

## An iterative algorithm for optimal variable weighting in K-means clustering

Shaonan Zhang, Shanshan Li, Jiaqiao Hu, Haipeng Xing & Wei Zhu

To cite this article: Shaonan Zhang, Shanshan Li, Jiaqiao Hu, Haipeng Xing & Wei Zhu (2018): An iterative algorithm for optimal variable weighting in K-means clustering, Communications in Statistics - Simulation and Computation, DOI: [10.1080/03610918.2017.1414244](https://doi.org/10.1080/03610918.2017.1414244)

To link to this article: <https://doi.org/10.1080/03610918.2017.1414244>



Published online: 17 Jan 2018.



Submit your article to this journal [↗](#)



Article views: 8



View related articles [↗](#)



View Crossmark data [↗](#)



# An iterative algorithm for optimal variable weighting in K-means clustering

Shaonan Zhang, Shanshan Li, Jiaqiao Hu, Haipeng Xing, and Wei Zhu

Department of Applied Mathematics and Statistics, Stony Brook University, NY, United States

## ABSTRACT

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 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 a 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.

## ARTICLE HISTORY

Received 13 March 2016  
Accepted 28 November 2017

## 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 (Hartigan and Wong 1979; Jain 2010). Assuming that we have  $n$  objects with  $m$  variables:  $x_i = (x_{i1}, x_{i2}, \dots, x_{im})$ ,  $i = 1, 2, \dots, n$ , 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:

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

where  $I_g$  is the set of objects belonging to cluster  $g = 1, 2, \dots, k$  and  $c_g = (c_{g1}, c_{g2}, \dots, c_{gm})$  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 (Wagstaff et al. 2001).

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 <sup>[5,6]</sup>

**CONTACT** Shanshan Li  [shaniavina@gmail.com](mailto:shaniavina@gmail.com)  Department of Applied Mathematics and Statistics, Stony Brook University, NY, United States.

Color versions of one or more of the figures in the article can be found online at [www.tandfonline.com/lssp](http://www.tandfonline.com/lssp).

© 2017 Taylor & Francis Group, LLC

is to scale each variable by the variable mean  $\bar{x}_{.j}$  and standard deviation  $s_j$ :

$$z_{ij} = \frac{x_{ij} - \bar{x}_{.j}}{s_j}. \quad (1)$$

However, this standardization approach is not unique and needs to be chosen carefully based on the underlying data structure (Kaufman and Rousseeuw 2009). 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 (1984) 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 Fayyad (1998), Kanungo et al. (2002), Huang et al. (2005), and Modha and Spangler (2003). 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  $w = (w_1, w_2, \dots, w_m)$  that minimize the sum of the WGSS with a penalty term as follows:

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

where  $\alpha$  is an additional parameter that penalizes the increased discrepancy among variable weights. When the penalty parameter  $\alpha$  is chosen large, the minimization of Equation (2) forces all weights to be close to 1, leading to small differences among variable weights, whereas small values of  $\alpha$  generally allow large differences among variable weights. The idea is to stabilize the variable weights by carefully choosing the penalty parameter  $\alpha$ . 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 (Lagarias et al. 1998) 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 (Karush 1939; Kuhn and Tucker 1951) 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 (2) 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  $\beta$  (within-cluster mean squares on each variable) in [Section 2.2](#), and propose a new data-driven approach to select the penalty parameter  $\alpha$  in [Section 2.3](#).

Note that by switching the order of the summations, the objective function (2) can be equivalently written as a function of variable weights  $(w_1, w_2, \dots, w_m)$ :

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

Assuming that the true cluster centroids  $c_g (g = 1, 2, \dots, k)$  are known, we denote the coefficient of  $w_j$  as  $\beta_j$ , 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  $\beta_1 \leq \beta_2 \leq \dots \leq \beta_m$ . For a given  $\alpha$ , the optimal variable weights can be obtained as the solution to the following quadratic optimization problem:

$$\begin{aligned} \text{Minimize : } f(w; \alpha) &= \sum_{j=1}^m \beta_j w_j + \alpha \sum_{j=1}^m \frac{(w_j - 1)^2}{m-1} \\ \text{Subject to : } \sum_{j=1}^m w_j &= m; \\ w_j &\geq 0, j = 1, 2, \dots, m. \end{aligned} \quad (4)$$

Thus, by applying the Lagrangian multiplier method, we define the lagrangian function as follows:

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

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

$$\frac{\partial L}{\partial w_j} = \beta_j + \frac{2\alpha}{m-1}(w_j - 1) + \lambda + \mu_j = 0, \quad j = 1, 2, \dots, m \quad (5)$$

$$\sum_{j=1}^m w_j - m = 0 \quad (6)$$

$$\mu_j w_j = 0, \quad j = 1, 2, \dots, m \quad (7)$$

$$w_j \geq 0, \quad j = 1, 2, \dots, m. \quad (8)$$

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

$$\begin{cases} w_j(\alpha, t_{opt}) = \frac{m}{t_{opt}} + \frac{(\bar{\beta}_{t_{opt}} - \beta_j)(m-1)}{2\alpha} & j \leq t_{opt} \\ w_j(\alpha, t_{opt}) = 0 & j > t_{opt}, \end{cases} \quad (9)$$

where  $t_{opt}$  is the optimal number of non-zero variable weights and  $\bar{\beta}_t = \frac{\sum_{i=1}^t \beta_i}{t}$ ; see [Appendix A](#) for detailed derivation steps.

## 2.1. An iterative algorithm

Equation (9) provides the closed-form expression for the optimal variable weights in k-means clustering when  $\beta \triangleq (\beta_1, \dots, \beta_m)$  (within-cluster mean squares) is known for all variables. However, the actual value of  $\beta$  is unknown unless the clustering partition is given. To address

this issue, we propose a recursive procedure to iteratively estimate  $\beta$ ,  $\alpha$ , and subsequently the optimal variable weights:

- Step 1. Standardize data matrix using Equation (1). Specify an initial  $\beta$  and the corresponding penalty parameter  $\alpha$ .
- Step 2. Given  $\beta$  and  $\alpha$ , calculate the optimal variable weights  $(w_1, w_2, \dots, w_m)$  according to (9).
- Step 3. Run k-means clustering on the weighted variable  $Z^* = Z * D$ , where  $Z$  is the data matrix whose element is  $z_{ij}$  and,  $D$  is a diagonal matrix with diagonal entries  $\text{Diag}(D) = (\sqrt{w_1}, \sqrt{w_2}, \dots, \sqrt{w_m})$ , and the operator  $*$  stands for matrix multiplication. Calculate within-cluster mean squares on each variable  $\beta_j$  and update penalty parameter  $\alpha$  accordingly.
- Step 4. Repeat steps 2 and 3 until the parameter vector  $\beta$  converges.

The choices of initial  $\beta$  values and the determination of the penalty parameter  $\alpha$  are discussed in detail in the following subsections.

## 2.2. Initial $\beta$ estimation

From Equation (3), we have the following linear relationship between the overall within cluster sums of squares on weighted variables  $Z^*$  and  $\beta$ :

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

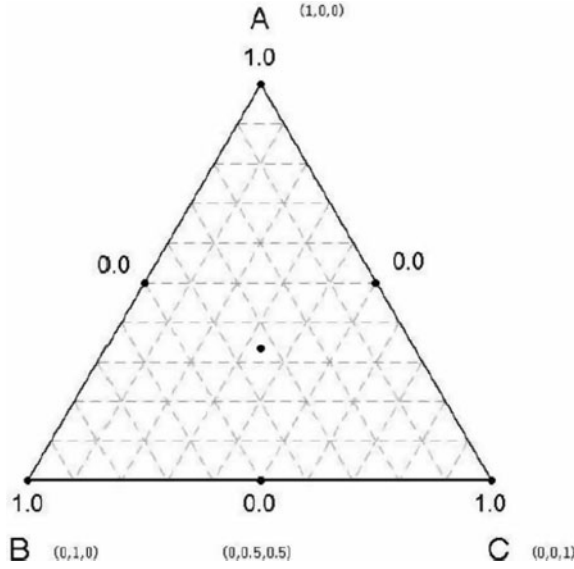
where  $z_{ij}^* = z_{ij} \sqrt{w_j}$ ,  $c_{gj}^* = c_{gj} \sqrt{w_j}$  are the re-scaled variables and the corresponding cluster centroids.

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

$$y = \sum_{j=1}^m \beta_j w_j + \varepsilon. \quad (10)$$

To estimate the initial  $\beta$ , we apply an  $\{m, 2\}$  simplex lattice design (Myers, Montgomery, and Anderson-Cook 2016) with a center point to generate initial variable weights and estimate  $\beta$  afterwards. Generally, an  $\{m, p\}$  simplex lattice design generates a set of  $m$ -dimensional points  $(x_1, x_2, \dots, x_m)$  such that each component can take  $p+1$  equally spaced values from 0 to 1, that is,  $x_i = 0, 1/p, 2/p, \dots, 1$ ; for  $i = 1, 2, \dots, m$  and the sum of all the components equal to 1. Graphically, it consists of all  $m$  vertices and  $p$ -equal-division-points on  $\binom{m}{2}$  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 the 3 edges.

In our algorithm, we generate an  $\{m, 2\}$  simplex lattice design to obtain a set of vectors  $p = (p_1, p_2, \dots, p_m)$ . Then for each design  $p$ , we run the k-means clustering on weighted variables with weight  $w = m * p$  and calculate the overall within cluster sum of squares and subsequently the response variable  $y$  in model (10). Finally, we fit the linear model and take the least square estimate as a starting value for  $\beta$ .



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

### 2.3. Selection of penalty parameter $\alpha$

In the optimal solution (9),  $\alpha$  serves as a penalty parameter for the heterogeneity in variable weighting and also as a tuning parameter to stabilize the optimal weights.

It can be shown that when  $\alpha$  stays in a certain range, the clustering partition remains the same. In fact, we prove in [Appendix B](#) that there is a unique vector  $g = \{g(t) \triangleq \frac{t(\beta_t - \bar{\beta}_t)(m-1)}{2m}, t = 1, 2, \dots, m\}$  that splits the range of  $\alpha$  into  $(m+1)$  intervals and the optimal clustering partition remains the same when  $\alpha$  changes within each interval. Therefore, the determination of  $\alpha$  is essentially equivalent to determining the value of  $t$  so that  $\alpha \in (g(t), g(t+1)]$  can be 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:

$$RV_i = \frac{1 - \beta_i}{\sum_{i=1}^m (1 - \beta_i)}, \quad \sum_{i=1}^m RV_i = 1.$$

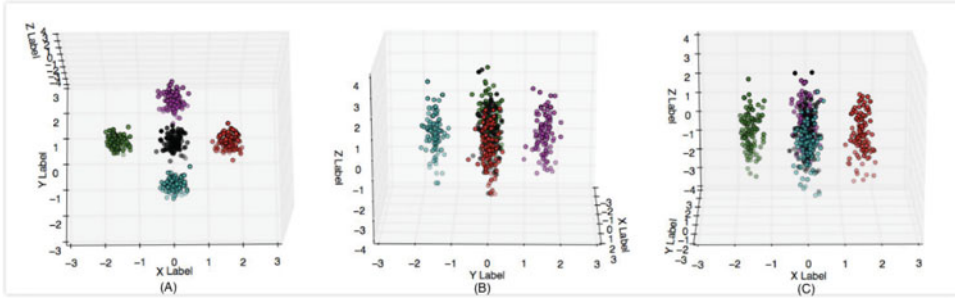
Then we will select

$$t_{\text{selected}} = \min \left\{ t \mid \sum_{i=1}^t RV_i > \frac{m-1}{m} \right\} \quad (11)$$

as an estimate of the optimal  $t$  value and therefore take the value of  $\alpha$  as

$$\alpha_{\text{selected}} = \frac{g(t_{\text{selected}}) + g(t_{\text{selected}} + 1)}{2}.$$

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  $\frac{m-1}{m}$  in (11) on cumulative RVs always shows stable performance in terms of removing noisy variables without losing too much information. In practice, such a threshold



**Figure 2.** Scatter plot of the first dataset.

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 seven sets of simulated data (Section 3.1) and four real datasets from 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 containing two informative variables and one noisy variable. Each group follows a 3-dimensional multivariate normal distribution,  $N(\mu, I_3)$ . The five group means are  $(5,0,0)$ ,  $(-5,0,0)$ ,  $(0,5,0)$ ,  $(0,-5,0)$ ,  $(0,0,0)$ . Figure 2 shows a pictorial description of the five groups in three dimensions. From the picture, we can easily see that two components are signals while the third one is noise.

For the first dataset, our algorithm takes only two iterations in less than 1 second. The results are reported in Table 1. From the table, we see that the components of  $\beta$  corresponding to the first two informative variables are very small while the component corresponding to the third noisy variable is almost 1, which is as expected. Also note that through (11), the algorithm correctly indicates  $t_{\text{selected}} = 2$ , which is identical to the true number of informative variables  $t_{\text{real}}$ . The values of  $\beta$  and  $\alpha$  obtained at the end of the iterations are then used in (9) to calculate the optimal variable weights, and the resulting cluster partition is given in Table 2. We see that all objects are correctly classified in this case.

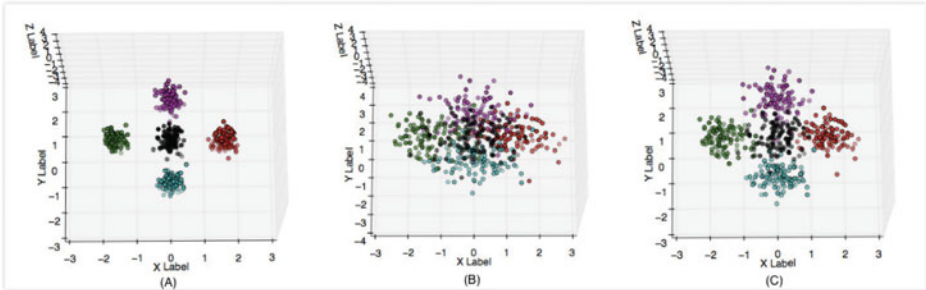
Simulated Case 2: From Figure 2(A), we notice that the centroids of observations in each group are quite scattered. So we reset the group means to  $(1,0,0)$ ,  $(-1,0,0)$ ,  $(0,1,0)$ ,  $(0,-1,0)$ ,  $(0,0,0)$  (Second group dataset; see Figure 3(B)) and  $(3,0,0)$ ,  $(-3,0,0)$ ,  $(0,3,0)$ ,  $(0,-3,0)$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

**Table 1.** Parameters of the first group dataset.

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



**Figure 3.** (A) Scatter plot of the first group dataset; (B) Scatter plot of the second group dataset; (C) Scatter plot of the third group dataset.

For the second and third group datasets, the estimated  $\beta$  and  $\alpha$  obtained in each iteration and the resulting partitions are reported in [Tables 3, 4](#) and [Tables 5, 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.

Simulated Case 3: Note that in Case 1, the numbers of observations are the same across different groups. So we have considered two other group datasets. In the fourth group dataset (shown in [Figure 4\(B\)](#)), we change the number of observations in each respective group to

**Table 3.** Parameters of the second group dataset.

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

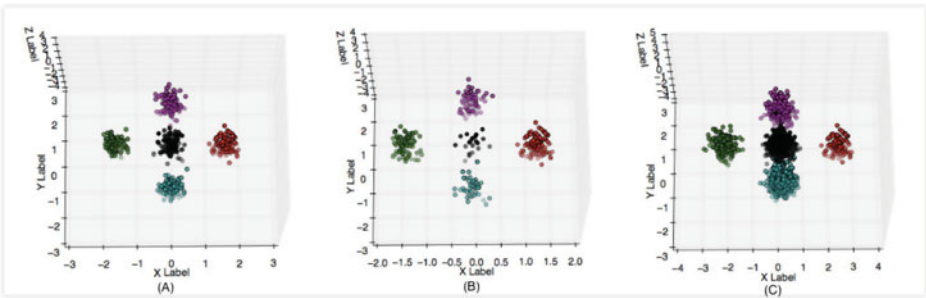


**Table 5.** Parameters of the third group dataset.

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



**Figure 4.** (A) Scatter plot of the first group dataset; (B) Scatter plot of the fourth group dataset; (C) Scatter plot of the fifth group dataset.

(100, 80, 60, 40, 20); whereas in the fifth dataset (Figure 4(C)), we increase the sample sizes of different groups to (100, 200, 300, 400, 500).

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

**Table 7.** Parameters of the fourth group dataset.

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

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

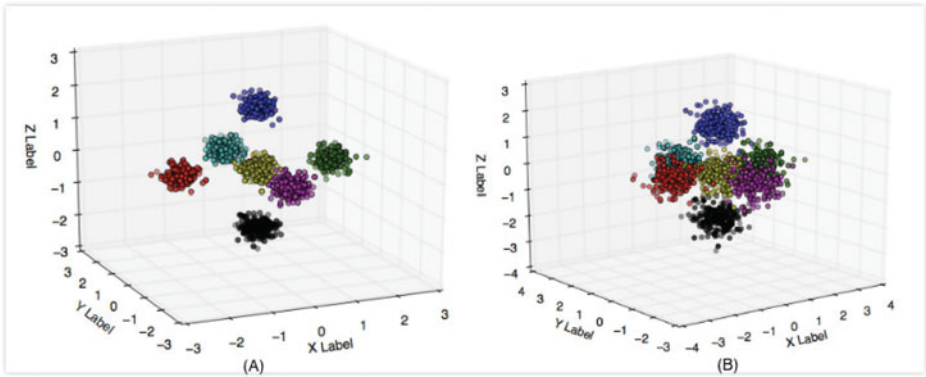


Figure 5. (A) Scatter plot of the sixth group dataset; (B) Scatter plot of the seventh group dataset.

Simulated Case 4: To test the performance of our method on high-dimensional large dataset, we created 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 have 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)).

For the sixth and seventh datasets, the estimated parameters in each iteration and partition results are given in Tables 11, 12 and Tables 13, 14. We observe that in both cases, although the initial values of  $t_{selected}$  are quite different from the true numbers of informative variables  $t_{real}$ , our algorithm is able to quickly identify its correct values after a single iteration.

To further test the robustness of the algorithm, we have repeated our experiments 100 times on each of the seven datasets considered (each based on an independently generated dataset). The performance of the algorithm, averaged over the 100 runs, are recorded in Table 15, where each entry in the table represents the percentage of correct classification in each of the respective cases.

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

Table 11. Parameters of the sixth group dataset.

	$\beta$			$t_{real}$	$t_{selected}$	$\alpha_{selected}$	variable weights
Initial	0.0614, 0.0638, 0.064, 0.1842, 0.1848, 0.1866, 0.1871, 0.1923			3	7	0.732	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	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	3	0.6318	2.6740, 2.6690, 2.6570, 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	Cluster 7
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

### 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, the 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 16 and 17. We see that only 6 instances are mistakenly clustered.

**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 18.

**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 18.

**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 18.

### 3.3. Numerical comparison

In this section, we compare the performance of our method to that of Huh and Lim on the first and sixth group datasets in Section 3.1 and the iris data in Section 3.2. To allow for a fair comparison of both algorithms, 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 used by Huh and Lim is primarily designed for unconstrained optimization and is not known to be globally convergent. Consequently, it is likely that their approach will lead to non-optimal clustering partitions. 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  $\alpha$  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  $\alpha$  with stable variable weighting. If the algorithm is not stable, it will be very difficult to find the feasible range.

Table 13. Parameters of the seventh group dataset.

	$\beta$			$t_{real}$	$t_{selected}$	$\alpha_{selected}$	variable weights
Initial	0.1178, 0.1206, 0.1221, 0.1900, 0.1903, 0.1923, 0.1928, 0.1980			3	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	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	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 6	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

**Table 15.** Averaged correct classification rate over 100 simulation runs (Unit: %).

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7
1st Dataset	99.34	99.40	99.24	99.32	98.93	N/A	N/A
2nd Dataset	75.44	75.01	76.61	74.00	44.71	N/A	N/A
3rd Dataset	91.01	91.62	91.20	91.01	76.01	N/A	N/A
4th Dataset	99.03	99.26	98.98	99.33	97.60	N/A	N/A
5th Dataset	99.45	99.38	99.34	99.35	97.37	N/A	N/A
6th Dataset	99.42	99.35	99.64	99.56	98.77	98.48	97.71
7th Dataset	99.45	99.47	99.60	99.56	97.75	98.12	95.59

**Table 16.** Parameters of the eighth group dataset.

	$\beta$	$t_{selected}$	$\alpha_{selected}$	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

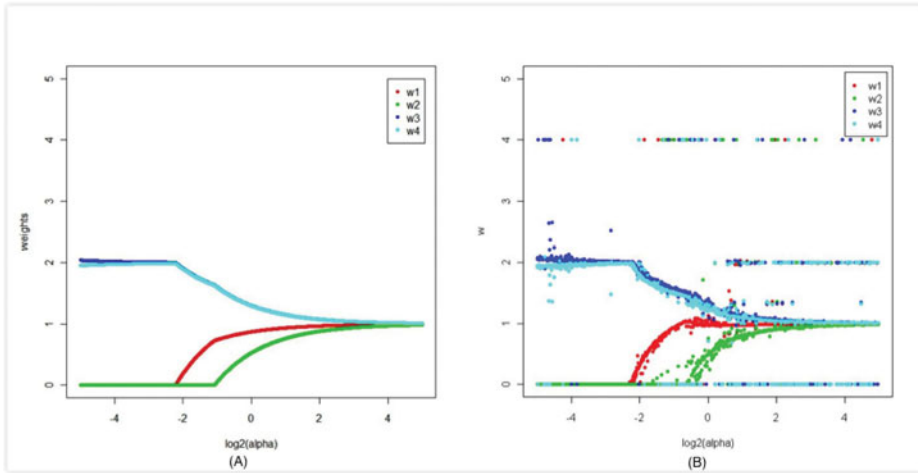
Recall from Equation (2), when  $\alpha$  increases from 0 to 1, the penalty part is emphasized and therefore all weights in the objective function will gradually move towards 1. In the following figures (See Figure 6–8), we see that our method captures this movement very nicely in all three datasets with all weights moving slowly towards 1. However, for the method in Huh and Lim, this behavior 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  $\alpha$  selection approach based on the weighting curve is not reliable.

**Table 17.** 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

**Table 18.** Comparison results of the real datasets.

	Instances/ Observations	Variables/ Features	Iteration	Optimal Variable Weights	MR
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	2.3422,2.3423,1.3733,1.0040,0.9382,0,0,0	602/1484
CMC Data	1473	9	2	4.4570,4.2607,0.2637,0.0187,0,0,0,0,0	321/1473

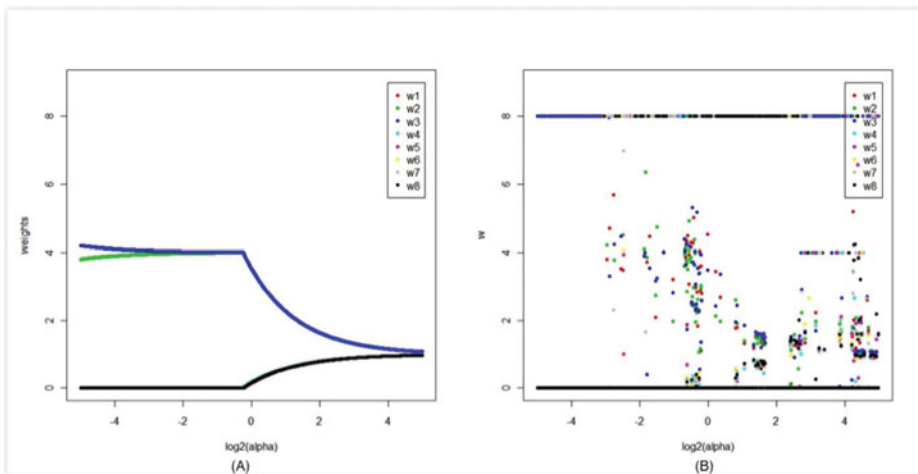


**Figure 6.** (A) Weighting curve of the first group dataset using our method; (B) Weighting curve of the First group dataset using Huh and Lim's method.

### 3.3.2. Clustering accuracy

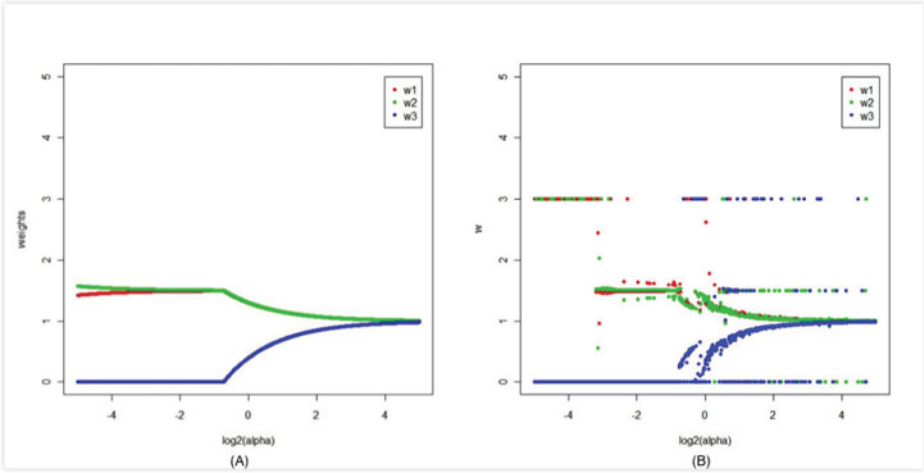
To investigate the clustering accuracy, we compare the misclassification rate of both methods. For each dataset, first, each cluster is identified as the group of majority members in the cluster. Then we count the number of misclassified cluster members in all clusters and divide it by the total number of observations to get the misclassification rate (MR). The results, together with the corresponding variable weights, are shown in Table 19. The optimal penalty parameter  $\alpha$  is determined using our approach (12) and the same  $\alpha$  is used for both methods (which is different from the one used in Huh and Lim for Iris data) in the comparison.

We see that in the first group dataset, both methods show comparable performance. However, on the other two datasets, our method finds different variable weighting than the original method of Huh and Lim, which results in better clustering partitions and lower misclassification rates. 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



**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.

**Table 19.** Comparison results of our method and Huh and 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

fails in the sixth group of simulated data. This result is somewhat as expected, since on low-dimensional data, the Nelder-Mead method performs relatively well, whereas as the problem dimension increases, the Nelder-Mead Simplex method may perform poorly and fail to find the global optimal solution for the constrained optimization problem.

4. Conclusion

It is well understood that the K-means clustering algorithm is ideal for detecting clusters that are homogenous, spherical, and without the presence of noise variables. The weighted K-means algorithm can lead to improved 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 approach 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  $\alpha$  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 potentially be utilized on high dimensional datasets,

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. Thus, an important future work would be to examine and validate our method on high dimensional real datasets.

## Appendix A. 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  $w_i^*$ ,  $i = 1, 2, \dots, m$  be the optimal solutions to Equation (4). 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

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

Therefore, by switching the order of  $w_i^*$  and  $w_j^*$ , one can construct another set of weights  $\bar{w}$  such that  $f(\bar{w}; \alpha) < f(w_i^*, \alpha)$ , which contradicts the fact that  $w_i^*$  is the optimal solution to Equation (4).

Now we start to solve the system of Equations (5)–(8) derived from the KKT conditions. First, from Equation (7), we know for each  $j$ , either  $\mu_j$  or  $w_j$  must be zero. Assuming there are  $t$  variables that have nonzero weights, based on inequality (8) and Proposition 1, we have

$$\begin{aligned} w_1 &\geq w_2 \geq \dots \geq w_t > 0 = w_{t+1} = \dots = w_m \\ \mu_1 &= \mu_2 = \dots = \mu_t = 0. \end{aligned} \quad (12)$$

By substituting the above into Equations (5) and (6), we can obtain the  $t$  solutions with nonzero weights

$$\begin{cases} w_j(\alpha, t) = \frac{m}{t} + \frac{(\bar{\beta}_t - \beta_j)(m-1)}{2\alpha} & j \leq t \\ w_j(\alpha, t) = 0 & j > t, \end{cases} \quad (13)$$

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

In order to find the optimal variable weights, we need to decide the number of nonzero variable weights  $t$ . From Equation (13), we obtain

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

In the following, we denote

$$g(t) = \frac{t(\beta_t - \bar{\beta}_t)(m-1)}{2m}. \quad (14)$$

Since  $\beta_1 \leq \beta_2 \leq \dots \leq \beta_m$ , it is obvious that  $g(t)$  is a monotone function. Therefore, for a given  $\alpha \in (0, g(m)]$ , the set of all possible values of  $t$  satisfying Equation (13) is given by

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

In view of Equation (14), problem (1) can be re-formulated in terms of  $t$ , whose optimal solution can be obtained by solving the following optimization problem:

$$t_{opt} = \operatorname{argmin}_{t \in T(\alpha)} f(w; \alpha).$$

Finally, replacing  $t$  by  $t_{opt}$  in Equation (14), we obtain the optimal variable weights

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

□

## Appendix B. 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  $g(t) = \frac{t(\beta_t - \bar{\beta}_t)(m-1)}{2m}$ ,  $t = 1, 2, \dots, m$ .

**Proposition 2.** *Assume that the cluster assignment is unique. If  $\exists \alpha^* \in (g(t), g(t+1)]$  so that  $\vec{z}_0 = \{z_{0j}\}$  belongs to cluster  $g_0$ , then for all  $\alpha \in (g(t), g(t+1)]$ ,  $\vec{z}_0 = \{z_{0j}\}$  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  $C_{g_0} = \{c_{g_01}, \dots, c_{g_0m}\}$  as follow:

$$D_\alpha(\vec{z}_0, g_0) = \sum_{j=1}^m \left( z_{0j} \sqrt{w_j(\alpha)} - c_{g_0j} \sqrt{w_j(\alpha)} \right)^2.$$

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

$$F_\alpha(\vec{z}_0, g_0, g_i) = D_\alpha(\vec{z}_0, g_0) - D_\alpha(\vec{z}_0, g_i).$$

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

$$\vec{z}_0 = \{z_{0j}\} \in g_0 \Leftrightarrow D_\alpha(\vec{z}_0, g_0) < D_\alpha(\vec{z}_0, g_i) \text{ for } \forall i \neq 0 \Leftrightarrow F_\alpha(\vec{z}_0, g_0, g_i) < 0.$$

Therefore, we can view  $F_\alpha(\vec{z}_0, g_0, g_i)$  as a function of  $\alpha$  and write it as  $F_{\vec{z}_0, g_0, g_i}(\alpha)$ . Proposition 2 is then mathematically equivalent to the following statement:

$$\begin{aligned} & \text{If } \exists \alpha^* \in (g(t), g(t+1)], \text{ s.t. } F_{\vec{z}_0, g_0, g_i}(\alpha^*) < 0, \\ & \text{then } F_{\vec{z}_0, g_0, g_i}(\alpha) < 0 \text{ for all } \alpha \in (g(t), g(t+1)]. \end{aligned} \quad (\text{B.1})$$

Thus, we can prove Equation (B.1) instead. First, we show that  $F_{\vec{z}_0, g_0, g_i}(\alpha)$  is actually a Hyperbolic function of  $\alpha$  with location parameter  $H_1$  and scale parameter  $H_2$ .

$$\begin{aligned} F_{\vec{z}_0, g_0, g_i}(\alpha) &= D_\alpha(\vec{z}_0, g_0) - D_\alpha(\vec{z}_0, g_i) \\ &= \sum_{j=1}^m \left\{ \left( c_{g_i j} \sqrt{w_j(\alpha)} - c_{g_0 j} \sqrt{w_j(\alpha)} \right) \left( 2z_{0j} \sqrt{w_j(\alpha)} - c_{g_i j} \sqrt{w_j(\alpha)} - c_{g_0 j} \sqrt{w_j(\alpha)} \right) \right\} \\ &= \sum_{j=1}^m \left\{ w_j(\alpha) (c_{g_i j} - c_{g_0 j}) (2z_{0j} - c_{g_i j} - c_{g_0 j}) \right\} \end{aligned}$$

$$\begin{aligned}
&= \sum_{j=1}^{t_{opt}} \left[ \frac{m}{t_{opt}} + \frac{(\bar{\beta}_{t_{opt}} - \beta_j)(m-1)}{2\alpha} \right] \{ (c_{g_{ij}} - c_{g_{0j}})(2z_{0j} - c_{g_{ij}} - c_{g_{0j}}) \} \\
&= H_1 + \frac{H_2}{\alpha},
\end{aligned}$$

where we have defined

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

Since the hyperbolic function is always monotonic in each branch, we will utilize this feature to prove Equation (B.1). The proof consists of two parts:

(1)  $H_2 > 0$ ; and (2)  $H_2 < 0$ .

(1) When  $H_2 > 0$ , according to monotonic feature of hyperbolic function,  $F_{\vec{z}_0, g_0, g_i}(\alpha)$  is strictly decreasing when  $\alpha > 0$ . Select a constant  $\epsilon \in (0, \alpha^* - g(t))$  and define  $\alpha_1 = g(t) + \epsilon$ . Clearly, since  $\alpha^* \in (g(t), g(t+1)]$ , we have  $g(t) < \alpha_1 < \alpha^*$ . Because  $\epsilon$  is arbitrary, the rest of the proof amounts to showing that

$$\text{If } \exists \alpha^* \in (g(t), g(t+1)] \text{ such that } F_{\vec{z}_0, g_0, g_i}(\alpha^*) < 0, \text{ then } F_{\vec{z}_0, g_0, g_i}(\alpha_1) < 0.$$

We proceed by contradiction and assume  $\exists i \neq 0$  such that  $F_{\vec{z}_0, g_0, g_i}(\alpha_1) > 0$ . This implies that  $\vec{z}_0 = \{z_{0j}\} \notin g_0$  and there exists a different cluster partition so that  $F_{\vec{z}_0, g'_0, g'_i}(\alpha_1) < 0$  for some clusters  $g'_0 \neq g_0$  and  $g'_i, i \neq 0$ . In addition, since  $F_{\vec{z}_0, g'_0, g'_i}(\alpha^*)$  is monotonically decreasing and  $\alpha^* > \alpha_1$ , it follows that  $F_{\vec{z}_0, g'_0, g'_i}(\alpha^*) < 0$ , for all  $i \neq 0$ . However, this, together with the condition  $F_{\vec{z}_0, g_0, g_i}(\alpha^*) < 0$ , implies that  $\vec{z}_0 \in g'_0$  and  $\vec{z}_0 \in g_0$  at the same time, which contradicts the assumption that the cluster assignment is unique.

(2) When  $H_2 < 0$ ,  $F_{\vec{z}_0, g_0, g_i}(\alpha)$  is strictly increasing when  $\alpha > 0$ . By following a similar argument as above, we can show that

$$\text{If } \exists \alpha^* \in (g(t), g(t+1)], \text{ s.t. } F_{\vec{z}_0, g_0, g_i}(\alpha^*) < 0, \text{ then } F_{\vec{z}_0, g_0, g_i}(g(t+1)) < 0.$$

Hence the proof is completed by combined the above two cases.  $\square$

## References

- Bradley, P. S., and U. M. Fayyad. 1998. Synthesized clustering: A method for amalgamating alternative clustering bases with differential weighting of variables. *ICML*, 98.
- DeSarbo, W. S., Carroll, J. D., Clark, L. A. and Green, P. E. 1984. Synthesized clustering: A method for amalgamating alternative clustering bases with differential weighting of variables. *Psychometrika*. 49:187–215.
- Huang, J. Z. et al. 2005. Automated variable weighting in k-means type clustering. *Automated Variable Weighting in k-means Type Clustering* 27 (5):657–68.
- Huh, M.-H., and Y. B. Lim. 2009. Weighting variables in K-means clustering. *Journal of Applied Statistics* 36 (1):67–78.
- Kanungo, T. et al. 2002. An efficient k-means clustering algorithm: Analysis and implementation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24 (7):881–92.

- Karush, W. 1939. *Minima of Functions of Several Variables with Inequalities as Side Conditions*, Master's Thesis, Department of Mathematics, University of Chicago.
- Kuhn, H. W., and A. W. Tucker. 1951. Nonlinear programming, in *2nd Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley: University of California Press, pp. 481–92.
- Modha, D. S., and W. S. Spangler. 2003. Feature weighting in k-means clustering. *Machine Learning* 52 (3):217–37.
- Wagstaff, K., Cardie, C., Rogers, S., and Schrödl, S. 2001. *Constrained k-means clustering with background knowledge*. In *ICML*, Vol. 1, pp. 577–584.