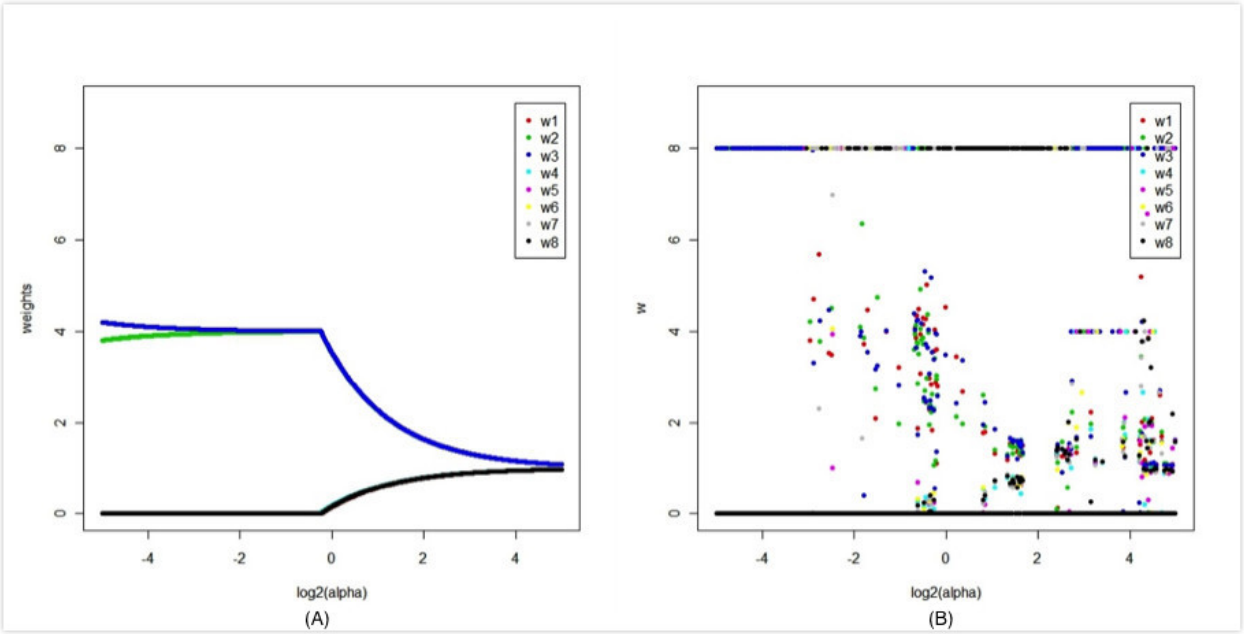

Figure 7: (A) Weighting curve of the Sixth group dataset using our method; (B) Weighting curve of the Sixth group dataset using Huh and Lim's method

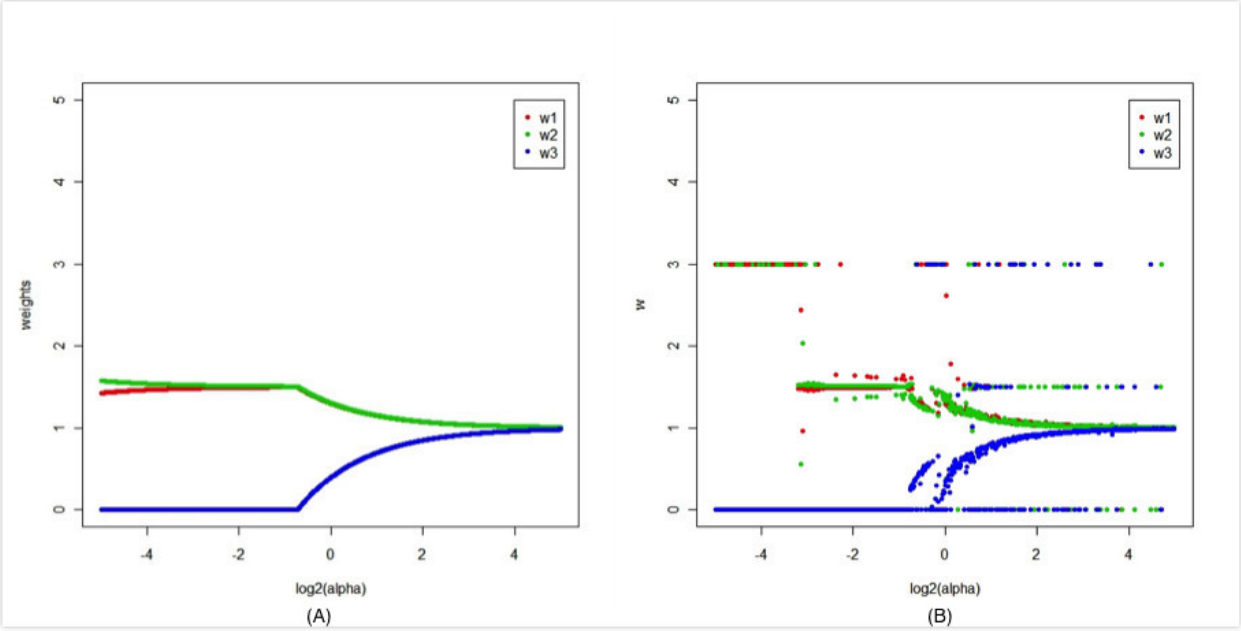

Figure 8: (A) Weighting curve of the iris dataset using our method; (B) Weighting curve of the iris dataset using Huh and Lim's method

3.3.2 Clustering Accuracy

To investigate the clustering accuracy, we compared the misclassification rate of both methods. For each dataset, first, each cluster is identified as the group of majority members in the cluster. Then we present the misclassification rate (MR) in both ratio and percentage, which are shown in Table 18. Optimal penalty parameter α was determined using our approach, equation (16), and the same α is used for both methods, which is different from the one used in Huh and Lim paper. Actually, they use weighting curve to determine the penalty parameter, which is very subjective to do so.

We can see that our method outperforms the original method by Huh and Lim with much lower misclassification rates. As pointed out in the previous section, due to the unstable behavior of their methods, the original method they proposed fail to identify the underlying true range for α and result in inferior performance.

The corresponding variable weights are also listed below (See Table 18). Our method finds different variable weighting than the original method, which results in better clustering partition. It is interesting to note that a slight difference in variable weighting can lead to a big difference in clustering partition as seen in all datasets. We also note that our method is always able to distinguish the informative variables from the noisy variables by assigning different weights. However the previous method fails in the sixth group of simulated data. This result is somehow as expected, since in lower dimensional data, the Nelder-Mead method performs relatively well, whereas as the dimensionality increases, the Nelder-Mead Simplex method may perform increasingly poorly and fail to find the global optimal for the constrained optimization problems.

Table 18: Comparison results of our method and Huh & Lim's method

		1st group dataset	6th group dataset	Iris Data
Our Method	MR	0/500	0/700	6/150
	MR in %	0	0	4.0%
	Optimal Weights	1.50,1.50, 0	2.6740,2.6690,2.6570,0,0,0,0,0	1.7475,1.7400,0.5126,0
Lim and Huh	MR	1/500	189/700	14/150
	MR in %	0.2%	27.0%	9.3%
	Optimal Weights	1.49,1.51,0	0,3.57,4.43,0,0,0,0,0	0.58,0,1.75,1.67

4 Conclusion

It is well understood that K-means clustering algorithm is ideal for detecting clusters that are homogenous and spherical and without the presence of noise variables. Weighted K-means algorithm can improve the performance on non-homogenous and non-spherical cases by suppressing the noise variables and transferring the non-spherical space into a spherical space with appropriate variable weighting. However, most existing studies are unable to find stable variable weights. Recently, Huh and Lim proposed a novel penalized objective function for weighted k-means problem that yields more stable and reasonable solutions for low dimensional case with a few variables.

In this paper, by adopting the same objective function used by Huh and Lim, we propose a more suitable optimization method to select the penalty parameter α and an improved iteration algorithm to achieve the optimal variable weights. Our preliminary data analysis indicates that our method can significantly improve the original method of Huh and Lim in terms of both algorithm stability and clustering accuracy, especially on high-dimensional datasets.

Since our method provides a closed form representation of optimal variable weights, it is more computationally efficient and can be utilized in high dimensional dataset, such as those arising in bioinformatics and financial market. However, discovering the natural structure in a high dimensional space itself is a non-trivial task no matter what algorithm is applied. It would be very interesting to validate our method on real high dimensional datasets, but that would not be the main focus in this paper.

References

[1] Huh, Myung-Hoe, and Yong B. Lim. "Weighting variables in K-means clustering". *Journal of Applied Statistics*, 36.1 (2009): 67-78.

[2] Hartigan, John A., and Manchek A. Wong. "Algorithm AS 136: A k-means clustering algorithm". *Applied statistics*, (1979):100-108.

[3] Jain, Anil K. "Data clustering: 50 years beyond K-means". *Pattern recognition letters*, 31.8 (2010): 651-666.

[4] Wagstaff, Kiri, et al. "Constrained k-means clustering with background knowledge". *ICML*.Vol. 1. 2001.

[5] Steinley, Douglas. "K-means clustering: a half century synthesis.". *British Journal of Mathematical and Statistical Psychology*, 59.1 (2006): 1-34.

[6] Steinley, Douglas. "Standardizing variables in K-means clustering.". *Classification, clustering, and data mining applications*, Springer Berlin Heidelberg, 2004. 53-60.

[7] Kaufman, Leonard, and Peter J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, Vol. 344. John Wiley & Sons, 2009.

[8] W. DeSarbo et al.. "Refining Initial Points for K-Means Clustering". *Psychometrika*, 49 (1984), pp. 57-78.

[9] Bradley, Paul S., and Usama M. Fayyad. "Synthesized clustering: A method for amalgamating alternative clustering bases with differential weighting of variables". *ICML*, Vol. 98. 1998.

[10] Kanungo, Tapas, et al.. "An efficient k-means clustering algorithm: Analysis and implementation". *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24.7 (2002): 881-892.

[11] Huang, Joshua Zhexue, et al.. "Automated variable weighting in k-means type clustering". *Automated variable weighting in k-means type clustering*, 27.5 (2005): 657-668.

[12] Modha, Dharmendra S., and W. Scott Spangler. "Feature weighting in k-means clustering". *Machine learning*, 52.3 (2003): 217-237.

[13] Lagarias, Jeffrey C., et al.. "Convergence properties of the Nelder–Mead simplex method in low dimensions". *Convergence properties of the Nelder–Mead simplex method in low dimensions*, 9.1 (1998): 112-147.

[14] W. Karush. "Minima of Functions of Several Variables with Inequalities as Side Constraints". *University of Chicago*, 1939.

[15] H.W. Kuhn, and A.W. Tucker. "Nonlinear Programming", *in 2nd Berkeley Symposium on Mathematical Staitistics and Probability, University of California Press, Berkeley*, 1951, pp. 481-492.

[16] Myers, Raymond H., Douglas C. Montgomery, and Christine M. Anderson-Cook. "Myers, Raymond H., Douglas C. Montgomery, and Christine M. Anderson-Cook.". *John Wiley & Sons*, John Wiley & Sons, 2009.

A Appendix 1. Derivation of optimal variable weights

In order to derive the closed-form solution on the optimal variable weights, we first show the following result:

Proposition 1. *Let $w_i^*,\ i = 1, 2, \ldots, m$ be the optimal solutions to equation (2). Then $w_i^* > w_j^*$ if and only if $\beta_i < \beta_j$.*

Proof. Assume $\exists\ w_i^* < w_j^*$ and $\beta_i < \beta_j$, then

$$\begin{aligned} \beta_i w_i^* + \beta_j w_j^* - \beta_i w_j^* - \beta_j w_i^* &= (\beta_i - \beta_j)(w_i^* - w_j^*) > 0 \\ \Leftrightarrow \beta_i w_i^* + \beta_j w_j^* &> \beta_i w_j^* + \beta_j w_i^* \end{aligned} \tag{17}$$

Therefore, by switching the order of w_i^* and w_j^*, one can construct another set of weights $\overline{w}$ such that $f(\overline{w}; \alpha) < f(w_i^*, \alpha)$, which contradicts the fact that w_i^* is the optimal solution in equation (2).

Now we start to solve the system of KKT conditions (7-10) derived from the KKT conditions. Firstly, from equation (9), we know for each j, either μ_j or w_j must be zero. Assuming there are t variables that have nonzero weights, based on inequation (10) and Proposition 1, we have:

$$w_1 \geq w_2 \geq \ldots \geq w_t > 0 = w_{t+1} = \ldots = w_m \tag{18}$$

$$\mu_1 = \mu_2 = \ldots = \mu_t = 0 \tag{19}$$

By substituting the above into equations (7) and (8), we can obtain the t solutions with nonzero weights:

$$\begin{cases} w_j(\alpha, t) = \frac{m}{t} + \frac{(\overline{\beta}_t - \beta_j)(m-1)}{2\alpha} & j \leq t \\ w_j(\alpha, t) = 0 & j > t \end{cases} \tag{20}$$

where $\overline{\beta}_t = \frac{\sum_{i=1}^{t} \beta_i}{t}$, $\lambda(\alpha, t) = \frac{2\alpha(t-m)}{t(m-1)} - \overline{\beta}_t$.

In order to find the optimal variable weights, we need to decide the number of nonzero variable weights t. From equation (18), we obtain:

$$w_t = \frac{m}{t} + \frac{(\overline{\beta}_t - \beta_j)(m-1)}{2\alpha} > 0 \Rightarrow \alpha > \frac{t(\beta_t - \overline{\beta}_t)(m-1)}{2m} \tag{21}$$

In the following, we denote

$$g(t) = \frac{t(\beta_t - \overline{\beta}_t)(m-1)}{2m} \tag{22}$$

Since $\beta_1 \leq \beta_2 \leq \ldots \leq \beta_m$, it is obvious that $g(t)$ is a monotone function. Therefore, for a given α, the set of all possible values of t satisfying equation (18) is given by:

$$T(\alpha) = \{t \mid g(t) < \alpha \leq g(t+1)\} \tag{23}$$

In view of equation (20), problem equation (2) can be re-formulated in terms of t, whose optimal solution can be obtained by solving the following optimization problem:

$$t_{opt} = argmin_{t \in T(\alpha)} f(w; \alpha) \tag{24}$$

Finally, replacing t by t_{opt} in equation (20), we obtain the optimal variable weights:

$$\begin{cases} w_j(\alpha, t_{opt}) = \frac{m}{t_{opt}} + \frac{(\overline{\beta}_{t_{opt}} - \beta_j)(m-1)}{2\alpha} & j \leq t_{opt} \\ w_j(\alpha, t_{opt}) = 0 & j > t_{opt} \end{cases} \tag{25}$$

□

B Appendix 2. Proof of clustering partition

Here we prove the following result, which shows the unique clustering partition for $\alpha \in (g(t),\ g(t+1)]$, where $g(t) = \frac{t(\beta_t - \overline{\beta}_t)(m-1)}{2m},\ t = 1, 2, \ldots, m$.

Proposition 2. *Assuming the uniqueness of cluster assignment, if* $\exists\ \alpha^* \in (g(t),\ g(t+1)]$, *so that* $\overrightarrow{z_0} = \{z_{0j}\}$ *belongs to cluster* g_0, *then for all* $\alpha \in (g(t),\ g(t+1)]$, $\overrightarrow{z_0} = \{z_{0j}\}$ *belongs to cluster* g_0.

Proof. First, we define the weighted squared-distance between object z_0 and cluster g_0 representing a cluster with the centroid $C_{g_0} = \{c_{g_0 1}, \ldots, c_{g_0 m}\}$ as follow:

$$D_\alpha(\overrightarrow{z_0}, g_0) = \sum_{j=1}^{m} (z_{0j}\sqrt{w_j(\alpha)} - c_{g_0 j}\sqrt{w_j(\alpha)})^2 \tag{26}$$

Then define function F as the difference between two weighted squared-distances:

$$F_\alpha(\overrightarrow{z_0}, g_0, g_i) = D_\alpha(\overrightarrow{z_0}, g_0) - D_\alpha(\overrightarrow{z_0}, g_i) \tag{27}$$

In K-means clustering, the object is always assigned to the nearest cluster with the smallest distance to the cluster center. That is,

$$\begin{gathered} \overrightarrow{z_0} = \{z_{0j}\} \in g_0 \Leftrightarrow \\ D_\alpha(\overrightarrow{z_0}, g_0) < D_\alpha(\overrightarrow{z_0}, g_i)\ for\ \forall i \neq 0 \\ \Leftrightarrow F_\alpha(\overrightarrow{z_0}, g_0, g_i) < 0 \end{gathered} \tag{28}$$

Therefore, if $F_\alpha(\overrightarrow{z_0}, g_0, g_i)$ is viewed as a function of α: $F_{\overrightarrow{z_0}, g_0, g_i}(\alpha)$, then Proposition 2 is mathematically equivalent to the following statement:

$$\begin{gathered} F_{\overrightarrow{z_0}, g_0, g_i}(\alpha) < 0\ for\ all\ \alpha \in (g(t),\ g(t+1)], \\ if\ \exists\ \alpha \in (g(t),\ g(t+1)],\ s.t.\ F_{\overrightarrow{z_0}, g_0, g_i}(\alpha^*) < 0. \end{gathered} \tag{29}$$

Thus, we can prove equation (29) instead. First, we show that $F_{\overrightarrow{z_0}, g_0, g_i}(\alpha)$ is actually a Hyperbolic function of α with location parameter H_1 and scale parameter H_2.

$$\begin{aligned} F_{\overrightarrow{z_0}, g_0, g_i}(\alpha) &= D_\alpha(\overrightarrow{z_0}, g_0) - D_\alpha(\overrightarrow{z_0}, g_i) \\ &= \sum_{j=1}^{m} \left\{ (c_{g_i j}\sqrt{w_j(\alpha)} - c_{g_0 j}\sqrt{w_j(\alpha)})(2z_{0j}\sqrt{w_j} - c_{g_i j}\sqrt{w_j(\alpha)} - c_{g_0 j}\sqrt{w_j(\alpha)}) \right\} \\ &= \sum_{j=1}^{m} \{ w_j(\alpha)(c_{g_i j} - c_{g_0 j})(2z_{0j} - c_{g_i j} - c_{g_0 j}) \} \\ &= \sum_{j=1}^{t_{opt}} [\frac{m}{t_{opt}} + \frac{(\overline{\beta}_{t_{opt}} - \beta_j)(m-1)}{2\alpha}] \{ (c_{g_i j} - c_{g_0 j})(2z_{0j} - c_{g_i j} - c_{g_0 j}) \} \\ &= H_1 + \frac{H_2}{\alpha} \end{aligned} \tag{30}$$

where,

$$\begin{aligned} H_1 &== \frac{m}{t_{opt}} \sum_{j=1}^{t_{opt}} \{ (c_{g_i j} - c_{g_0 j})(2z_{0j} - c_{g_i j} - c_{g_0 j}) \}; \\ H_2 &== \frac{m-1}{2} \sum_{j=1}^{t_{opt}} \left\{ (\overline{\beta}_{t_{opt}} - \beta_j)(c_{g_i j} - c_{g_0 j})(2z_{0j} - c_{g_i j} - c_{g_0 j}) \right\}. \end{aligned} \tag{31}$$

Since hyperbolic function is always monotonic in each branch, we will utilize this feature to prove equation (29).The uniqueness assumption is utilized to prove Proposition 2. The proof includes two parts: we prove Proposition 2 when $H_2 > 0$ and when $H_2 < 0$ separately.

1. When $H_2 > 0$, according to monotonic feature of hyperbolic function, $F_{\overrightarrow{z_0},g_0,g_i}(\alpha)$ is strictly decreasing when $\alpha > 0$. To prove equation (29), we only need to show that:

$$F_{\overrightarrow{z_0},g_0,g_i}(\alpha_{min} = g(t)) < 0,\ if\ \exists\ \alpha^* \in (g(t),\ g(t+1)],\ s.t.\ F_{\overrightarrow{z_0},g_0,g_i}(\alpha^*) < 0. \tag{32}$$

   We proceed by contradiction and assume $\exists\ i \neq 0, F_{\overrightarrow{z_0},g_0,g_i}(\alpha_{min} = g(t)) > 0$, but $F_{\overrightarrow{z_0},g_0,g_i}(\alpha_*) < 0$.That means $\overrightarrow{z_0} = \{z_{0j}\} \notin C_{g_0}$ when $\alpha_{min} = g(t)$. So there must be another cluster partition $C\prime \neq C$ so that $F_{\overrightarrow{z_0},g_0',g_i'}(\alpha_{min} = g(t)) < 0,\ for\ \forall i \neq 0$. Then, since $F_{\overrightarrow{z_0},g_0',g_i'}(\alpha^*)$ is monotonically decreasing and $\alpha^* > \alpha_{min} = g(t),\ F_{\overrightarrow{z_0},g_0',g_i'}(\alpha^*) < 0,\ for\ \forall i \neq 0$. On the other hand, we have $F_{\overrightarrow{z_0},g_0',g_i'}(\alpha^*) < 0$ by assumption, which implies $\overrightarrow{z_0} \in C_{g_0'}$ and also $\overrightarrow{z_0} \in C_{g_0}$ at the same time. This contradicts the assumption that the cluster assignment is unique.

2. When $H_2 < 0$, $F_{\overrightarrow{z_0},g_0,g_i}(\alpha)$ is strictly decreasing when $\alpha > 0$. By following a similar argument as above, we can show that:

$$F_{\overrightarrow{z_0},g_0,g_i}(\alpha_{max} = g(t+1)) < 0,\ if\ \exists\ \alpha^* \in (g(t),\ g(t+1)],\ s.t.\ F_{\overrightarrow{z_0},g_0,g_i}(\alpha^*) < 0. \tag{33}$$

   This further implies the desired result that equation (29) is true.

□