Robustification of the sparse K-means clustering algorithm

Simulation studies and data analyses

Yumi Kondo, Matías Salibián-Barrera and Ruben H. Zamar

December 12, 2011

Table of contents

Introduction

Clustering

Searching for groups of ‘similar’ objects.

- The current applications of clustering include a large number of variables. Typically, only relatively small number of variables are important to determine the class memberships of the objects.
- Furthermore, large datasets may contain outliers.

Outliers

Objects that do not belong to any of the given clusters.

- We may wish to use a wise algorithm that classifies objects and identifies the important variables and outliers.

Notation for a dataset $\boldsymbol{X}$

Let $\boldsymbol{X} \in \mathbb{R}^{N \times p}$ denote the data matrix with N cases and p features and let $\boldsymbol{x}_i \in \mathbb{R}^p$ denote the i^{th} row of $\boldsymbol{X}$ ($i = 1, \cdots, N$). Moreover, let $\boldsymbol{X}_j \in \mathbb{R}^N$ denote the j^{th} column of $\boldsymbol{X}$ ($j = 1, \cdots, p$). That is:

$$\boldsymbol{X} = [\boldsymbol{X}_1 \cdots \boldsymbol{X}_p] = \begin{bmatrix} \boldsymbol{x}_1^T \\ \vdots \\ \boldsymbol{x}_N^T \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & & \vdots \\ x_{N1} & \cdots & x_{Np} \end{bmatrix}.$$

The goal of clustering is to obtain a partition $\mathcal{C} = \{C_1, \cdots, C_K\}$, that contains K sets of similar cases.

Dissimilarity measure

It measures how different two cases, $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i'}$, are.

- The Euclidean distance:

$$||\boldsymbol{x}_i - \boldsymbol{x}_{i'}|| = \sqrt{\sum_{j=1}^{p}(x_{ij} - x_{i'j})^2}.$$

- The squared Euclidean distance:

$$||\boldsymbol{x}_i - \boldsymbol{x}_{i'}||^2.$$

- Dissimilarity measures based on the Pearson's correlation
- We will focus on the squared Euclidean distance for now.

K-means

Idea

Given a number of clusters K, find the partition which minimizes the sum of dissimilarities within a cluster (i.e. within sum of squares).

$$WSS(\mathcal{C}) = \sum_{k=1}^{K} \frac{1}{2n_k} \sum_{i,i' \in C_k} ||\boldsymbol{x}_i - \boldsymbol{x}_{i'}||^2$$
$$= \sum_{k=1}^{K} \sum_{i \in C_k} ||\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k||^2.$$

where n_k is the number of cases within the k^{th} cluster :$n_k = |C_k|$.
The goal of K-means is to find:

$$\mathcal{C}^* = \underset{\mathcal{C};\ |\mathcal{C}|=K}{\operatorname{argmin}}\ WSS(\mathcal{C}).$$

The performance of 2-means on a simulated dataset 1

- $N = 40$
- $p = 2$: 2 clustering features and no noise features

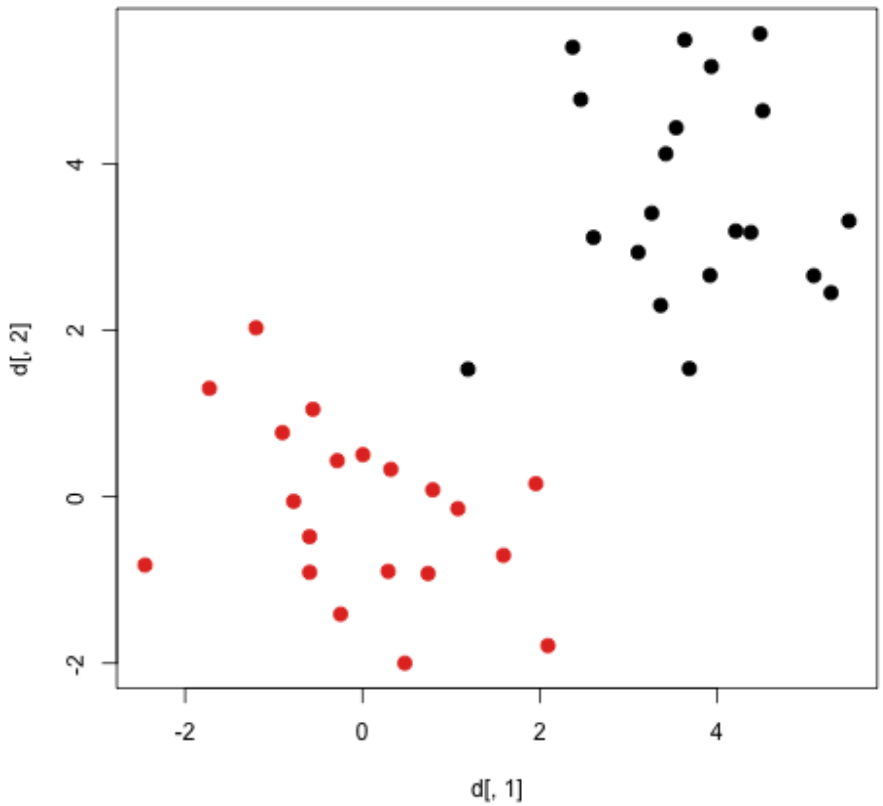


Figure: True cluster labels

The performance of 2-means on a simulated dataset 1

- $N = 40$
- $p = 2$: 2 clustering features and no noise features

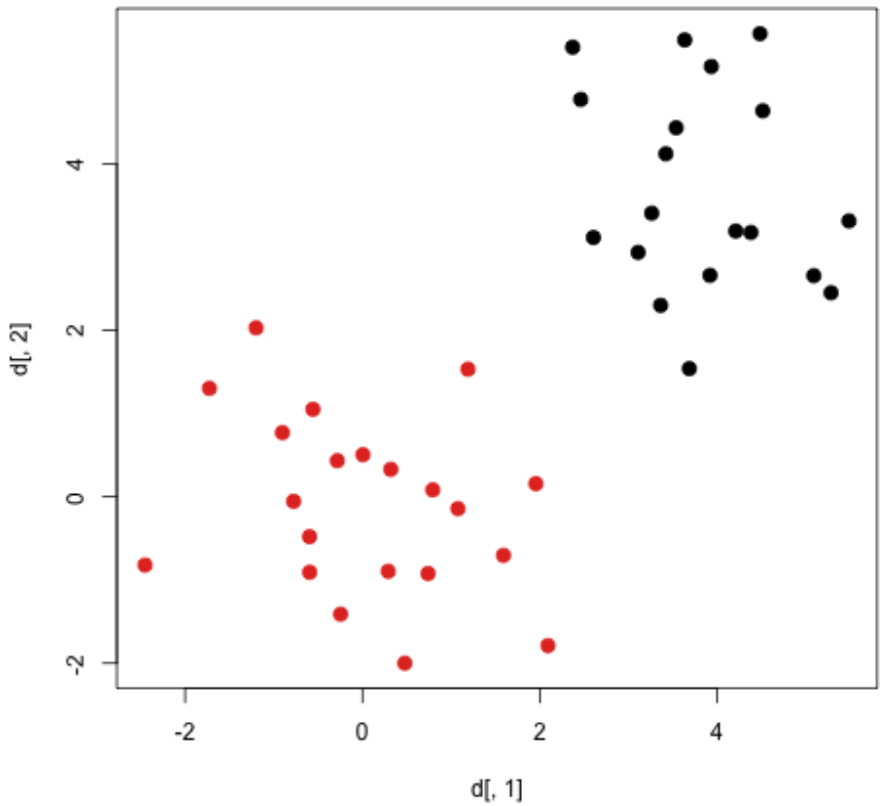


Figure: The partition from 2-means

Clustering features and noise features

- The clustering features are defined as features whose mean vary over clusters.
- If features are not clustering features then they are noise features.

How well 2-means perform if we added 998 noise features to the previous simulated dataset...?

(i.e. the dimension of the dataset increases from $p = 2$ to $p = 1000$!)

The performance of 2-means on a simulated dataset 2

- $N = 40$
- $p = 1000$: 2 clustering features and 998 noise features.
- All the noise features are independent standard normal variables.

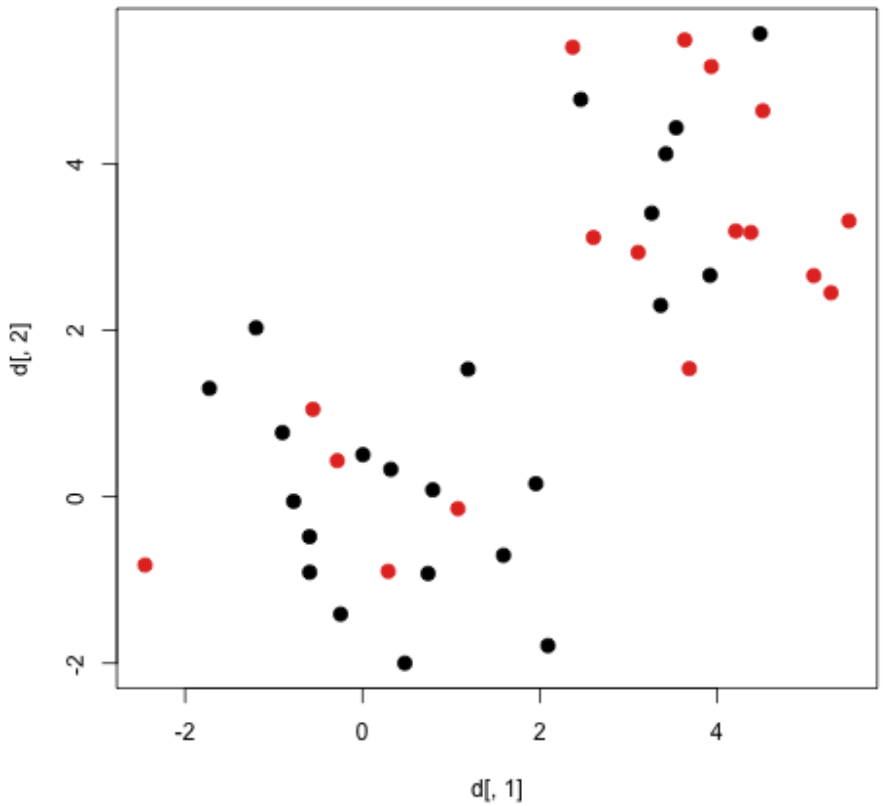


Figure: The partition from 2-means

Sparse K-means

Witten and Tibshirani (2010) proposed the sparse K-means algorithm which clusters the cases using an adaptively chosen subset of the features.

The sparse K-means algorithm views the problem of K-means as a maximization problem.

K-means as a maximization problem

The goal of K-means is to find:

$$\mathcal{C}^* = \underset{\mathcal{C};\ |\mathcal{C}|=K}{\text{argmin}}\ WSS(\mathcal{C}).$$

Define the total cluster sum of square to be:

$$TSS = \frac{1}{2N}\sum_{i=1}^{N}\sum_{i'=1}^{N}||\boldsymbol{x}_i - \boldsymbol{x}_{i'}||^2$$

$$= \sum_{i=1}^{N}||\boldsymbol{x}_i - \bar{\boldsymbol{x}}||^2$$

Define the between cluster sum of squares to be:

$$BSS(\mathcal{C}) = TSS - WSS(\mathcal{C})$$

$$= \sum_{k=1}^{K} n_k||\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{x}}||^2.$$

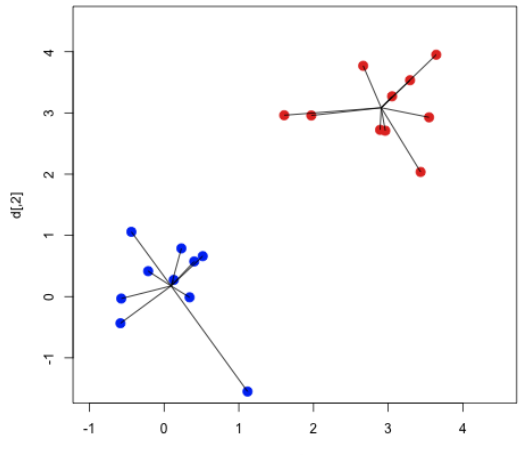


Figure: $WSS(\mathcal{C})$

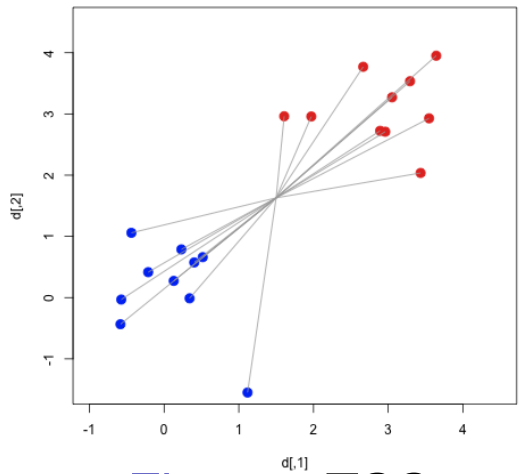


Figure: TSS

K-means as a maximization problem

The goal of K-means is to find:

$$\begin{aligned} \mathcal{C}^* &= \underset{\mathcal{C};\ |\mathcal{C}|=K}{\text{argmin}}\ WSS(\mathcal{C}) \\ &= \underset{\mathcal{C};\ |\mathcal{C}|=K}{\text{argmin}}\ TSS - BSS(\mathcal{C}) \\ &= \underset{\mathcal{C};\ |\mathcal{C}|=K}{\text{argmax}}\ BSS(\mathcal{C}). \end{aligned}$$

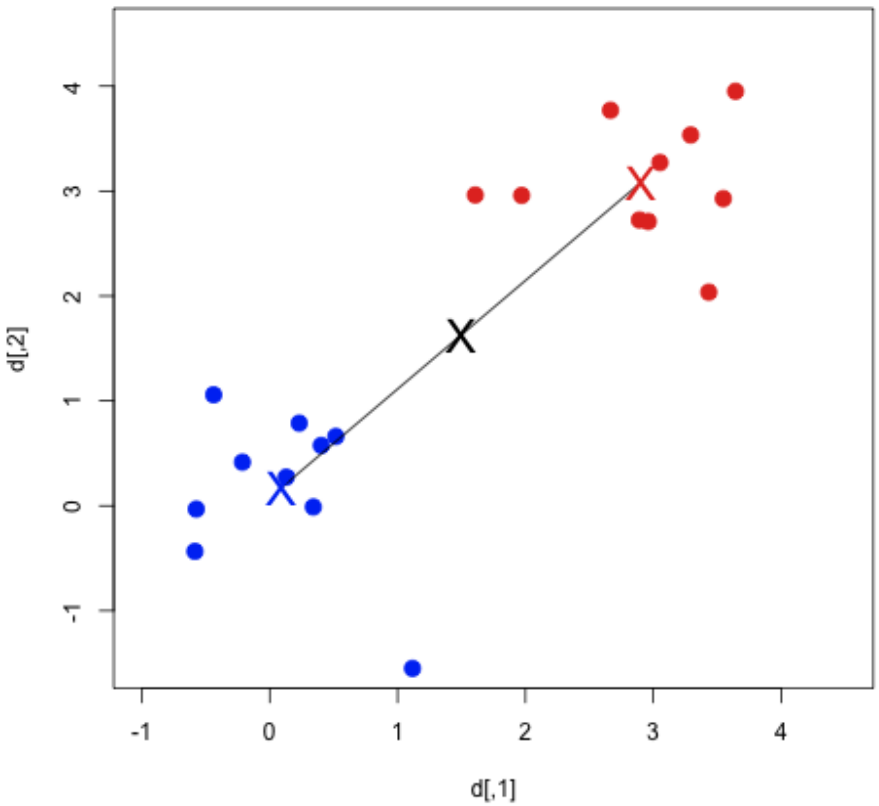


Figure: TSS and $WSS(\mathcal{C})$

The idea of sparse K-means clustering

- The between cluster sum of squares can be decomposed by each feature:

$$BSS = \sum_{j=1}^{p} BSS_j \quad \text{where} \quad BSS_j = \sum_{k=1}^{K} n_k (\bar{x}_{k,j} - \bar{x}_j)^2.$$

Idea

Between cluster sum of squares along clustering features would be larger than that of noise features.

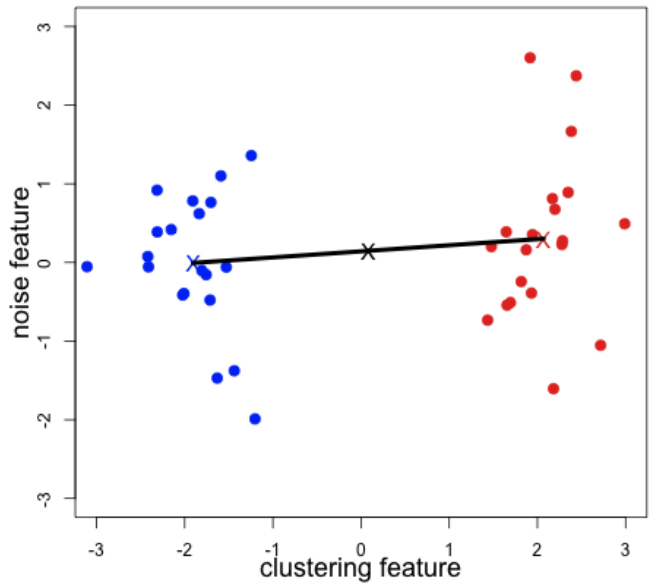


Figure: BSS along the noise feature is 0.95 while BSS along the clustering feature is 157.94.

The idea of sparse K-means clustering

- The between cluster sum of squares can be decomposed by each feature:

$$BSS = \sum_{j=1}^{p} BSS_j. \quad \text{where:} \quad BSS_j = \sum_{k=1}^{K} n_k (\bar{x}_{k,j} - \bar{x}_j)^2.$$

Idea

Between cluster sum of squares along clustering features would be larger than that of noise features.

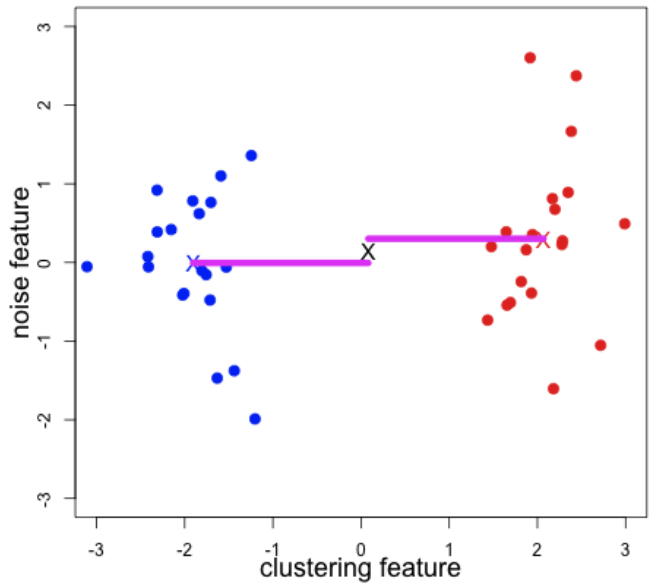


Figure: BSS along the noise feature is 0.95 while BSS along the clustering feature is 157.94.

The idea of sparse K-means clustering

- The between cluster sum of squares can be decomposed by each feature:

$$BSS = \sum_{j=1}^{p} BSS_j. \quad \text{where:} \quad BSS_j = \sum_{k=1}^{K} n_k(\bar{x}_{k,j} - \bar{x}_j)^2.$$

Idea

Between cluster sum of squares along clustering features would be larger than that of noise features.

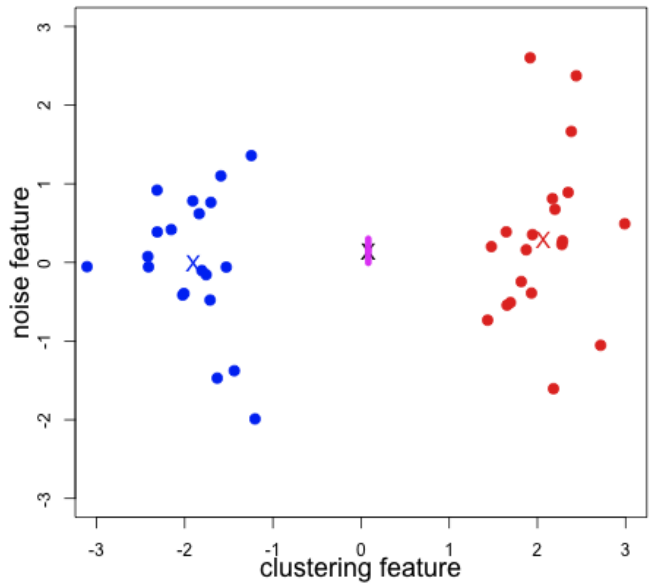


Figure: BSS along the noise feature i 0.95 while BSS along the clustering feature is 157.94.

Sparse K-means

Sparse K-means is defined as the solution to the problem:

$$\max_{\mathcal{C},\boldsymbol{w};\ |\mathcal{C}|=K} \sum_{j=1}^{p} w_j BSS_j(\mathcal{C}) \ \ s.t.\ \ ||\boldsymbol{w}||_1 \leq l_1, ||\boldsymbol{w}||^2 \leq 1, w_j \geq 0 \ \ \forall j, \qquad (1)$$

where $\boldsymbol{w} = [w_1, \cdots, w_p]^T$.

- The L_1 penalty on $\boldsymbol{w}$ results in sparsity for small values of the tuning parameter l_1.
- The L_2 penalty also serves an important role, since without it, only one element of $\boldsymbol{w}$ would be nonzero in general.

Sparse K-means algorithm

Initially let $\boldsymbol{w} = [1/\sqrt{p}, .., 1/\sqrt{p}]^T$ then repeat steps (a) and (b) below until convergence.

- Step (a): For a given $\boldsymbol{w}$, maximize (1) with respect to $\mathcal{C}$. That is, perform K-means on the transformed dataset $\boldsymbol{Y}=[\sqrt{w_1}\boldsymbol{X}_1 \cdots \sqrt{w_p}\boldsymbol{X}_p] \in \mathbb{R}^{N\times p}$:

$$\begin{aligned}\underset{\mathcal{C};\,|\mathcal{C}|=K}{\text{argmax}} \sum_{j=1}^{p} w_j BSS_j(\mathcal{C}) &= \underset{\mathcal{C};\,|\mathcal{C}|=K}{\text{argmin}} \sum_{j=1}^{p} w_j WSS_j(\mathcal{C}) \\ &= \underset{\mathcal{C};\,|\mathcal{C}|=K}{\text{argmin}}\ WSS_{\boldsymbol{Y}}(\mathcal{C})\end{aligned}$$

Sparse K-means algorithm

Step (b): For a given $\mathcal{C}$, maximize (1) with respect to $\boldsymbol{w}$:

$$\max_{\boldsymbol{w}} \boldsymbol{w}^T \boldsymbol{D} \quad s.t. \;\; ||\boldsymbol{w}||^2 \leq 1, ||\boldsymbol{w}||_1 \leq l_1, w_j \geq 0 \;\; \forall j,$$

where $\boldsymbol{D} = [BSS_1(\mathcal{C}), \cdots, BSS_p(\mathcal{C})]^T$. The analytic solution for this optimization problem is:

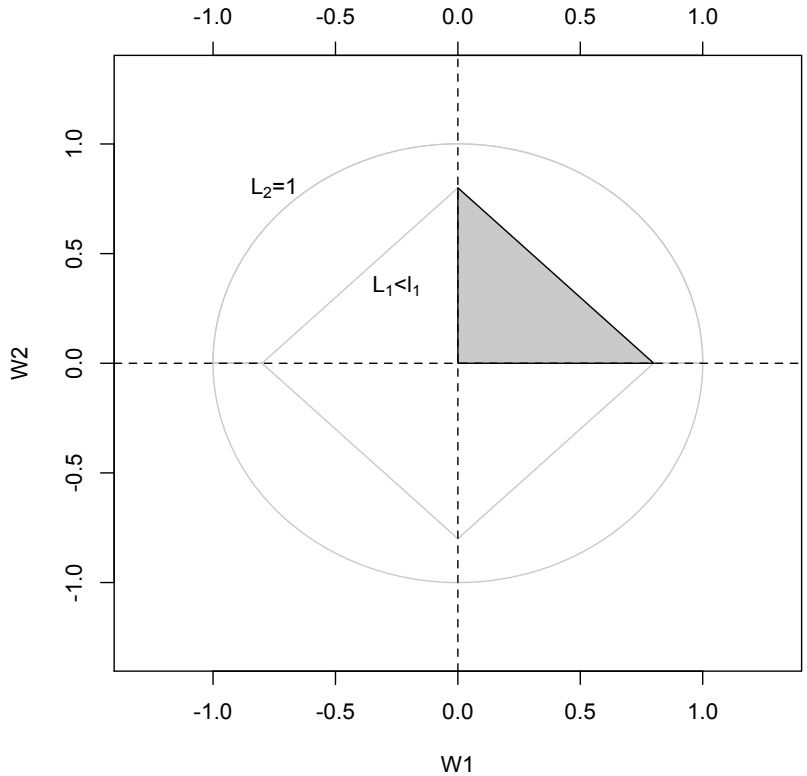
$$\boldsymbol{w} = \boldsymbol{w}(\triangle) = \frac{(\boldsymbol{D} - \triangle \mathbf{1})_+}{||(\boldsymbol{D} - \triangle \mathbf{1})_+||_2},$$

where:

$$\triangle^* = \begin{cases} 0 & \text{if } ||\boldsymbol{w}(0)||_1 \leq l_1, \\ \text{root of } \; ||\boldsymbol{w}(\triangle)||_1 - l_1 = 0 & \text{otherwise} \end{cases}.$$

The $L1$ tuning parameter value

The admissible regions for $\boldsymbol{w}$ in $\mathbb{R}^2$.

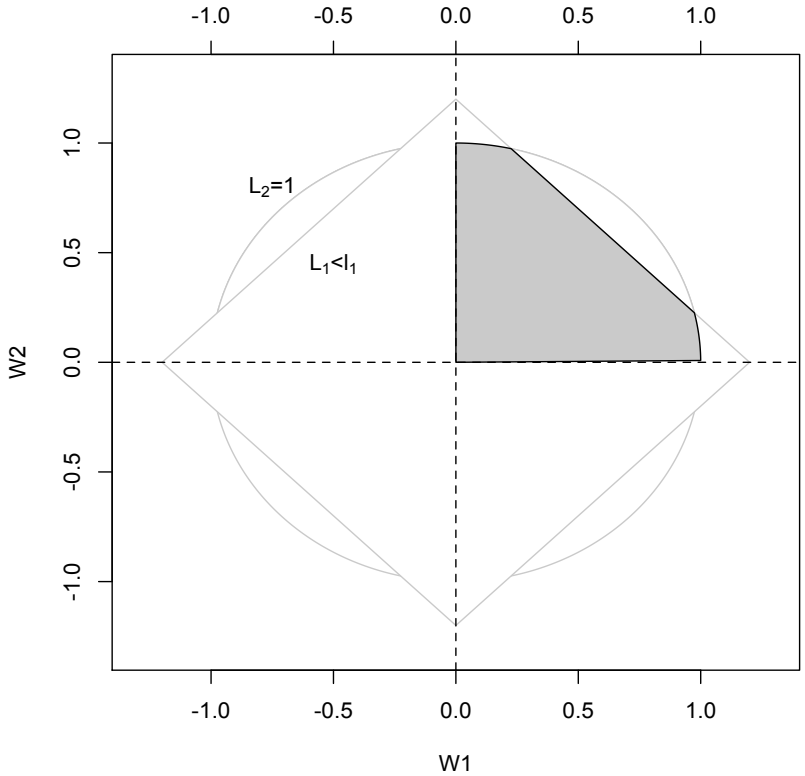


$$l_1 \leq 1$$

- $l_1 \leq 1$: The feature with the largest $BSS_j(\mathcal{C})$ receives all the weight.

The $L1$ tuning parameter value

The admissible regions for $\boldsymbol{w}$ in $\mathbb{R}^2$.

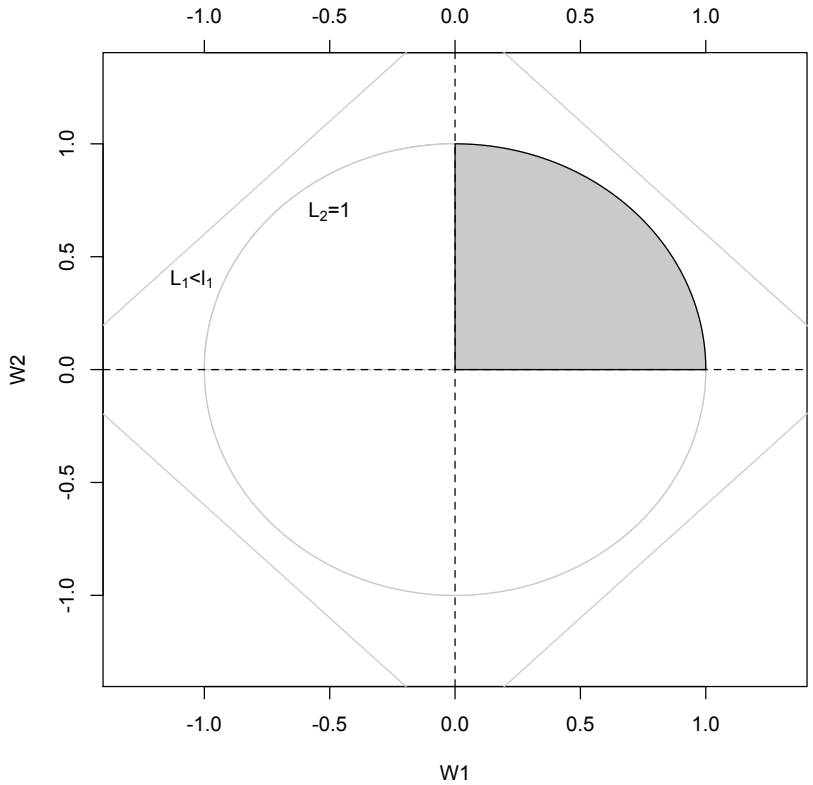


$$1 \leq l_1 \leq \sqrt{2}$$

- $1 < l_1 < \sqrt{p}$: Some features receive zero weights while others receive nonzero weights proportional to corresponding $BSS_j(\mathcal{C})$s.

The $L1$ tuning parameter value

The admissible regions for $\boldsymbol{w}$ in $\mathbb{R}^2$.

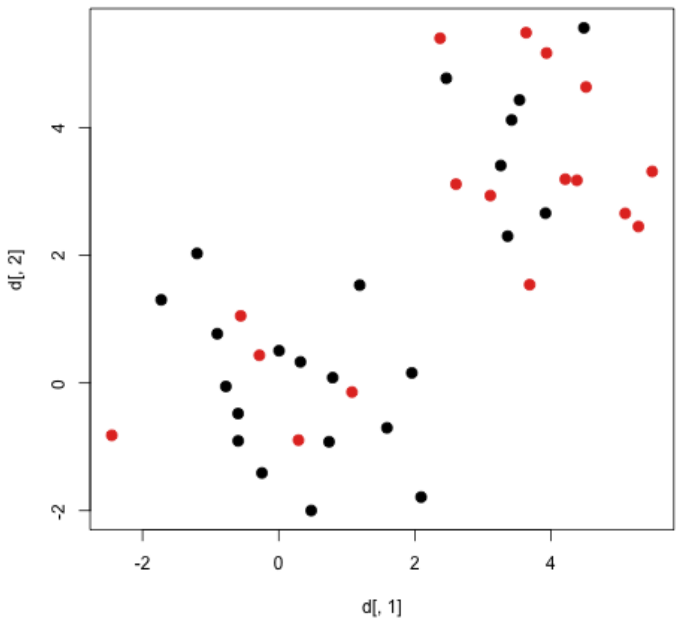


$$\sqrt{2} < l_1$$

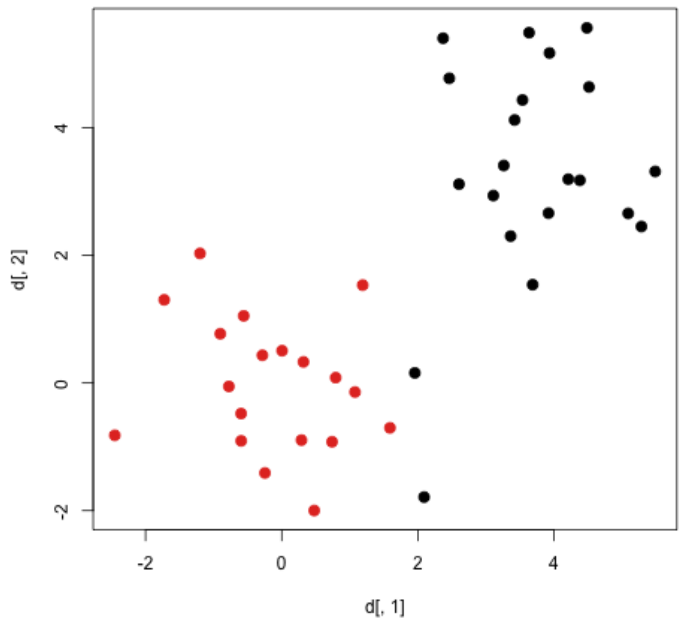
- $\sqrt{p} \leq l_1$: All the features receive nonzero weights that are proportional to corresponding $BSS_j(\mathcal{C})$s.

The performance of sparse 2-means on a simulated dataset 2

- $N = 40$
- $p = 1000$: 2 clustering features and 998 noise features



2-means



Sparse 2-means

Note: for sparse 2-means, the L_1 value is set to return 2 nonzero weights.

The performance of sparse K-means on a contaminated dataset

- $N = 40$
- $p = 1000$: 2 clustering features and 998 noise features
- 4 outliers in clustering features and noise featureas

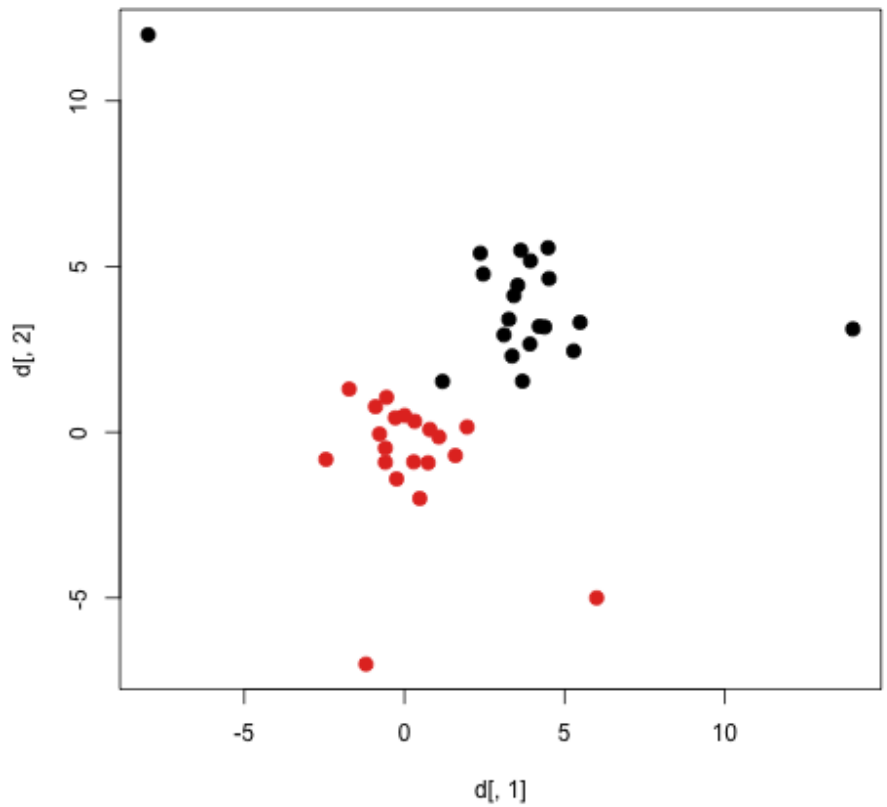


Figure: True cluster label

The performance of sparse K-means on a contaminated dataset

- $N = 40$,
- $p = 1000$: 2 clustering features and 998 noise features,
- 4 outliers are added in clustering features and noise features.

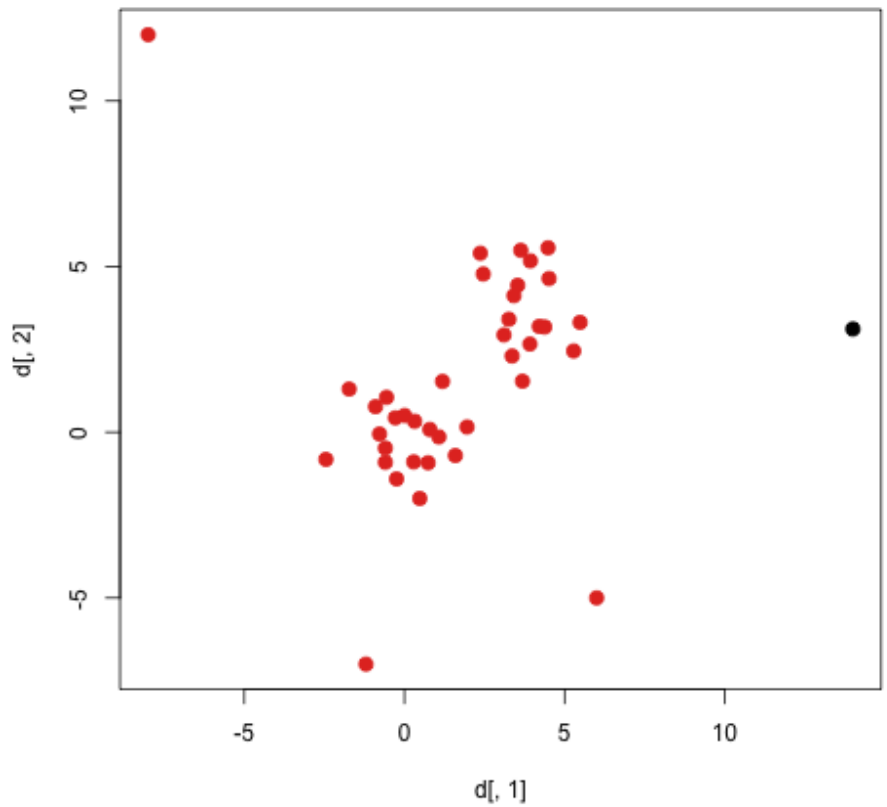


Figure: The partition from sparse 2-means

The performance of sparse K-means on a contaminated dataset

Our robustification of sparse K-means uses the idea of 'trimmed K-means' introduced by Gordaliza (1991 a).

Idea

The set of outliers, O, should be excluded in the definition of the cluster centers and within cluster sum of squares.

But we do not know which cases are outliers!

Trim $\alpha 100\%$ of cases with the largest squared Euclidean distances to their cluster centers when the cluster centers are defined in the standard K-means algorithm.

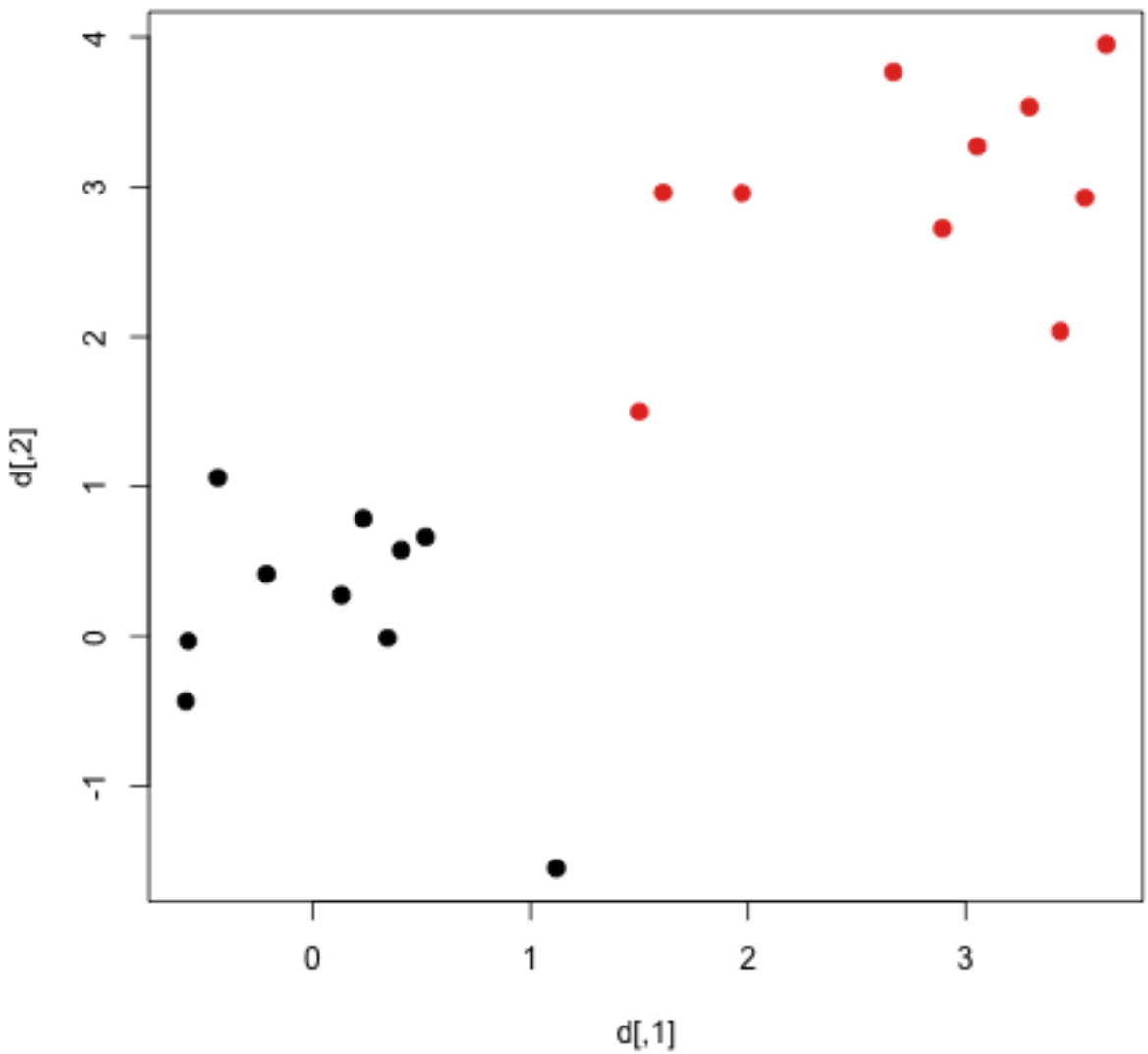


Figure: 20 cases are generated from a mixture of two bivariate normal distributions. The trimming proportion α is set to $2/20$.

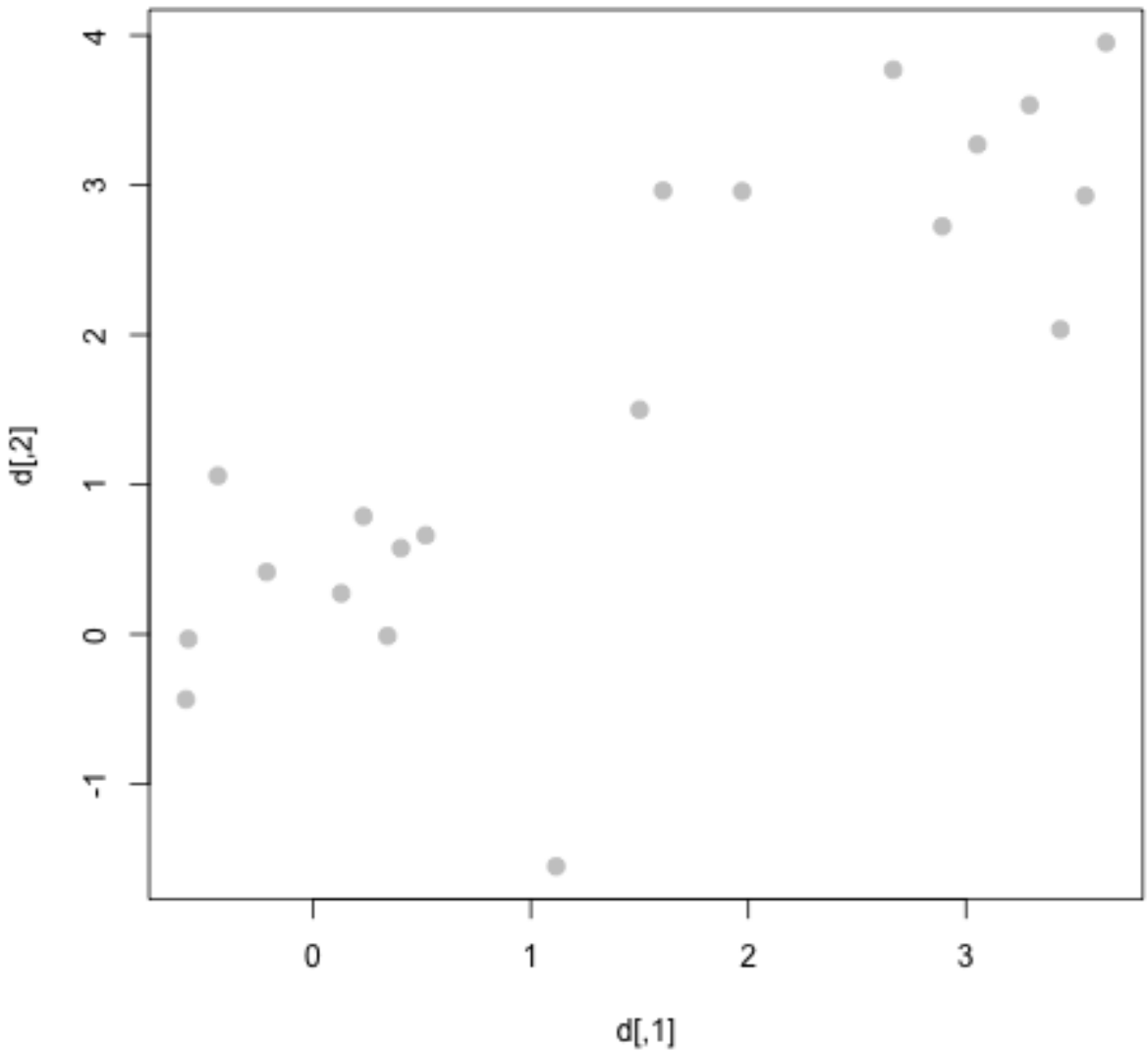


Figure:

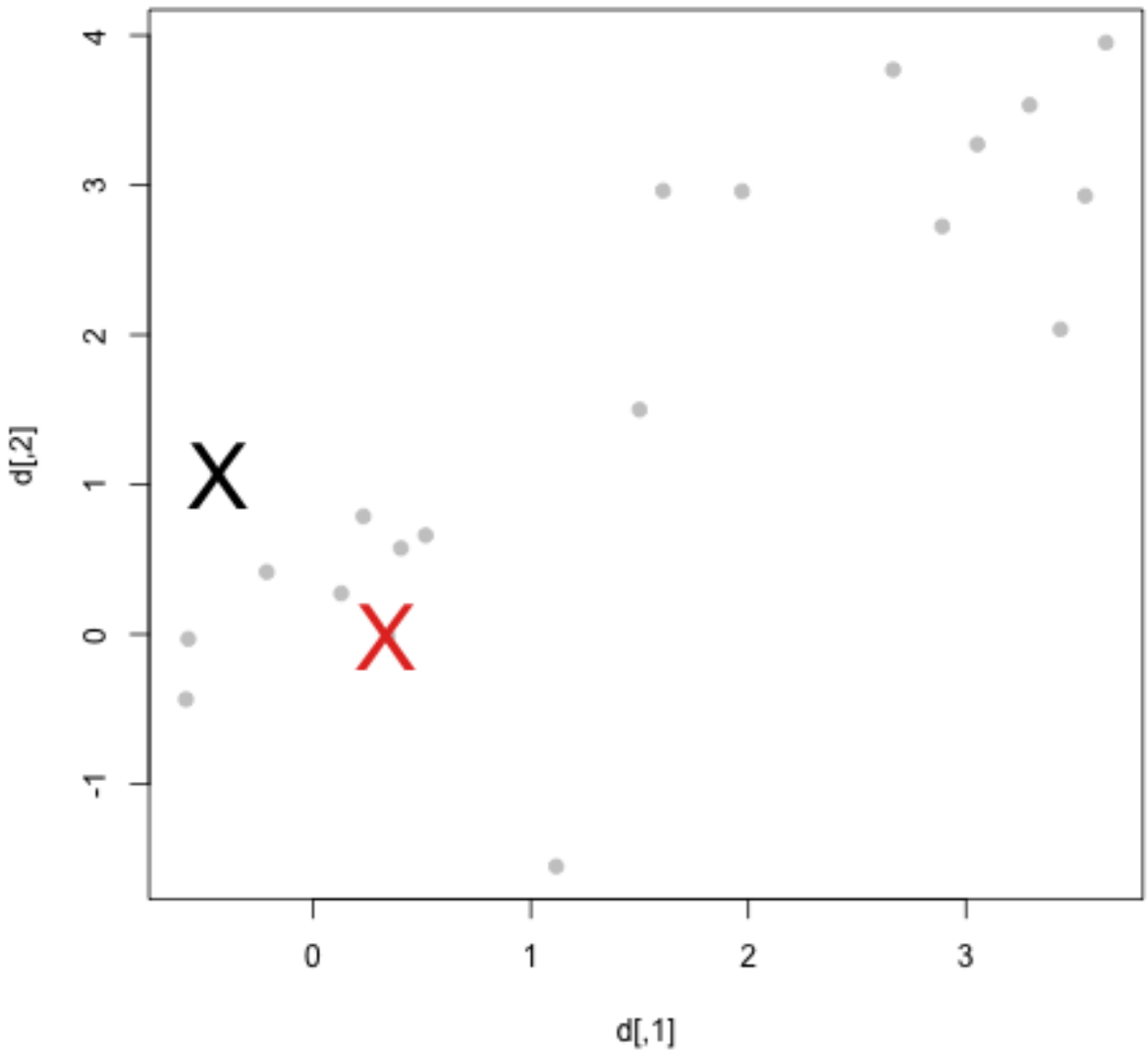


Figure: Randomly set initial cluster centers

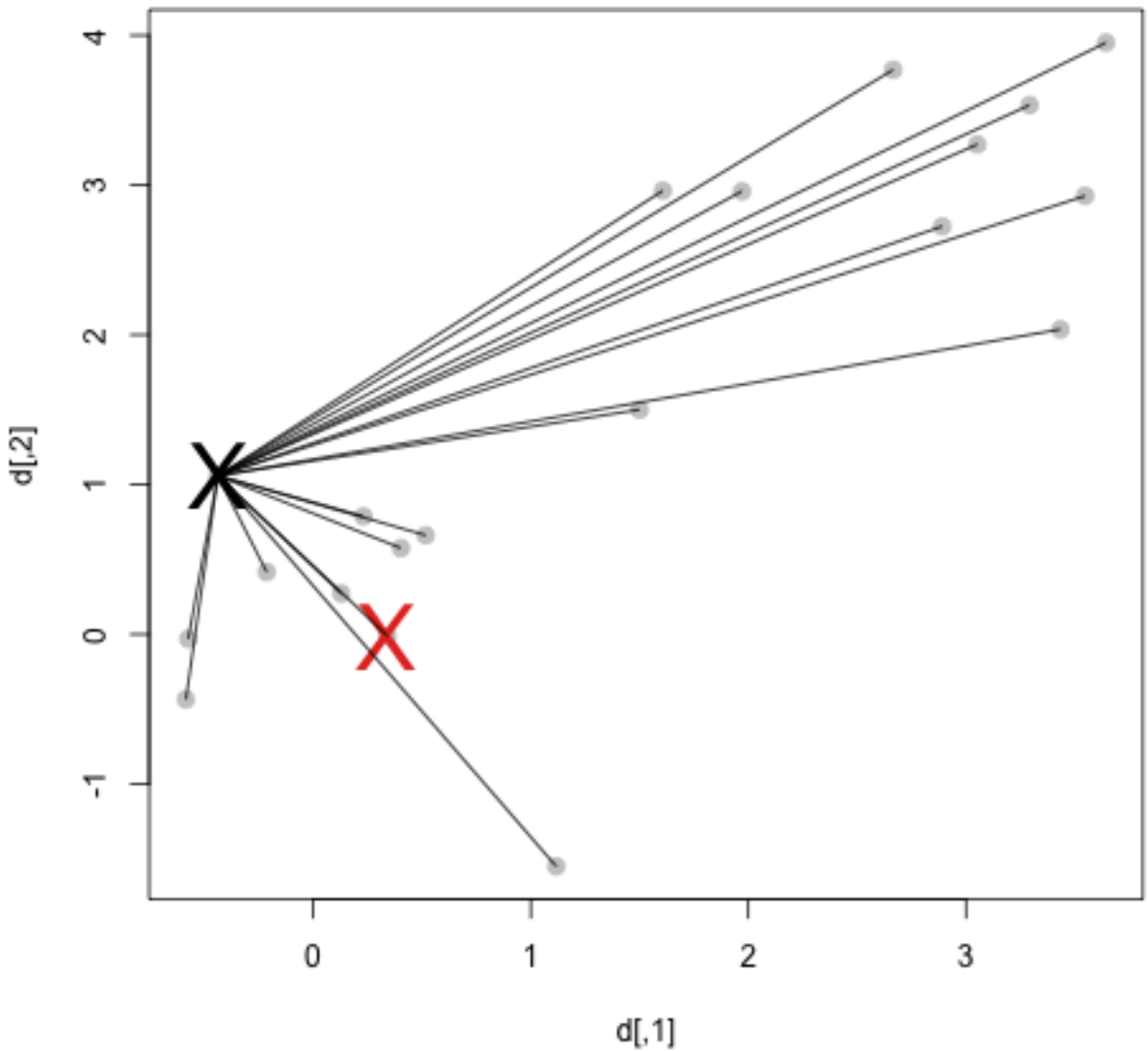


Figure: itr=1: given cluster centers, update class assignment.

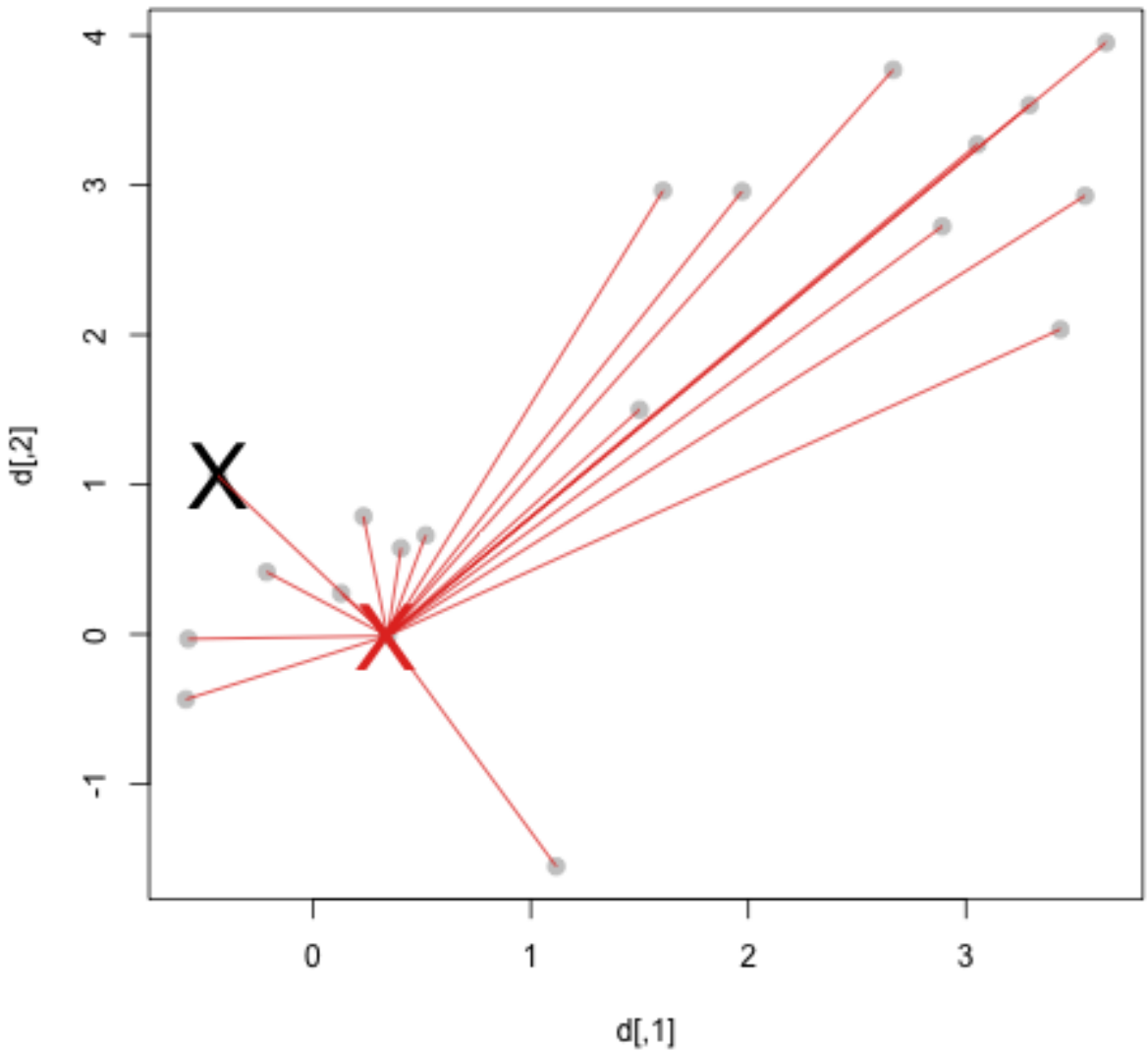


Figure: itr=1: given cluster centers, update class assignment.

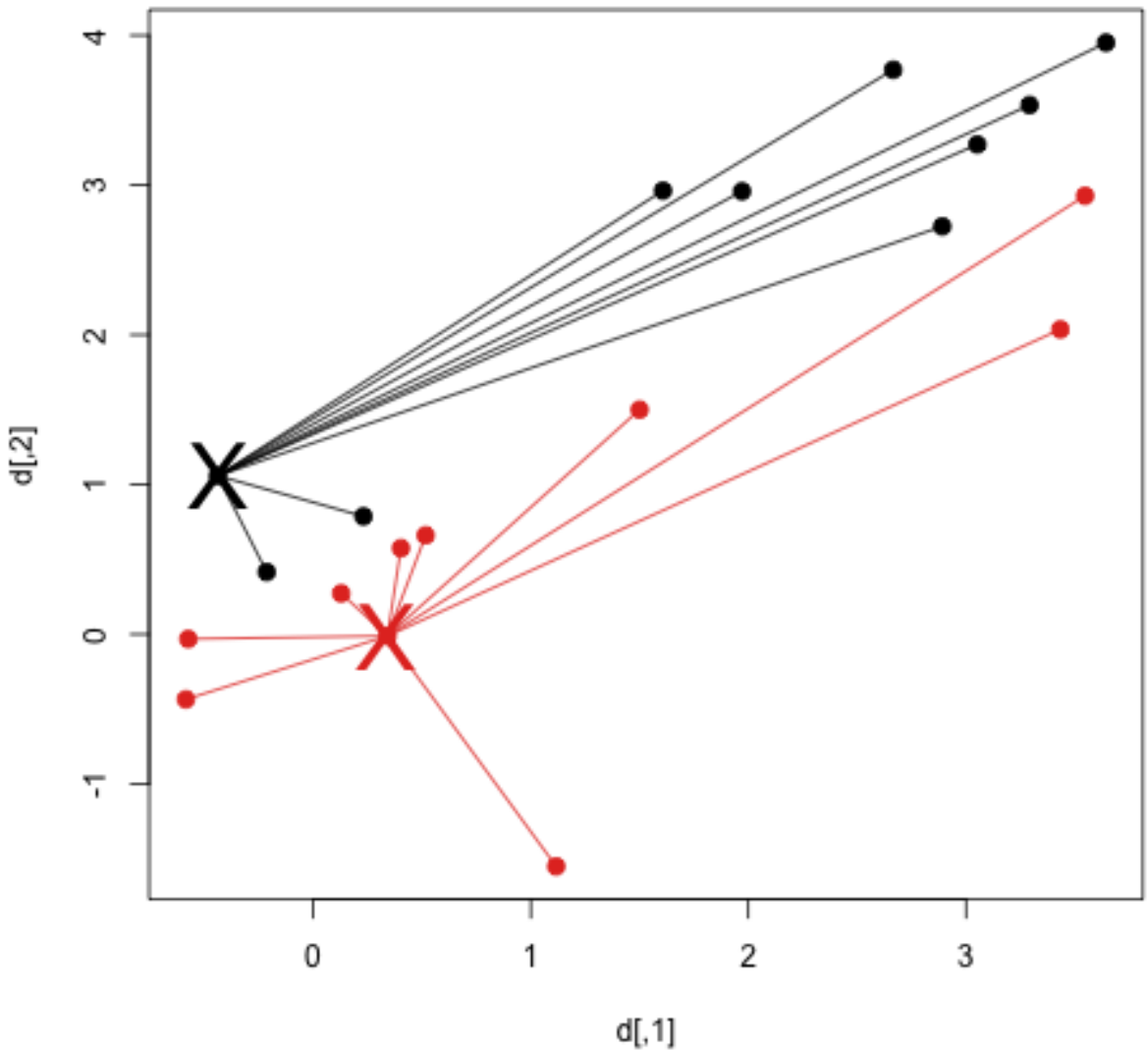


Figure: itr=1: given cluster centers, update class assignment.

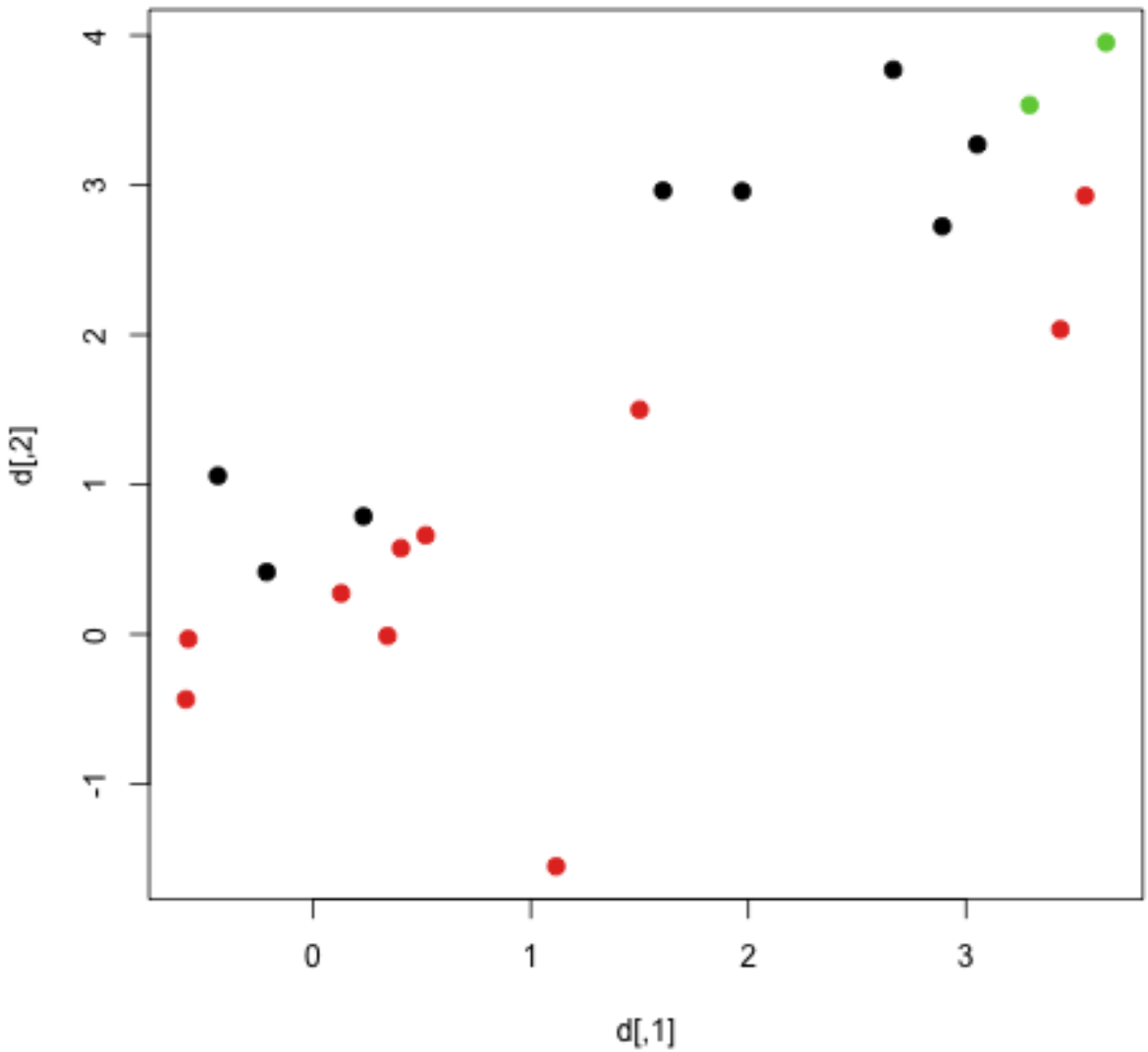


Figure: itr=1: trim cases

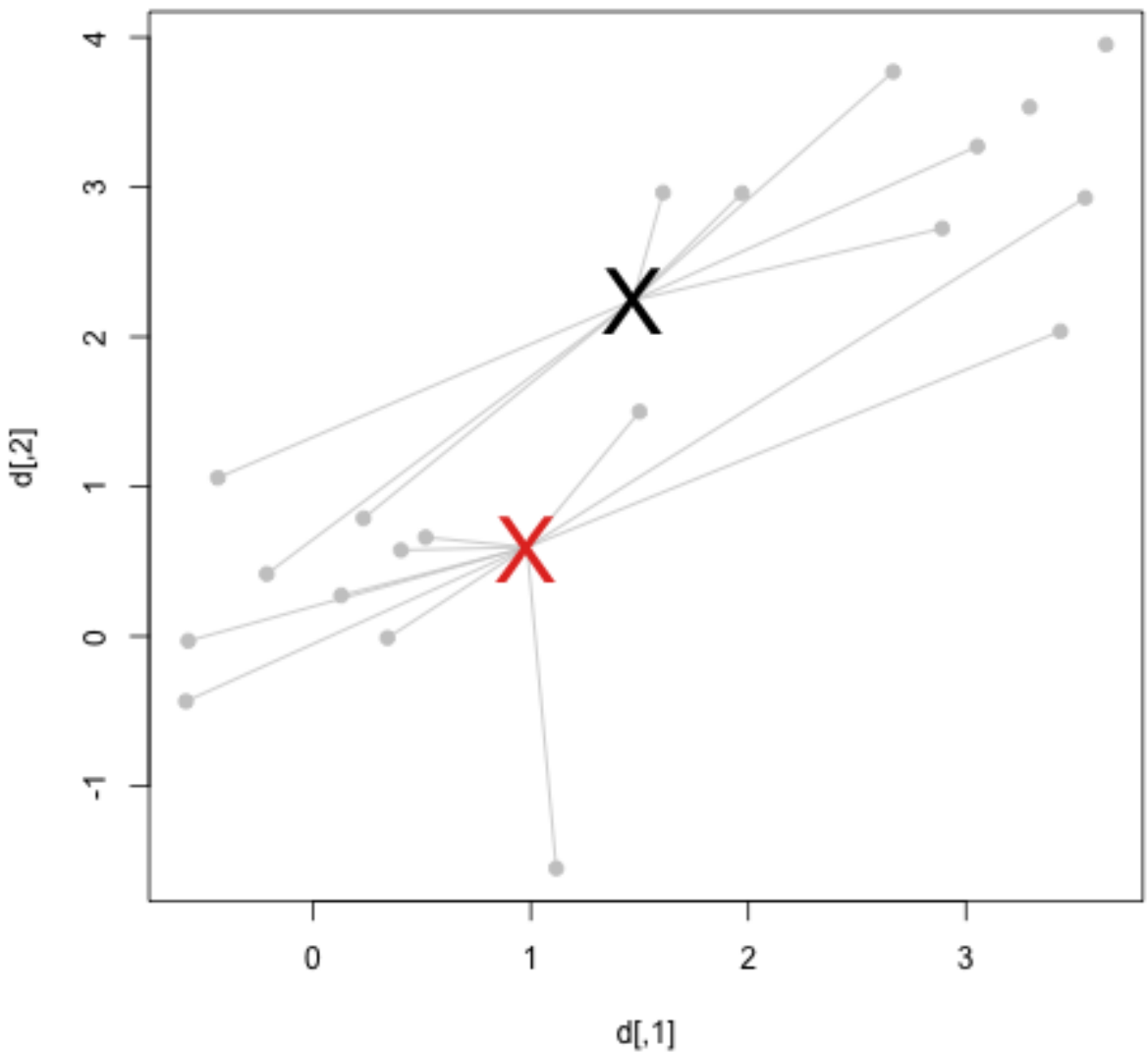


Figure: itr=1: given class assignments and trimmed cases update cluster centers.

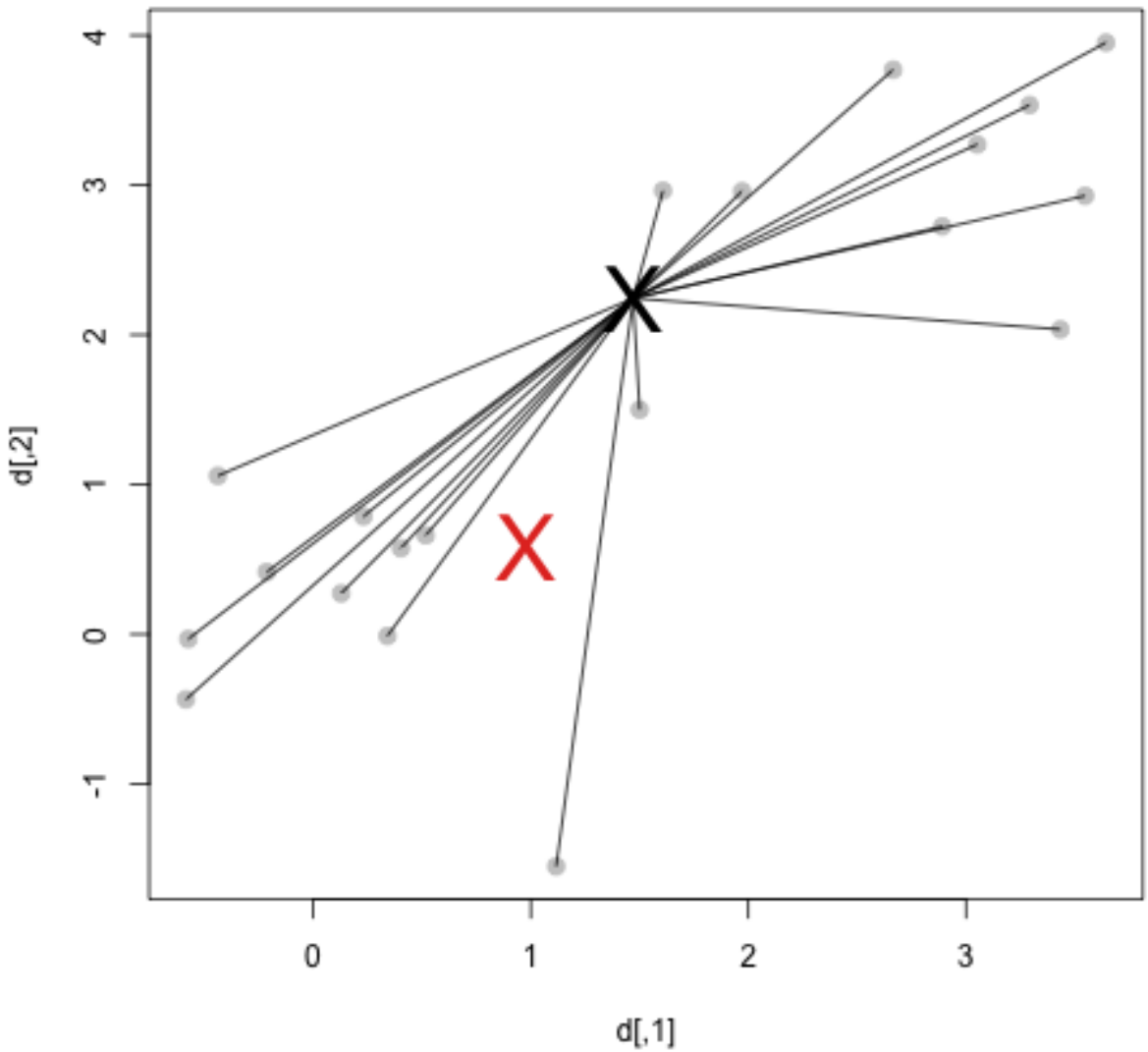


Figure: itr=2: given cluster centers, update the class assignments.

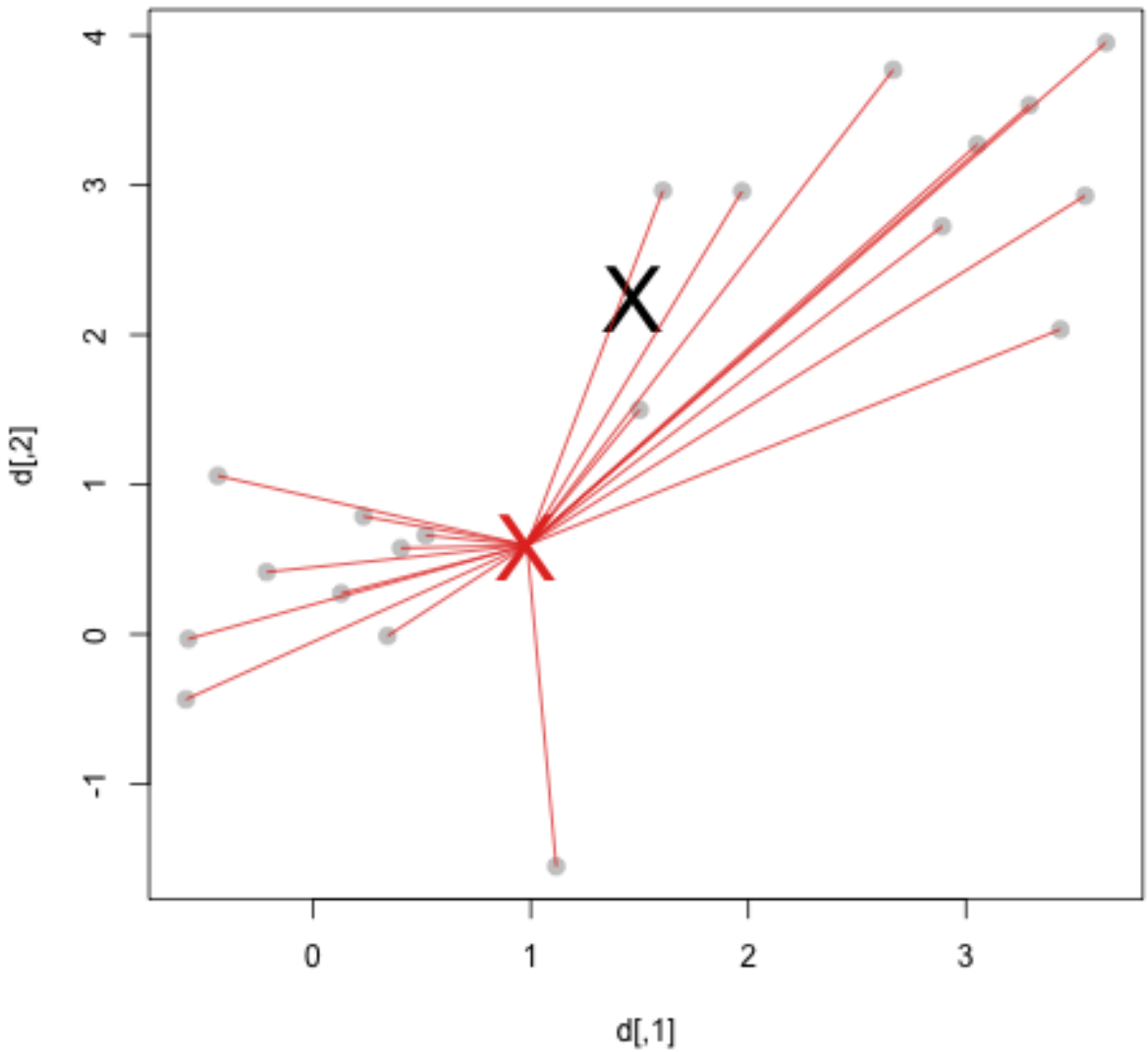


Figure: itr=2: given cluster centers, update the class assignments.

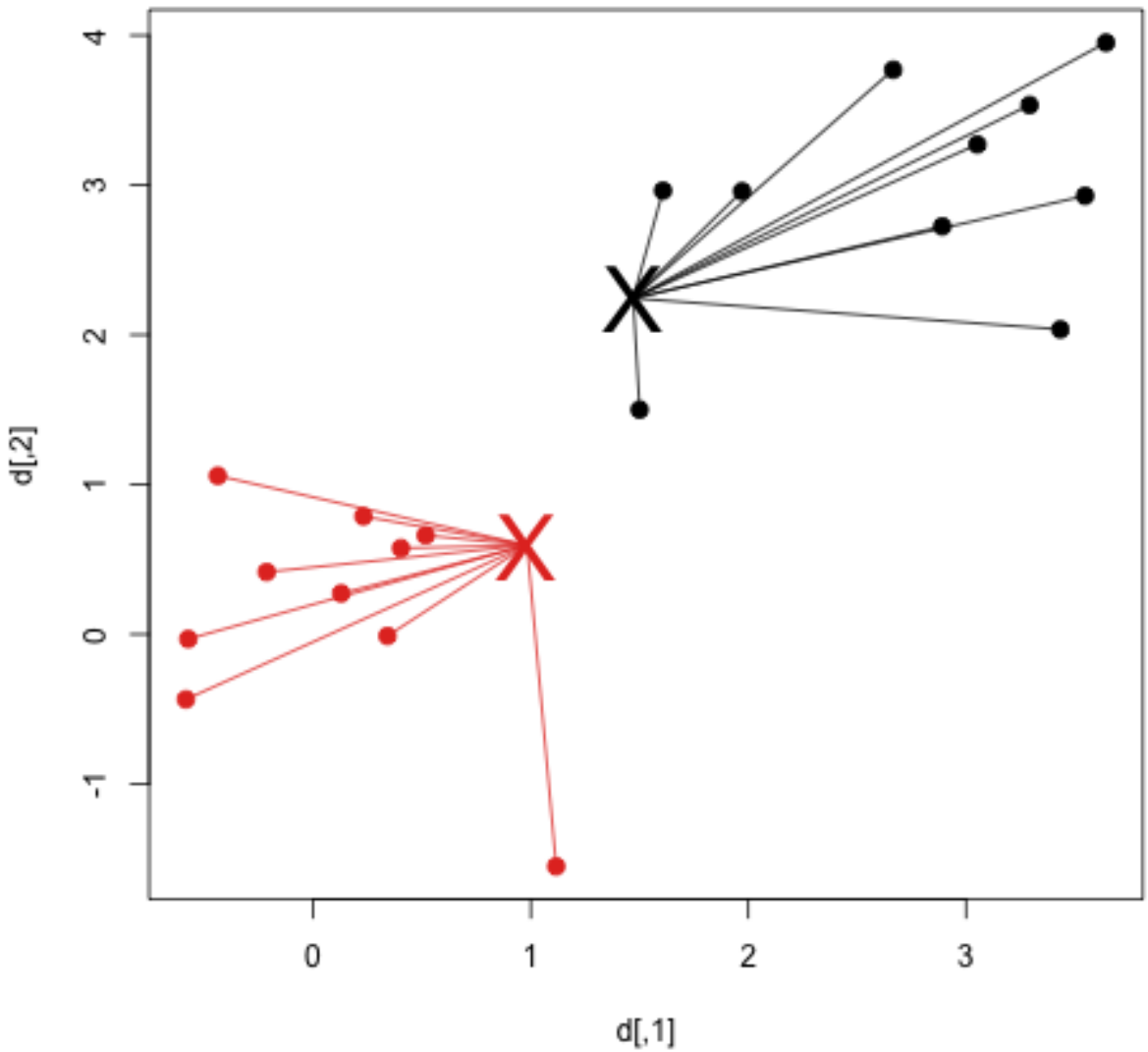


Figure: itr=2: given cluster centers, update the class assignments.

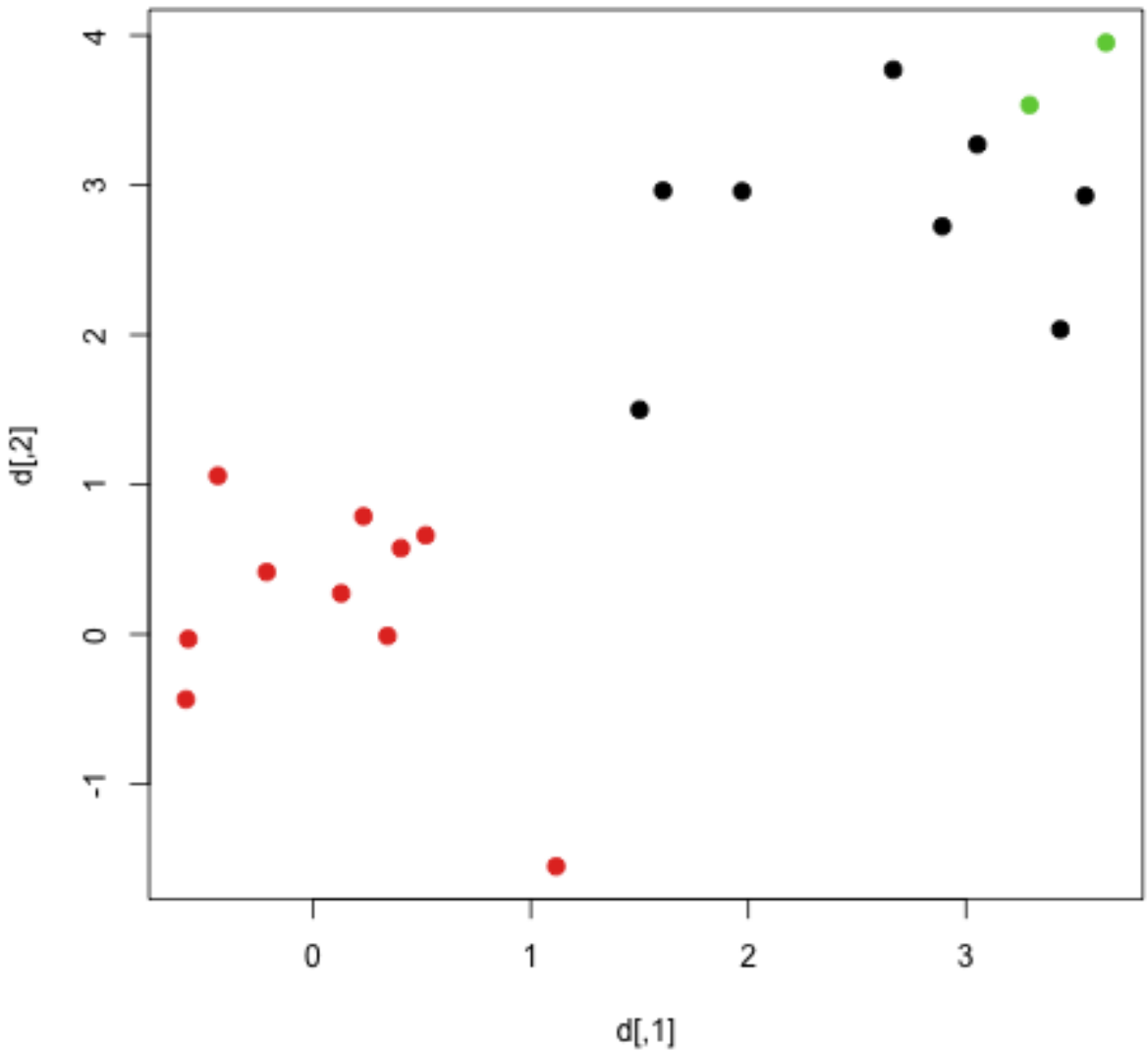


Figure: itr=2: trim cases.

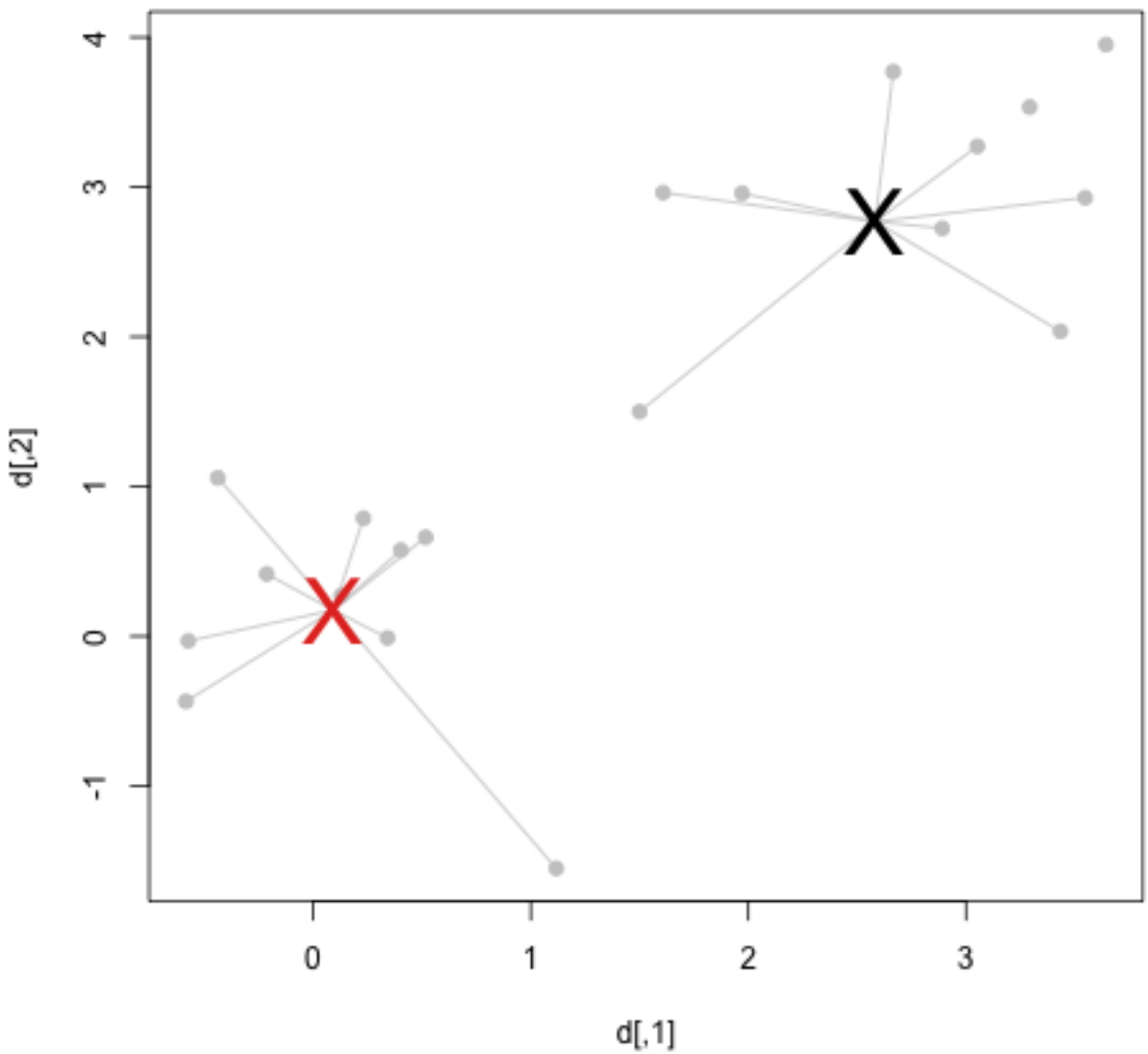


Figure: itr=2: given class assignments and trimmed cases, update cluster centers .

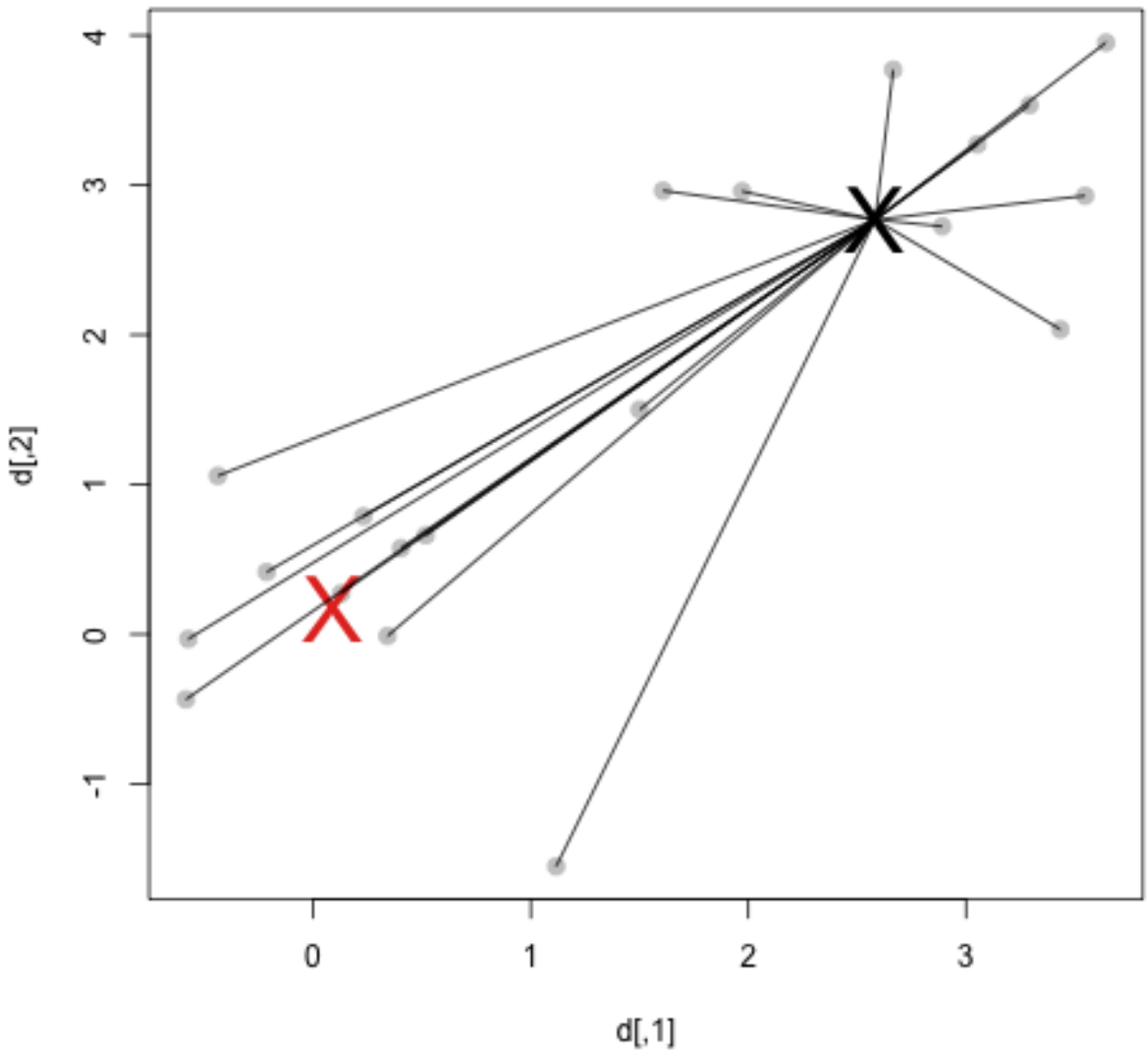


Figure: itr=3: given cluster centers, update the class assignments.

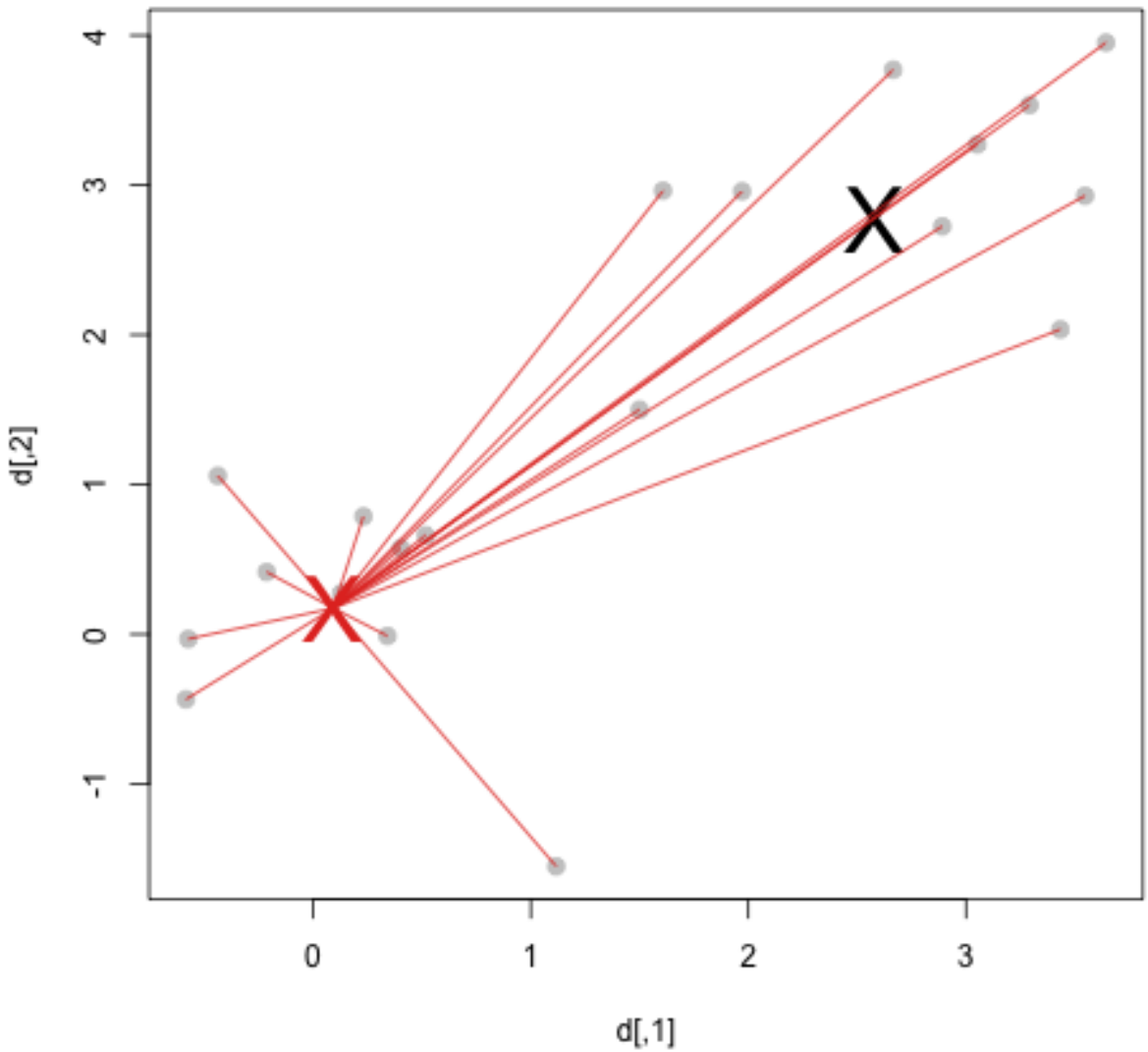


Figure: itr=3: given cluster centers, update the class assignments.

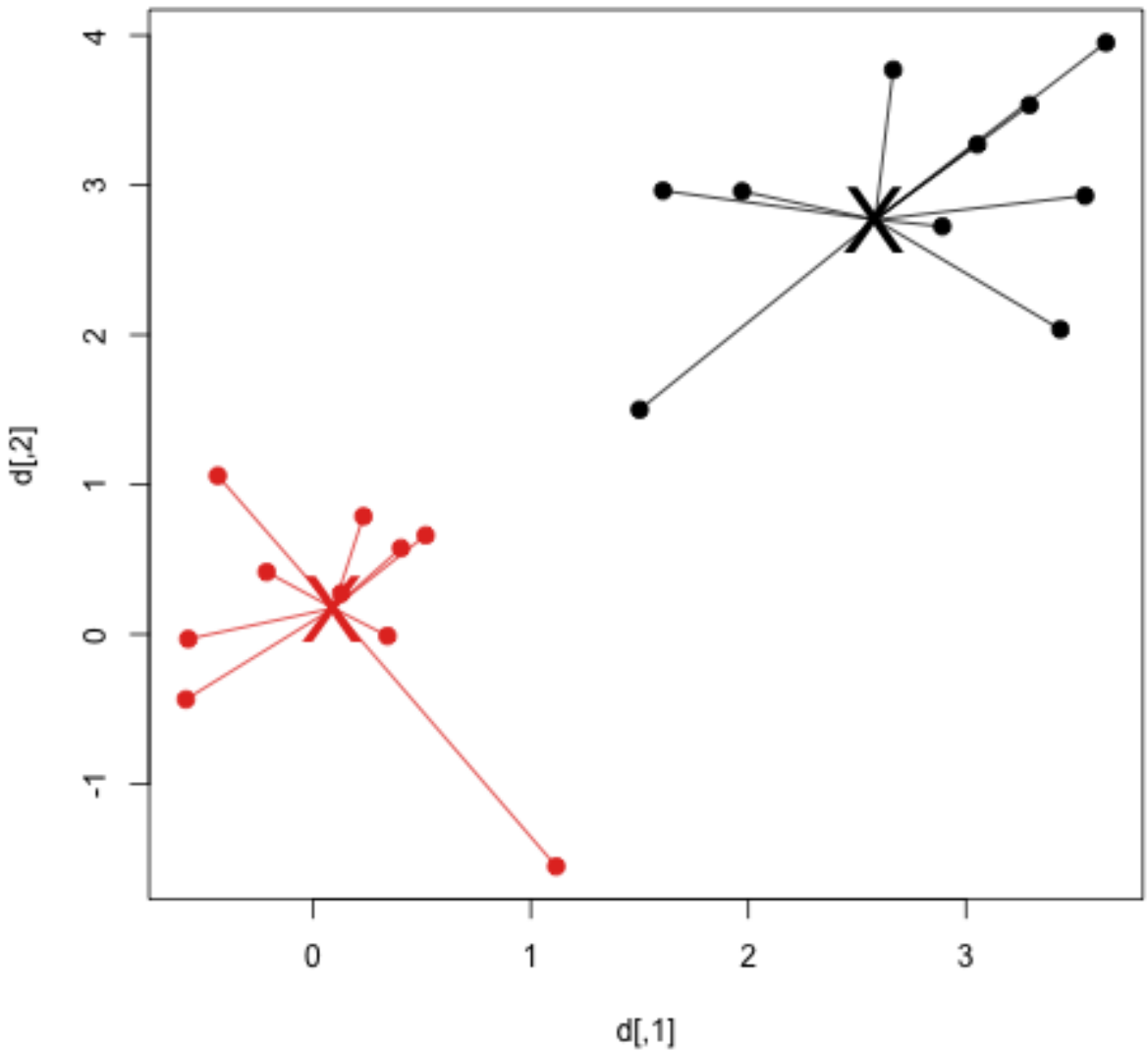


Figure: itr=3: given cluster centers, update the class assignments.

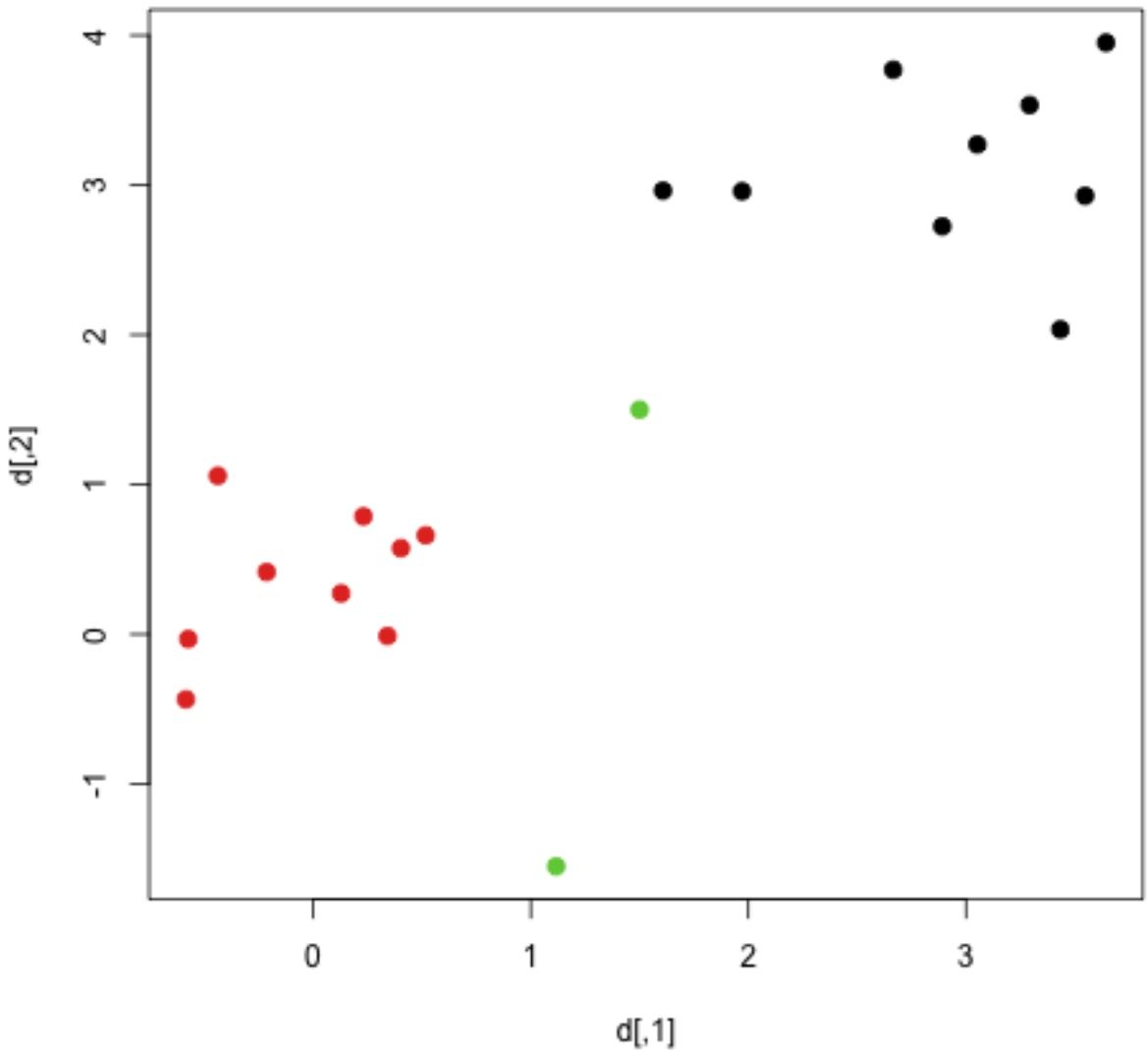


Figure: itr=3: trim cases.

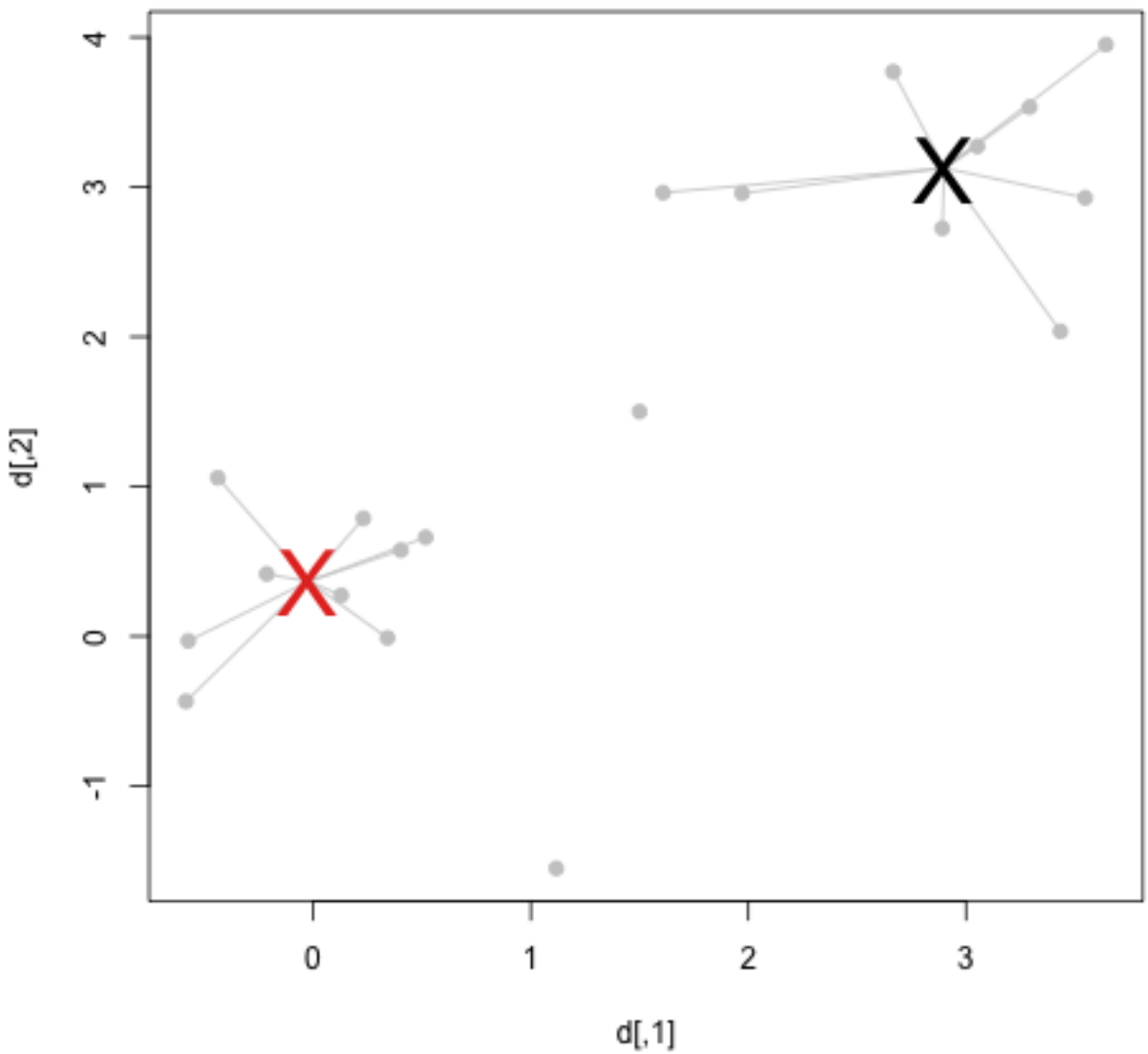


Figure: itr=3: given class assignments and trimmed cases, update cluster centers .

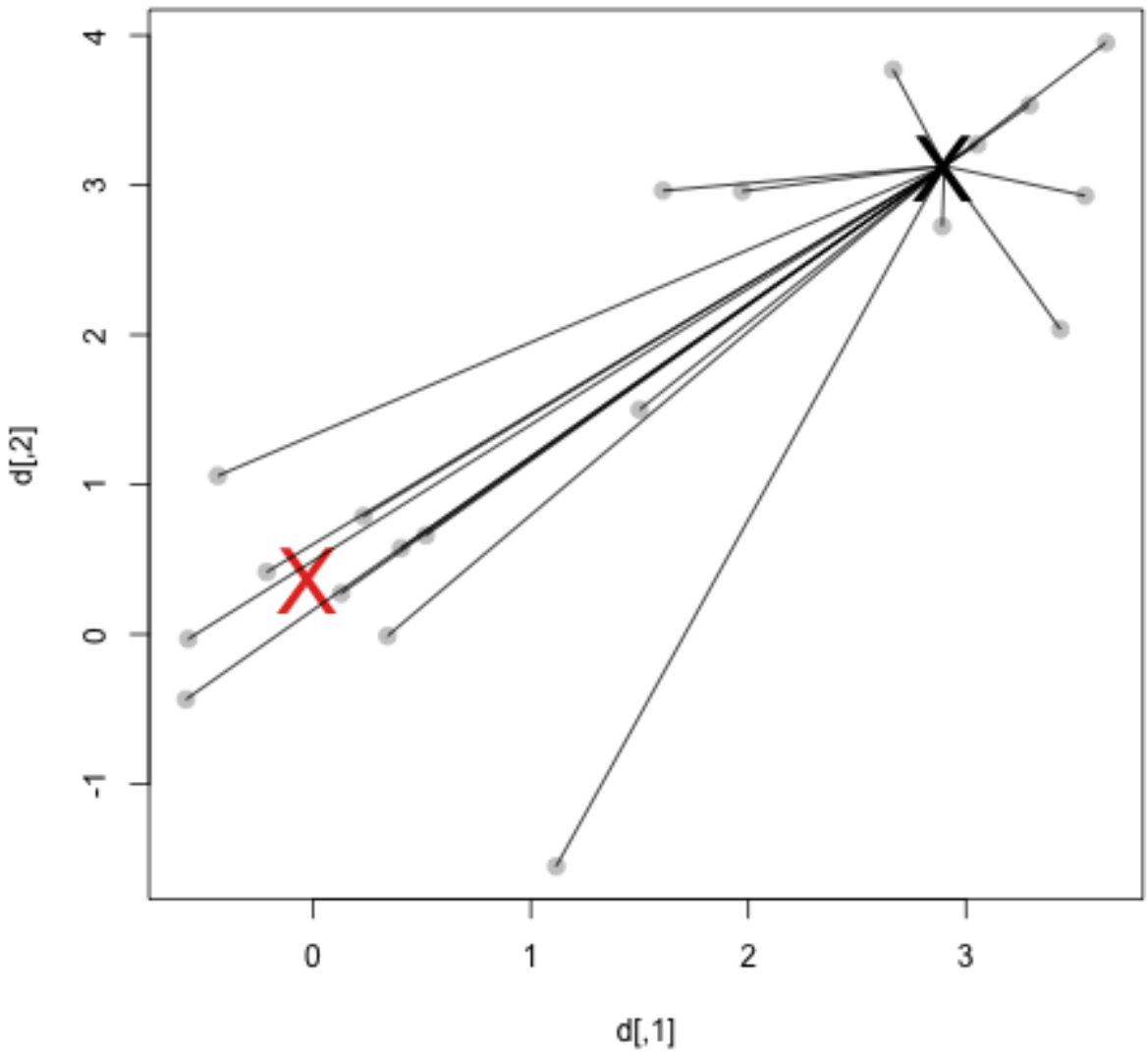


Figure: itr=4: given cluster centers, update the class assignments.

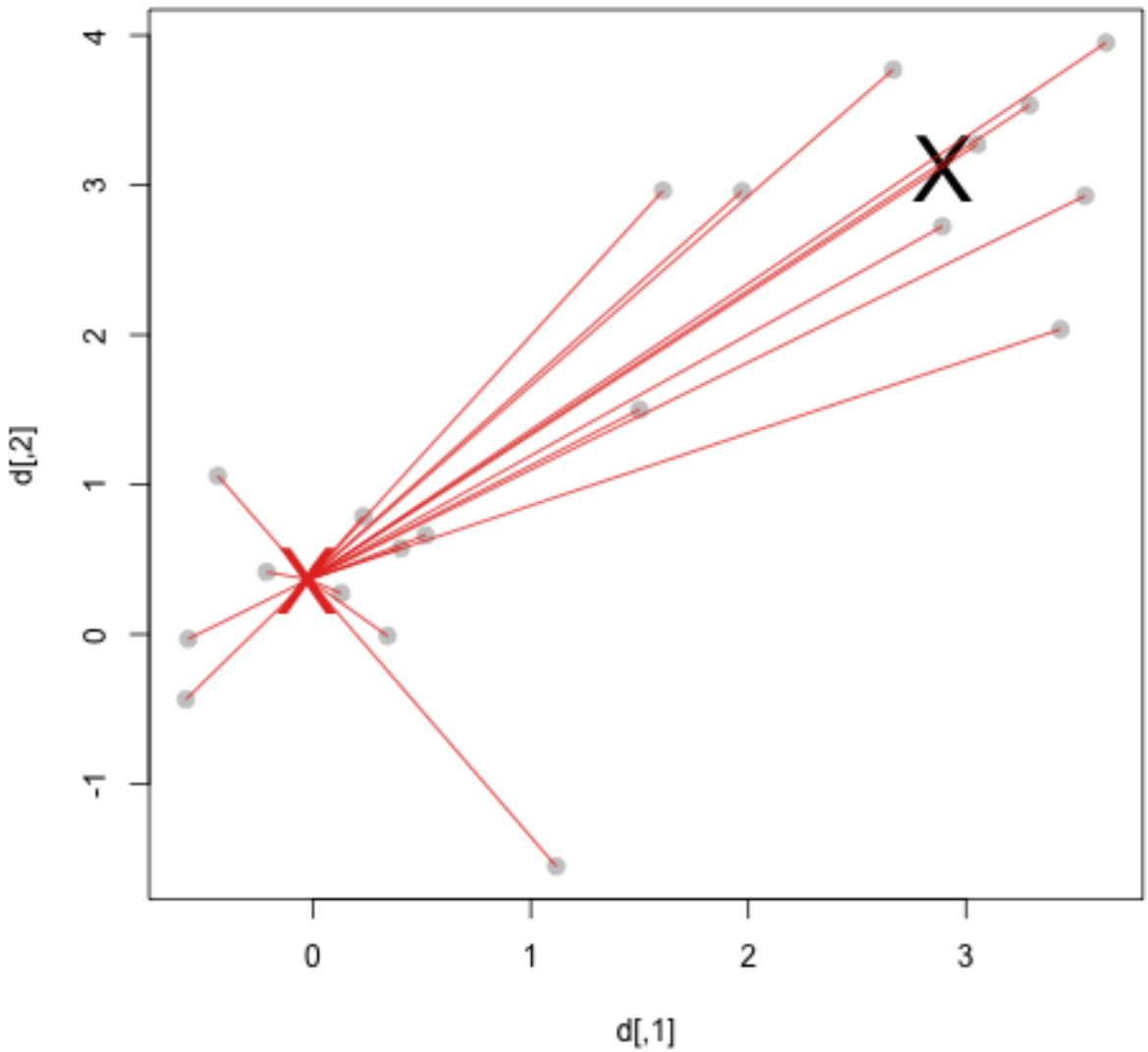


Figure: itr=4: given cluster centers, update the class assignments.

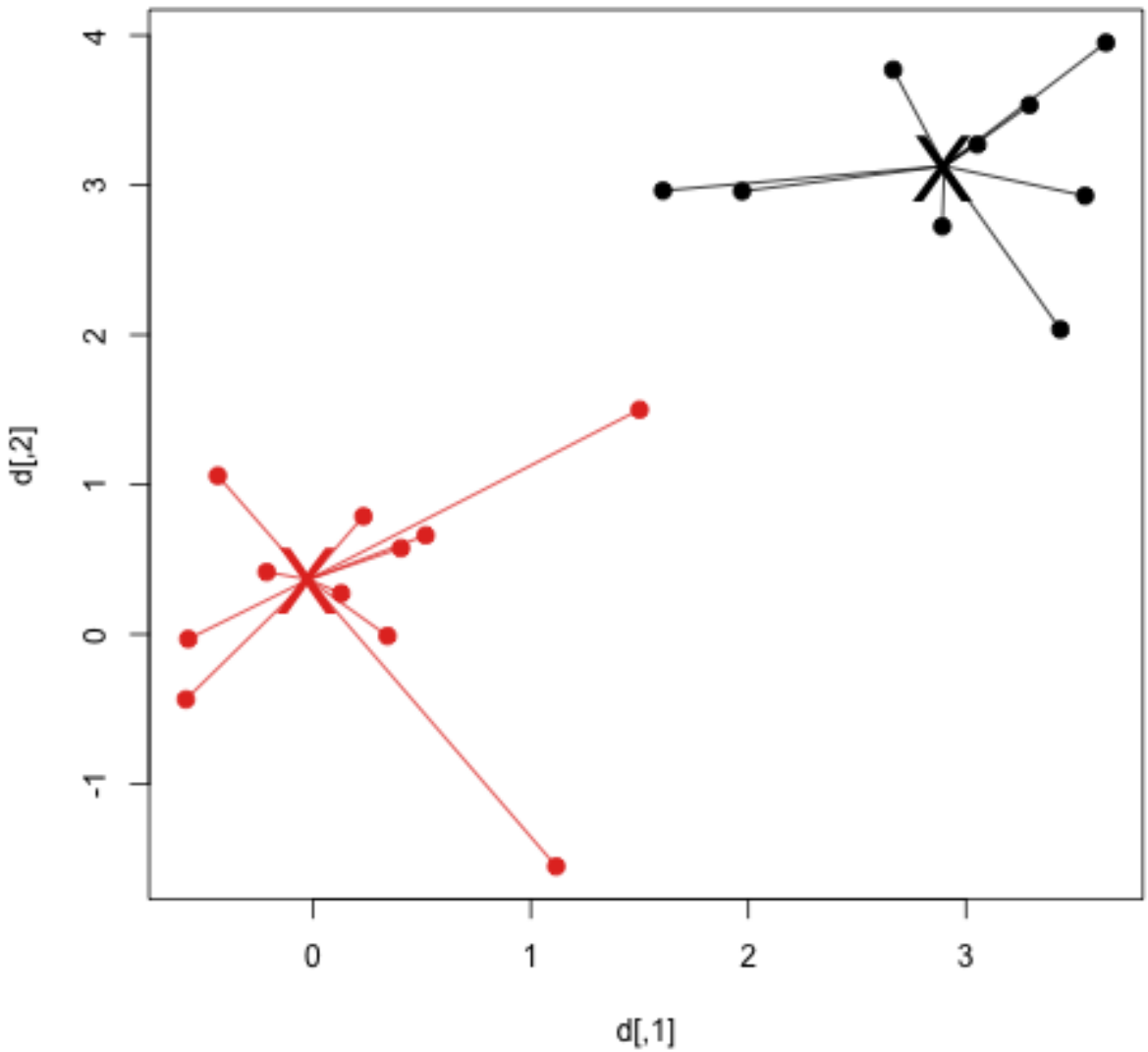


Figure: itr=4: given cluster centers, update the class assignments.

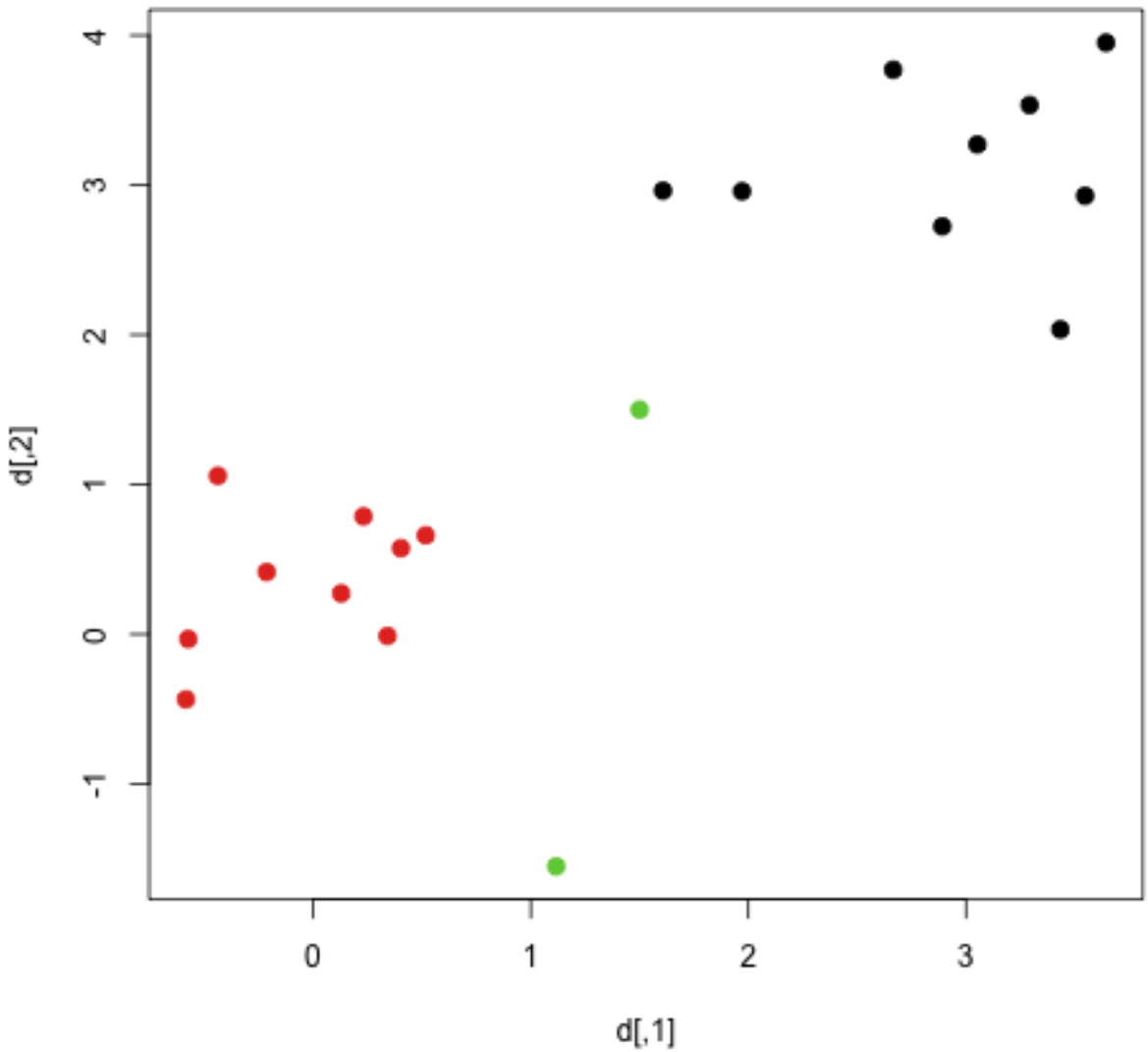


Figure: itr=4: trim cases.

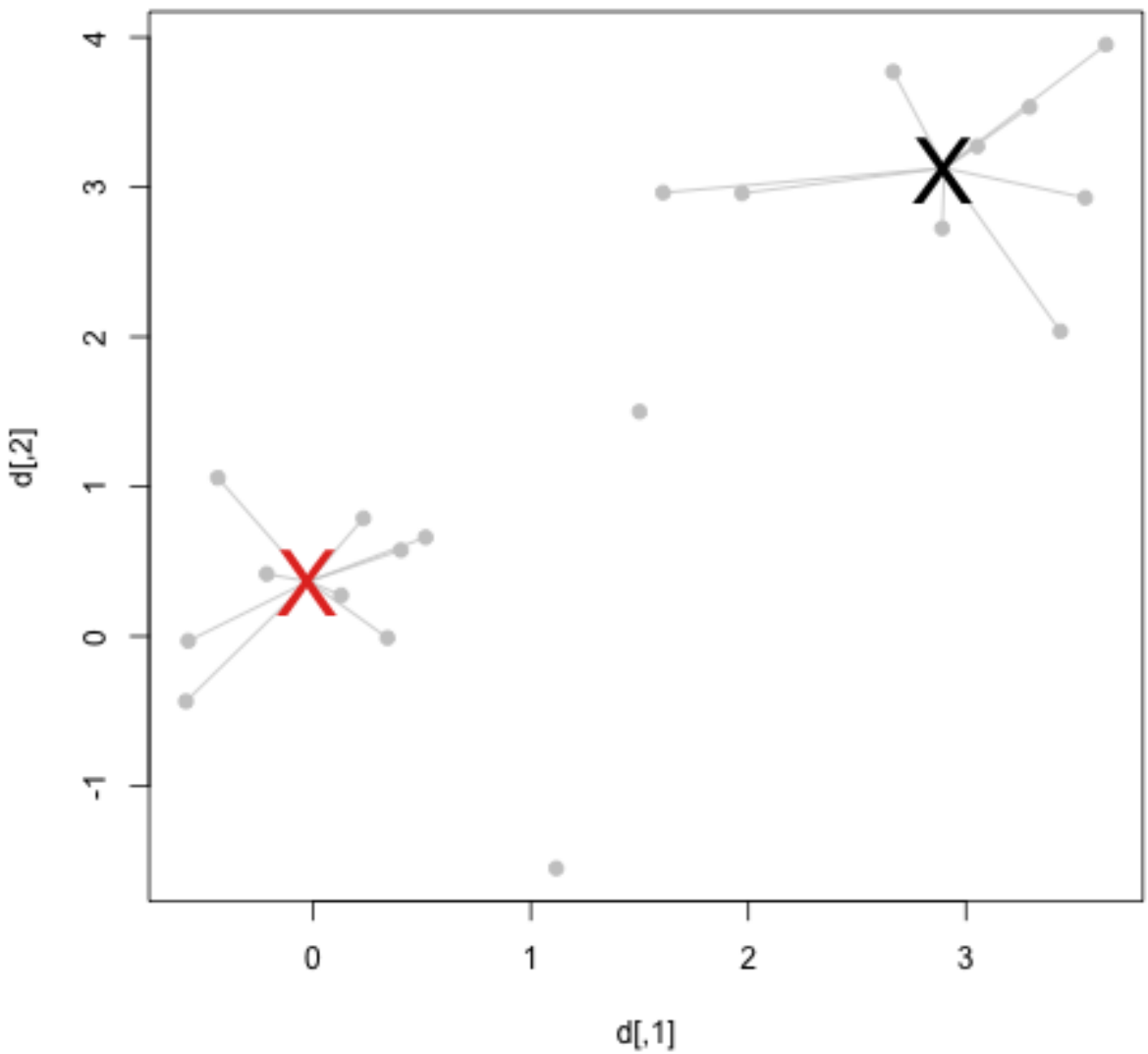


Figure: itr=4: given class assignments and trimmed cases, update cluster centers .

Naive robust sparse K-means

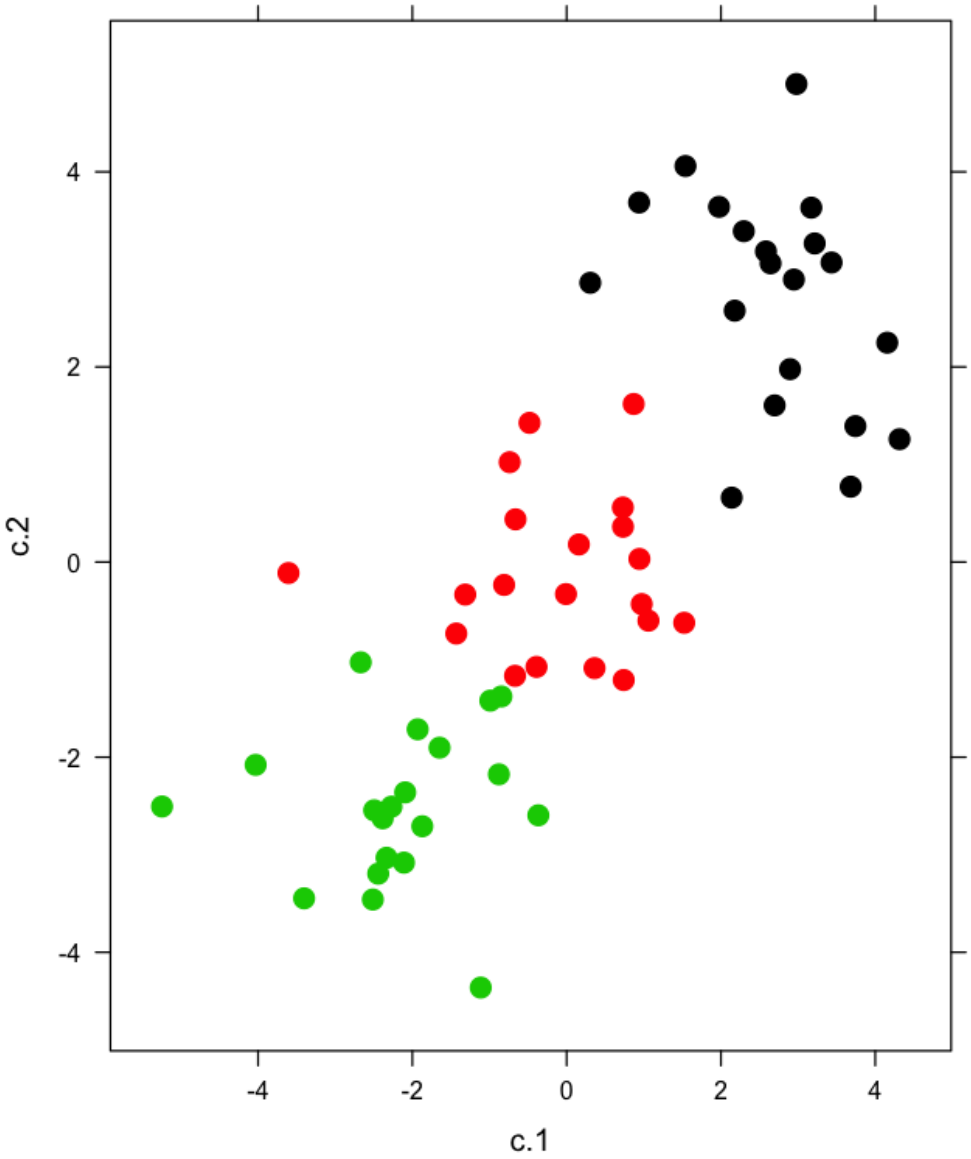
Idea

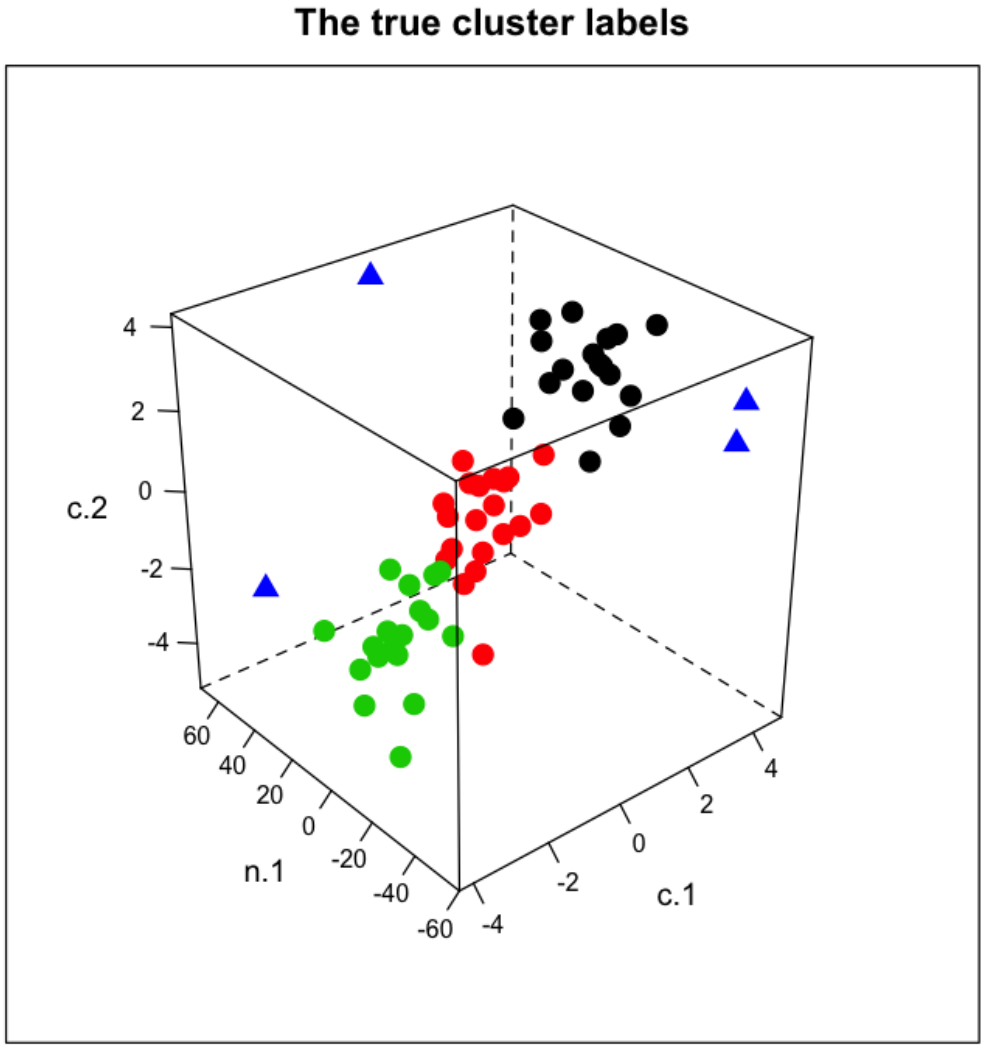
Trim cases in the weighted squared Euclidean distances.

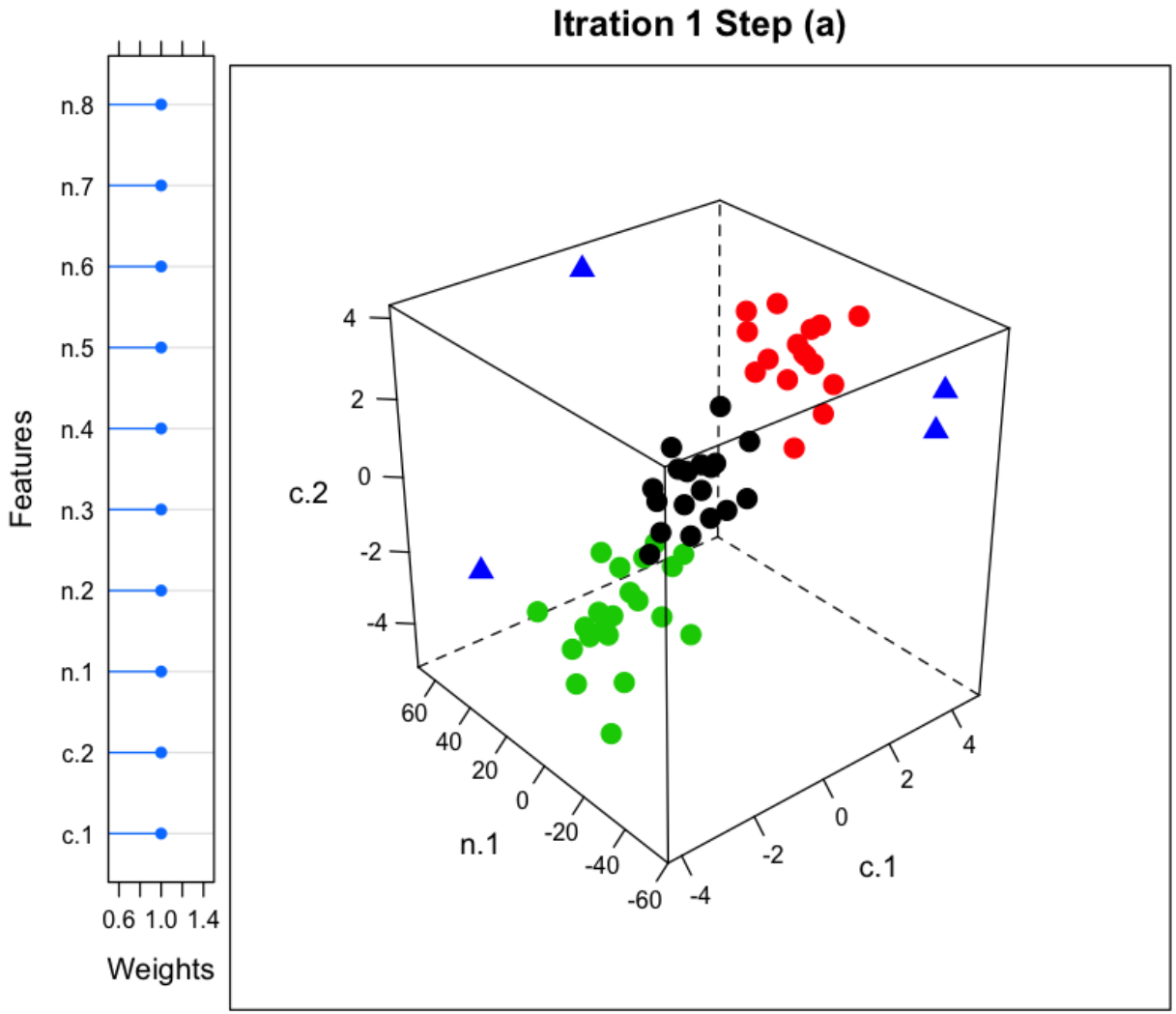
- Naive Robust Step(a):
  given $\boldsymbol{W}$ perform trimmed K-means on a transformed dataset, return $\mathcal{C}$ and trimmed cases.
- Naive Robust Step(b):
  given $\mathcal{C}$ and the trimmed cases, maximize weighted BSS, without trimmed cases, return $\boldsymbol{W}$.

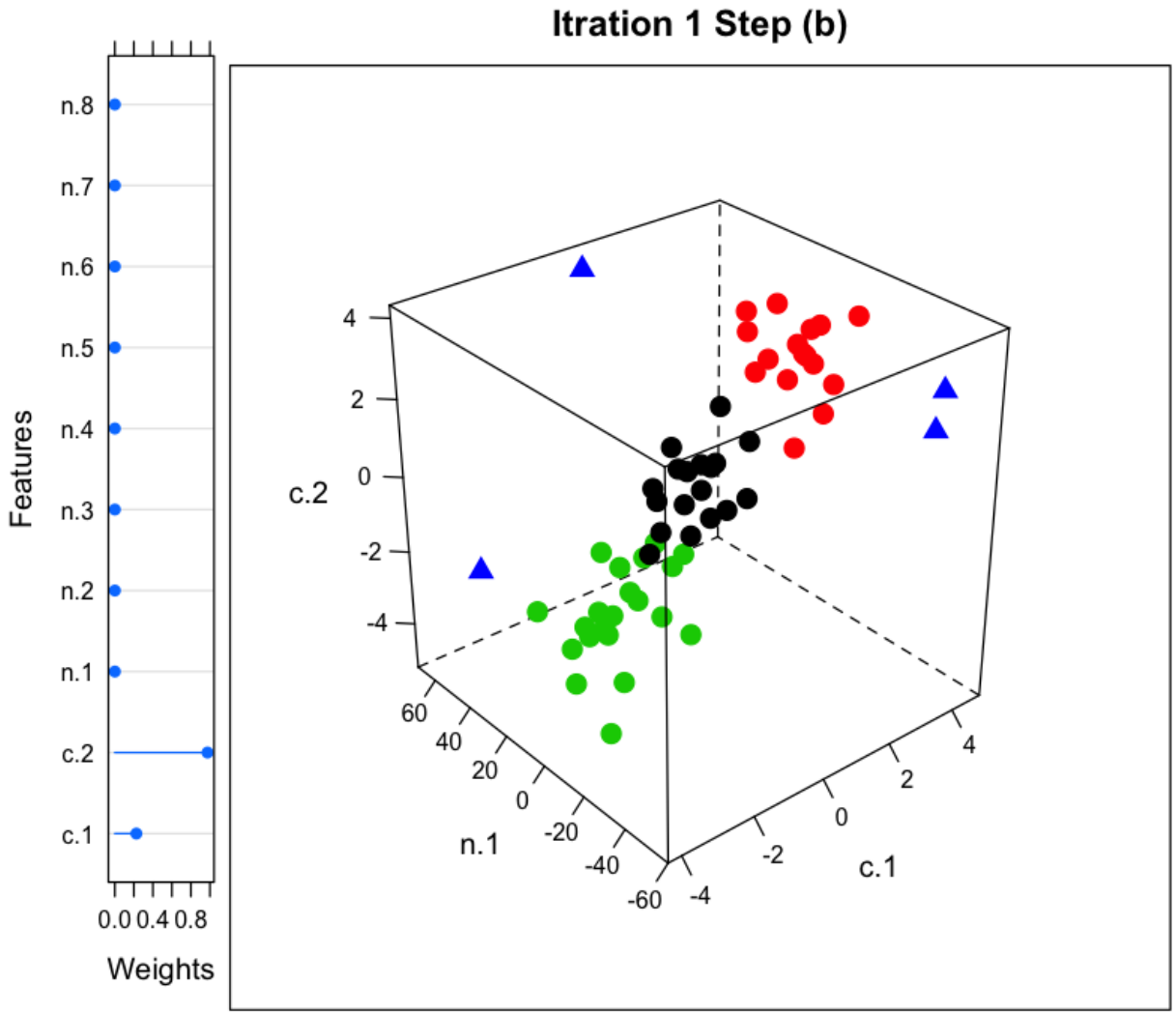
Potential issues

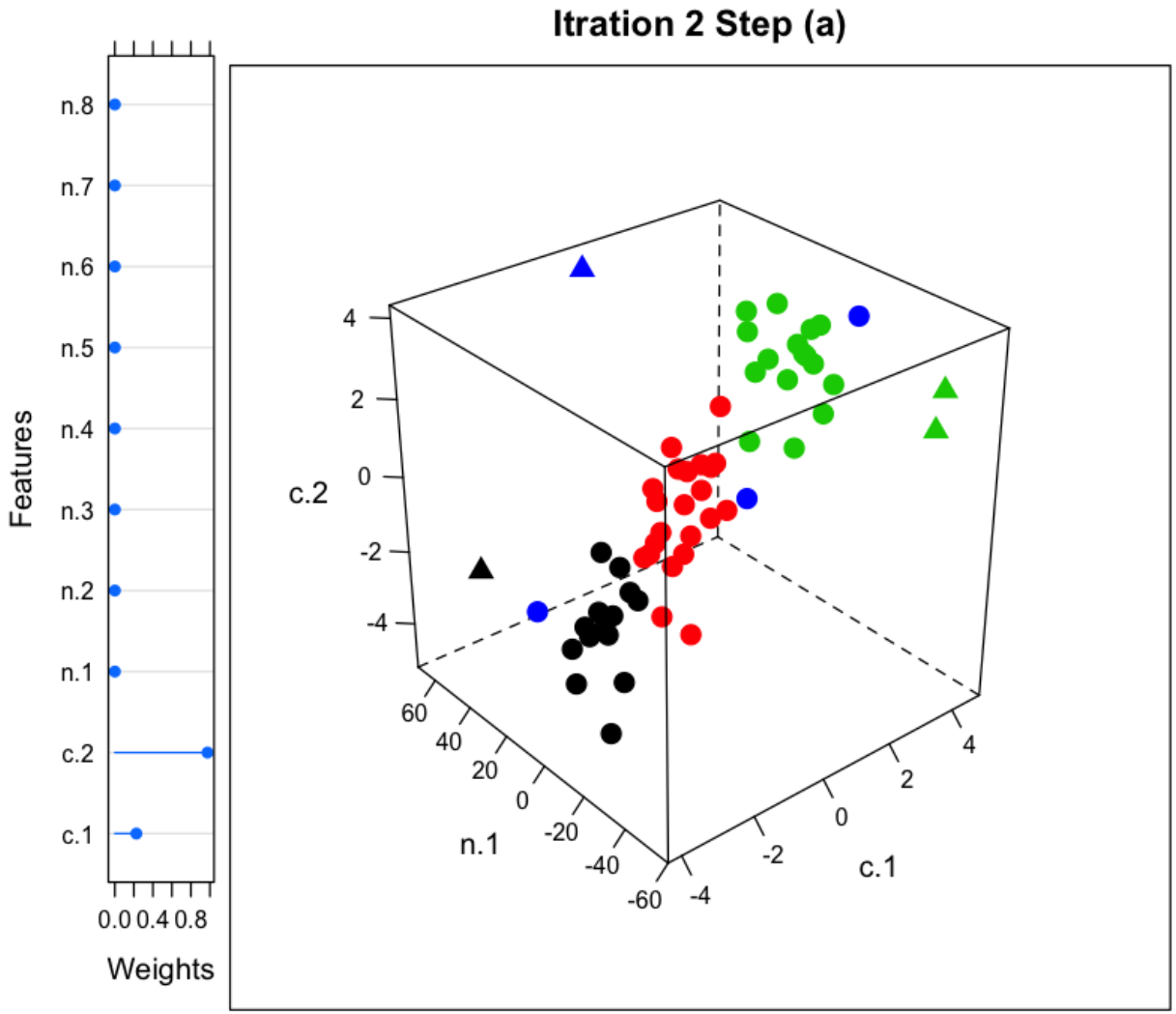
However, if a case is outlying in a feature that receive a small weight in Step (b), this case can "survive" the next Step (a) and upset the selection of feature weights.

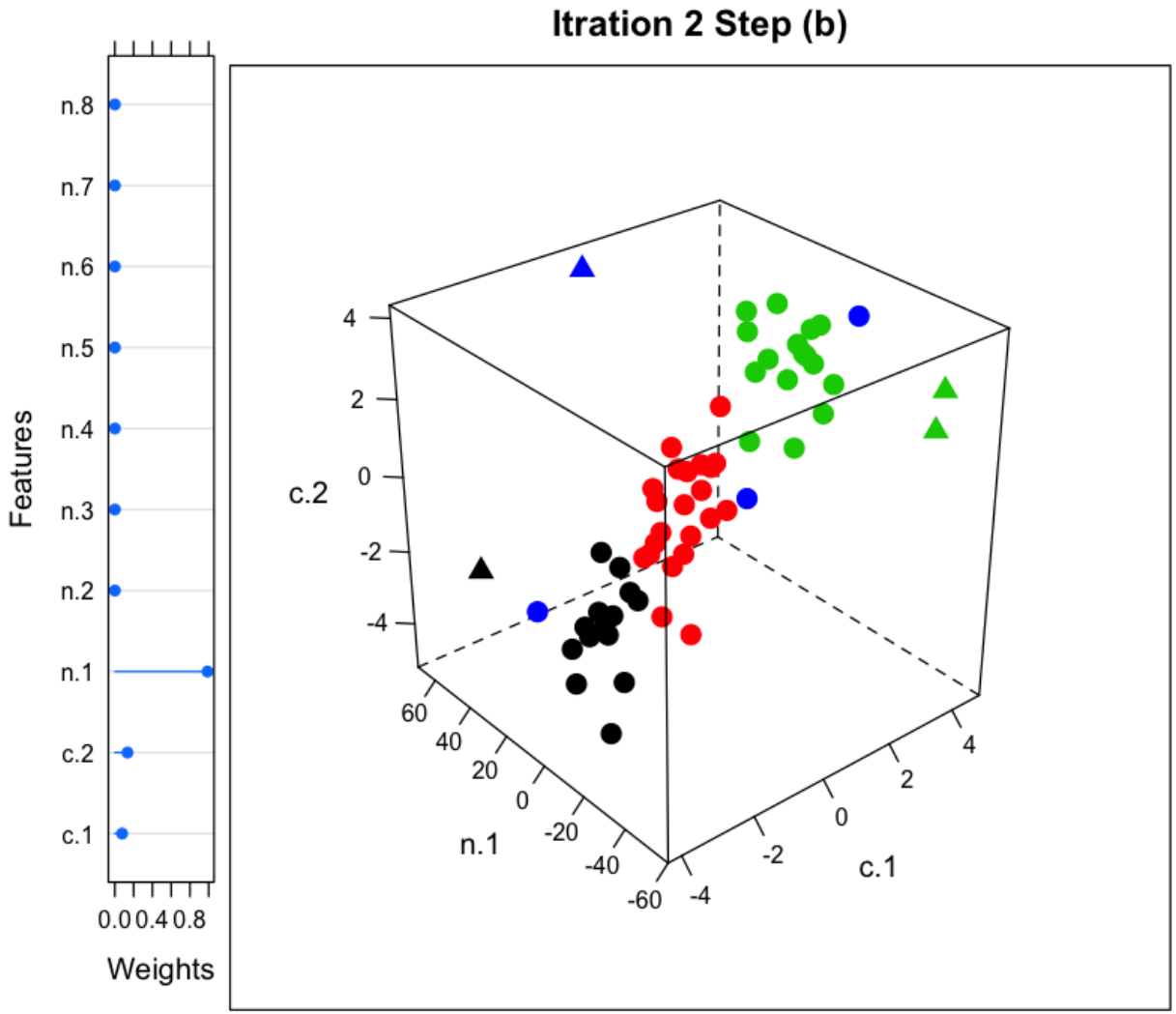

c.2
c.1
4
2
0
-2
-4
-4
-2
0
2
4

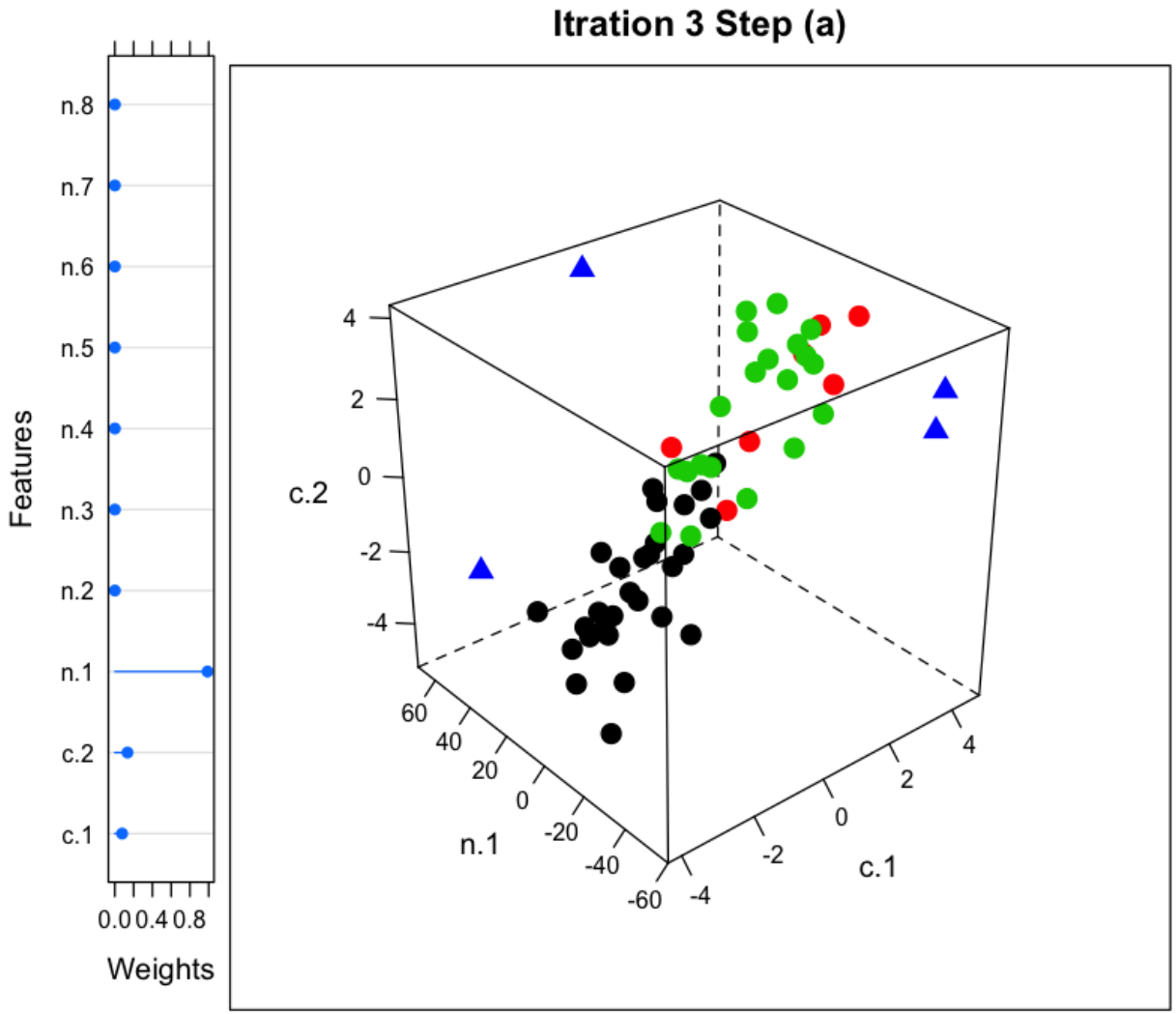

The true cluster labels
c.2
4
2
0
-2
-4
n.1
60
40
20
0
-20
-40
-60
c.1
-4
-2
0
2
4

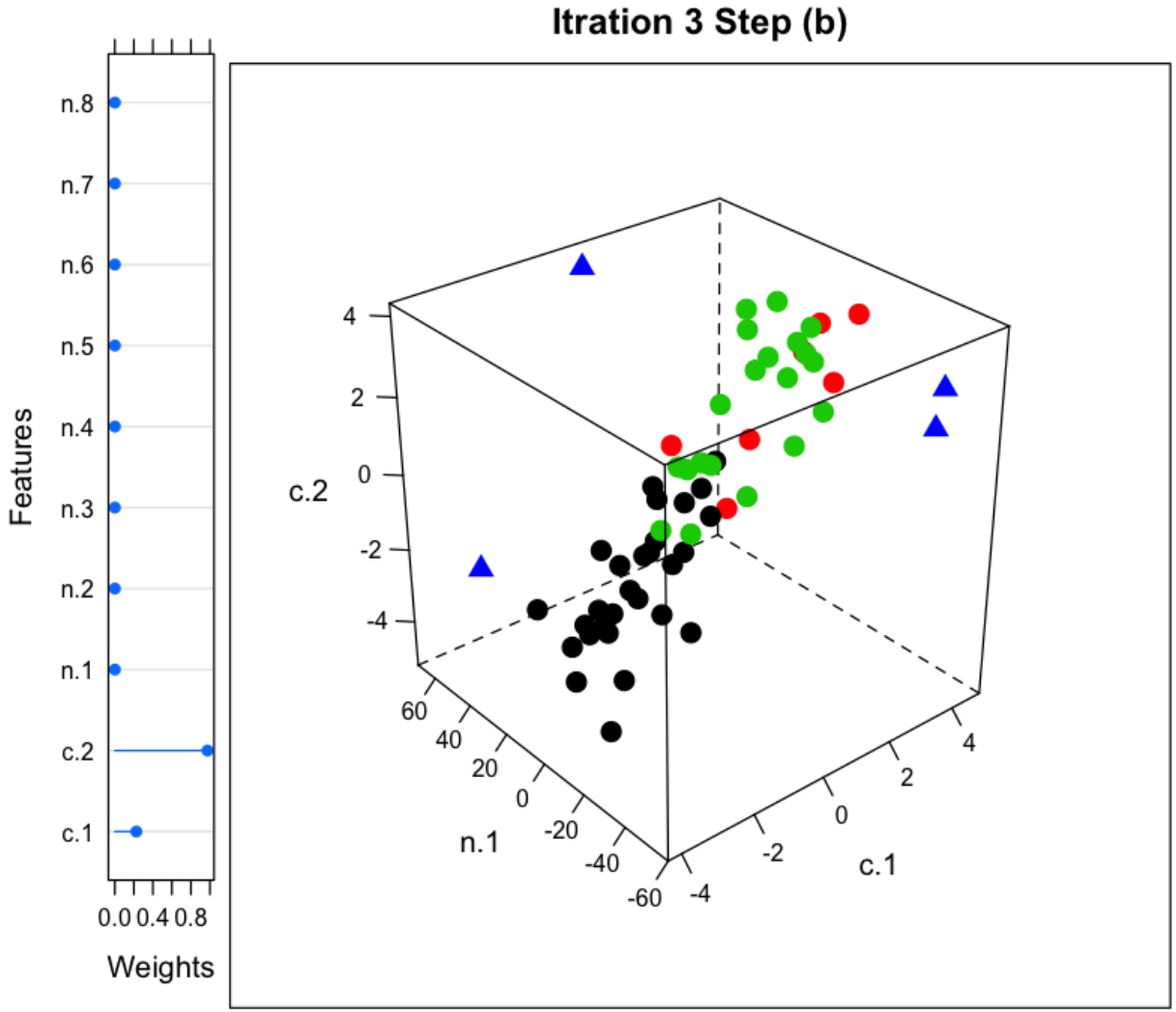

Itration 1 Step (a)
Features
n.8
n.7
n.6
n.5
n.4
n.3
n.2
n.1
c.2
c.1
0.6 1.0 1.4
Weights
c.2
4
2
0
-2
-4
n.1
60
40
20
0
-20
-40
-60
c.1
4
2
0
-2
-4

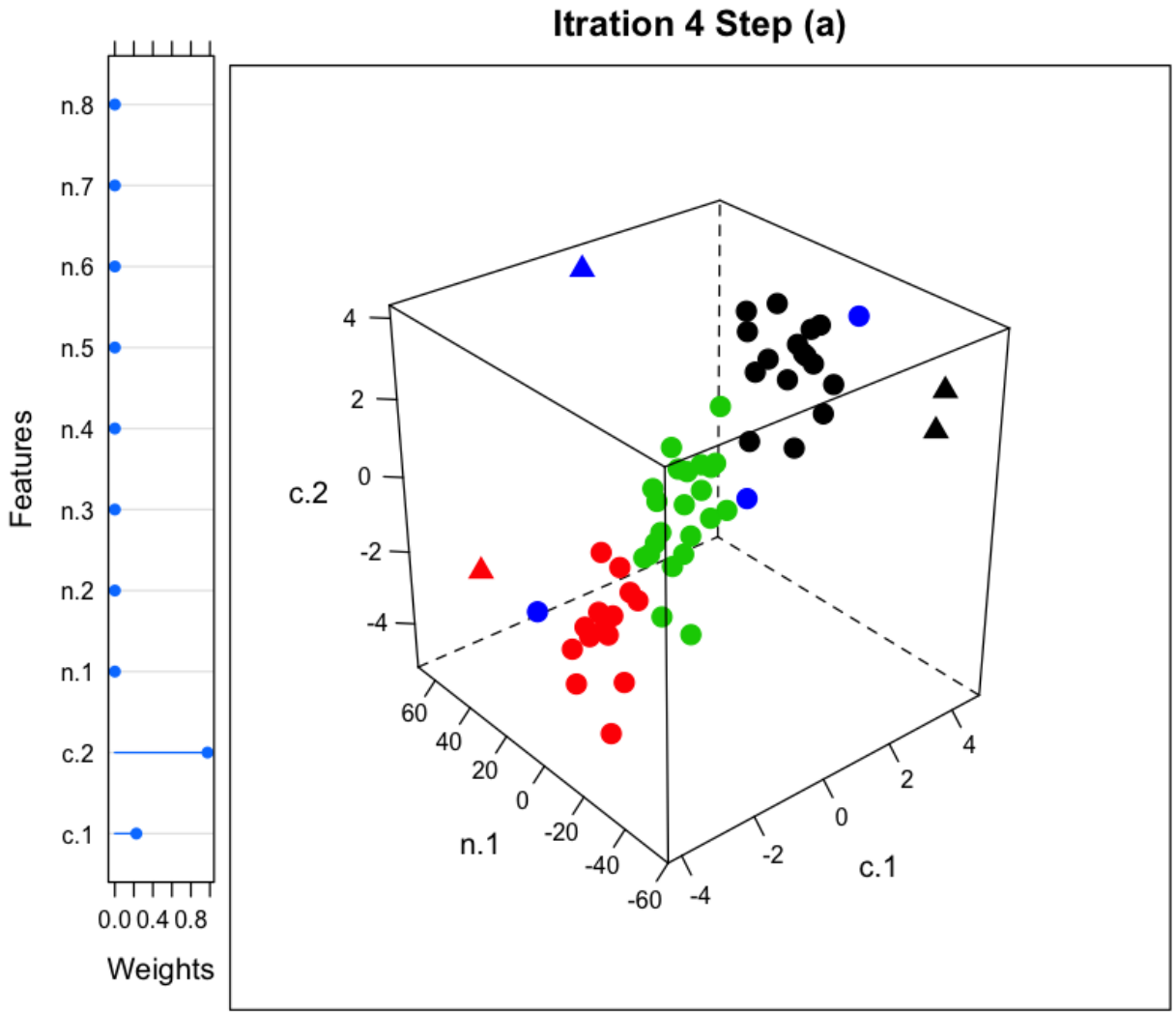

Itration 1 Step (b)
Features
n.8
n.7
n.6
n.5
n.4
n.3
n.2
n.1
c.2
c.1
0.0 0.4 0.8
Weights
c.2
4
2
0
-2
-4
n.1
60
40
20
0
-20
-40
-60
c.1
4
2
0
-2
-4

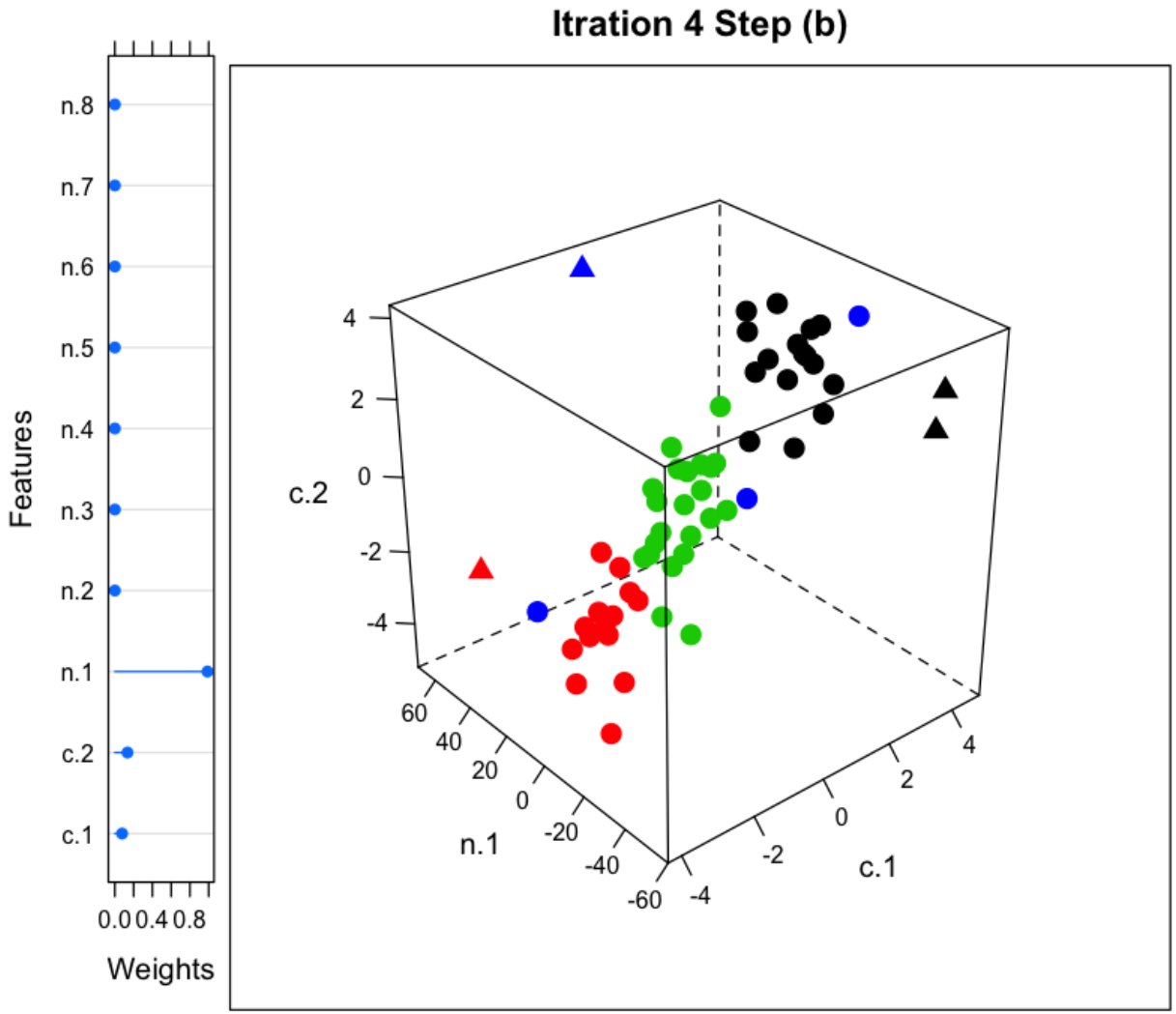

Itration 2 Step (a)
Features
n.8
n.7
n.6
n.5
n.4
n.3
n.2
n.1
c.2
c.1
0.0 0.4 0.8
Weights
c.2
4
2
0
-2
-4
n.1
60
40
20
0
-20
-40
-60
c.1
4
2
0
-2
-4

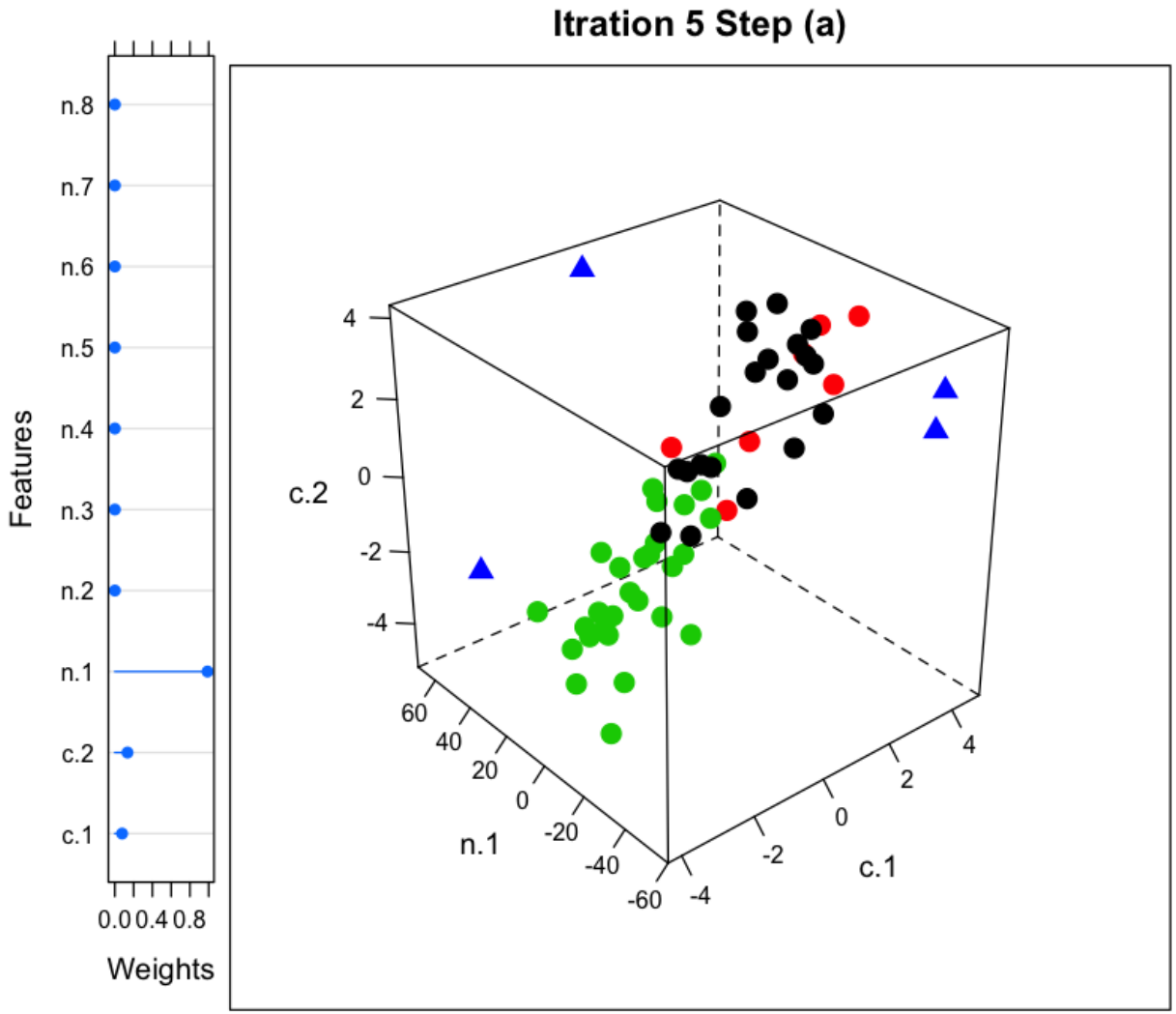

Itration 2 Step (b)
Features
n.8
n.7
n.6
n.5
n.4
n.3
n.2
n.1
c.2
c.1
0.0 0.4 0.8
Weights
c.2
4
2
0
-2
-4
n.1
60
40
20
0
-20
-40
-60
c.1
-4
-2
0
2
4

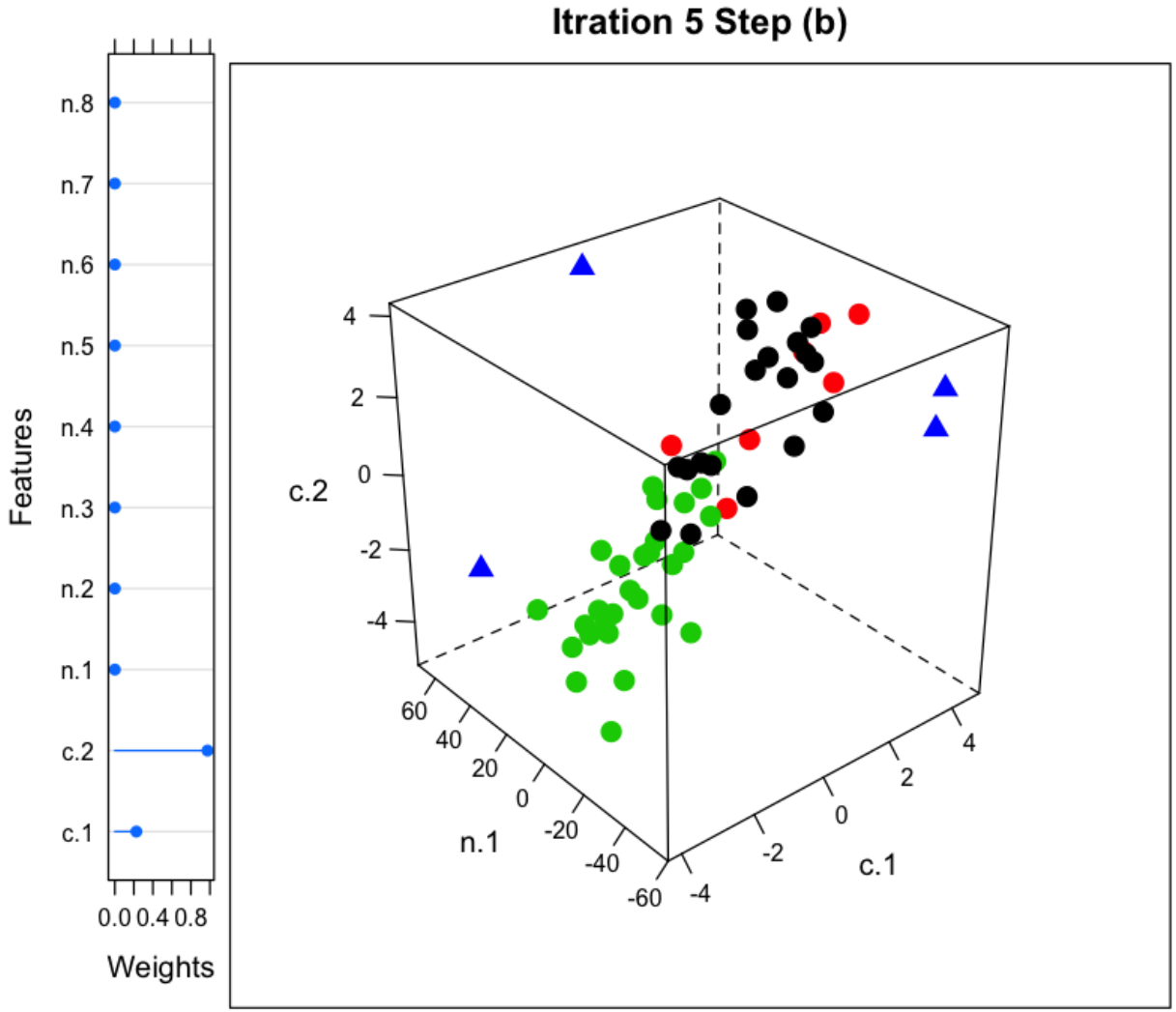

Itration 3 Step (a)
Features
n.8
n.7
n.6
n.5
n.4
n.3
n.2
n.1
c.2
c.1
0.0 0.4 0.8
Weights
c.2
4
2
0
-2
-4
n.1
60
40
20
0
-20
-40
-60
c.1
4
2
0
-2
-4

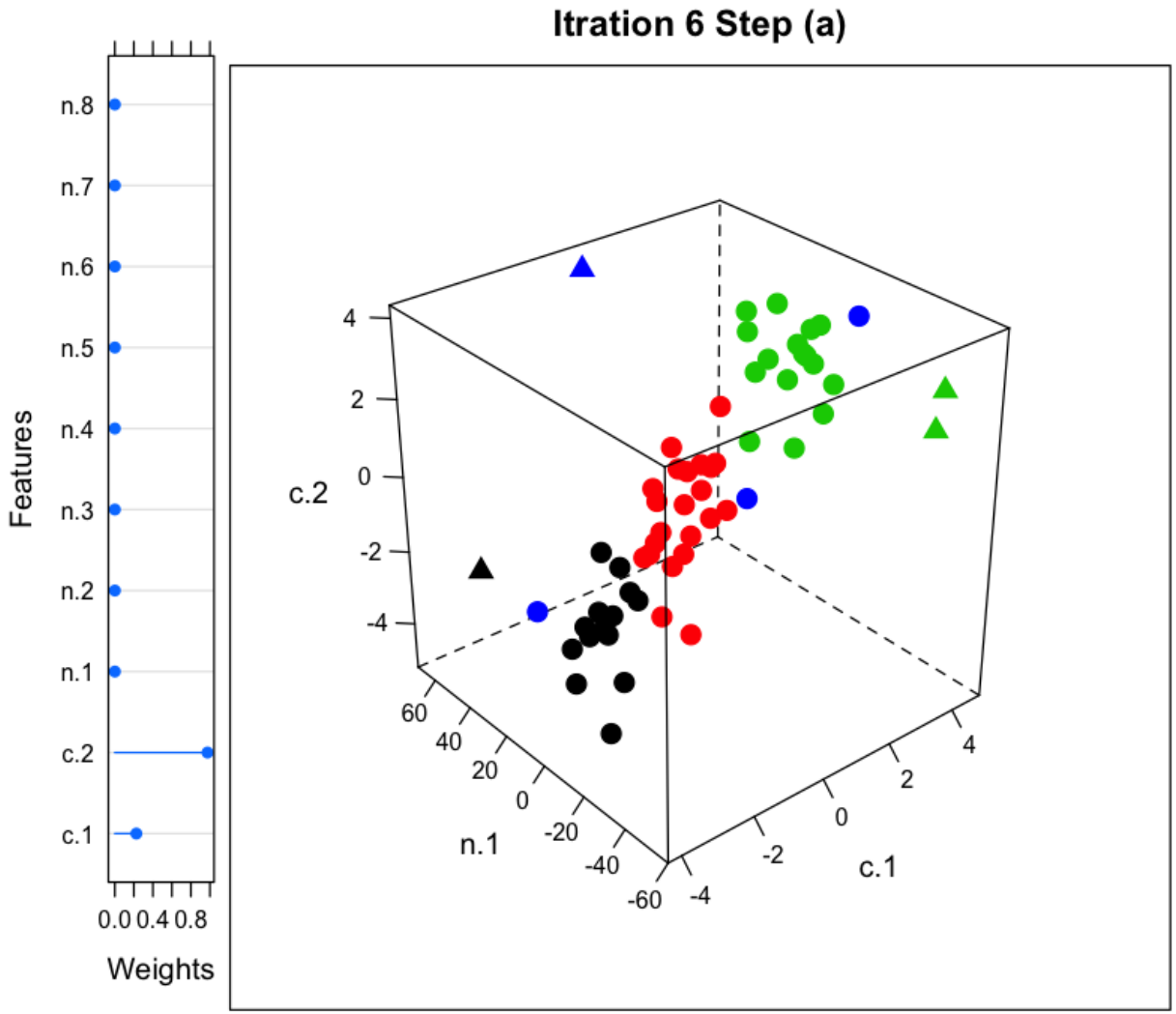

Itration 3 Step (b)
Features
n.8
n.7
n.6
n.5
n.4
n.3
n.2
n.1
c.2
c.1
0.0 0.4 0.8
Weights
c.2
4
2
0
-2
-4
n.1
60
40
20
0
-20
-40
-60
c.1
4
2
0
-2
-4

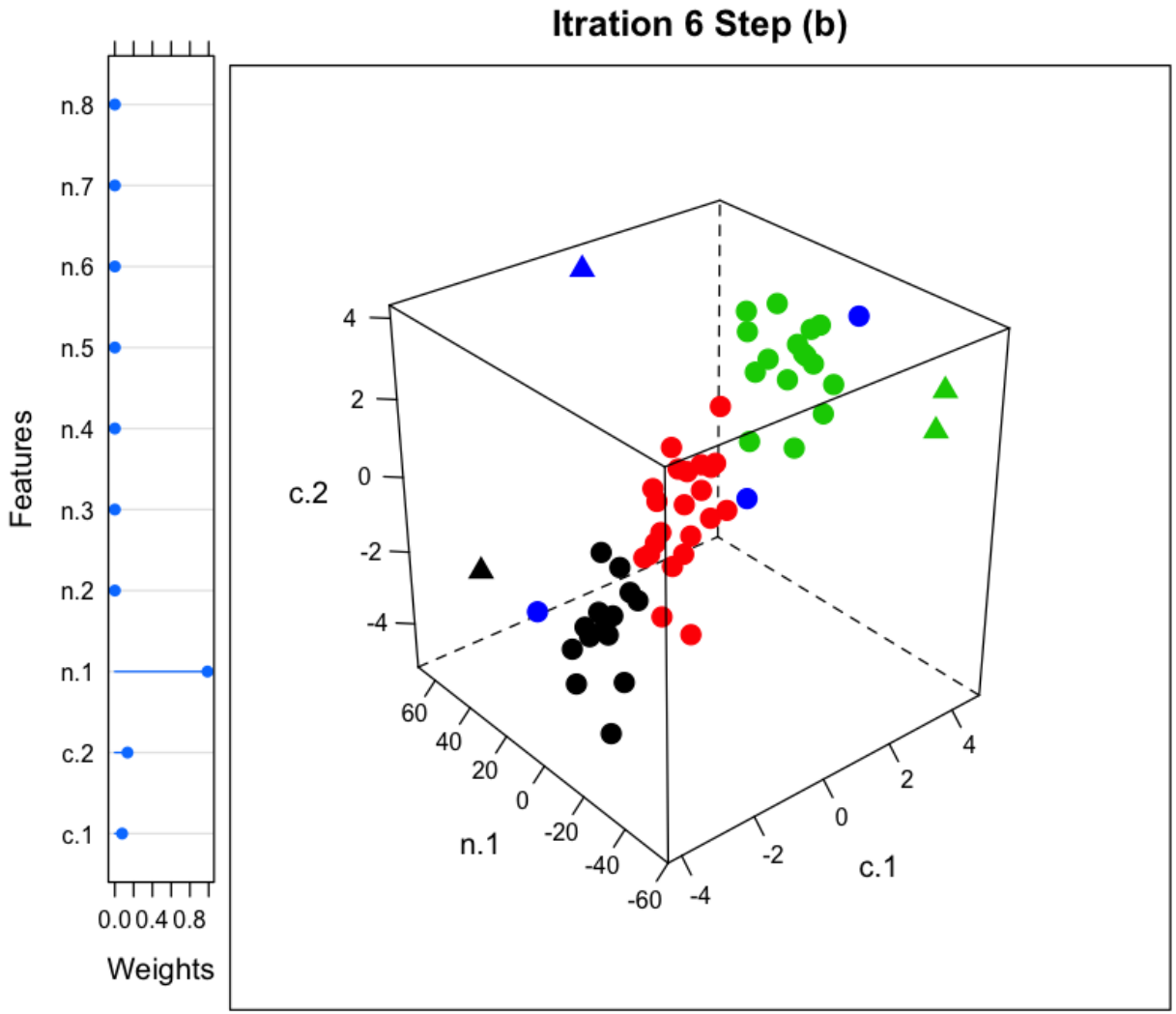

Itration 4 Step (a)
Features
n.8
n.7
n.6
n.5
n.4
n.3
n.2
n.1
c.2
c.1
0.0 0.4 0.8
Weights
c.2
4
2
0
-2
-4
n.1
60
40
20
0
-20
-40
-60
c.1
-4
-2
0
2
4


Itration 4 Step (b)
Features
n.8
n.7
n.6
n.5
n.4
n.3
n.2
n.1
c.2
c.1
0.0 0.4 0.8
Weights
c.2
4
2
0
-2
-4
n.1
60
40
20
0
-20
-40
-60
c.1
-4
-2
0
2
4


Itration 5 Step (a)
Features
n.8
n.7
n.6
n.5
n.4
n.3
n.2
n.1
c.2
c.1
0.0 0.4 0.8
Weights
c.2
4
2
0
-2
-4
n.1
60
40
20
0
-20
-40
-60
c.1
4
2
0
-2
-4


Itration 5 Step (b)
Features
n.8
n.7
n.6
n.5
n.4
n.3
n.2
n.1
c.2
c.1
0.0 0.4 0.8
Weights
c.2
4
2
0
-2
-4
n.1
60
40
20
0
-20
-40
-60
c.1
4
2
0
-2
-4


Itration 6 Step (a)
Features
n.8
n.7
n.6
n.5
n.4
n.3
n.2
n.1
c.2
c.1
0.0 0.4 0.8
Weights
c.2
4
2
0
-2
-4
n.1
60
40
20
0
-20
-40
-60
c.1
4
2
0
-2
-4


Itration 6 Step (b)
Features
n.8
n.7
n.6
n.5
n.4
n.3
n.2
n.1
c.2
c.1
0.0 0.4 0.8
Weights
c.2
n.1
c.1

The proposed robust sparse K-means algorithm

Idea

The cases are trimmed in weighted squared Euclidean distances and squared Euclidean distances.

- Robust Step(a):
  given $\boldsymbol{W}$, perform trimmed K-means on a transformed dataset
  return $\mathcal{C}$, trimmed cases.
- Robust Step(a-2):
  given $\mathcal{C}$, trim cases with the furthest distances to cluster centers in squared Euclidean distances.
- Robust Step(b):
  given $\mathcal{C}$ and the trimmed cases, maximize weighted BSS, without trimmed cases.

Repeat the steps until the stopping rule is satisfied.

The performance of robust sparse K-means on a contaminated dataset

- $N = 40$,
- $p = 1000$: 2 clustering features and 998 noise features,
- 4 outliers are added in clustering features and noise features.

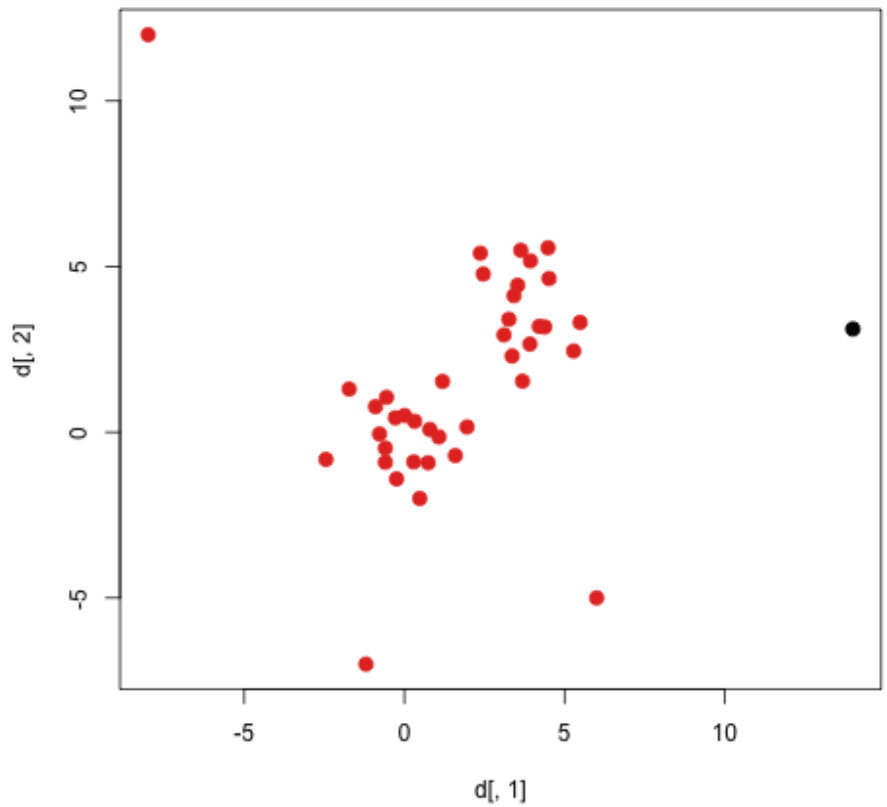


Figure: The partition from sparse 2-means

The performance of robust sparse K-means on a contaminated dataset

- $N = 40$,
- $p = 1000$: 2 clustering features and 998 noise features,
- 4 outliers are added in clustering features and noise features.

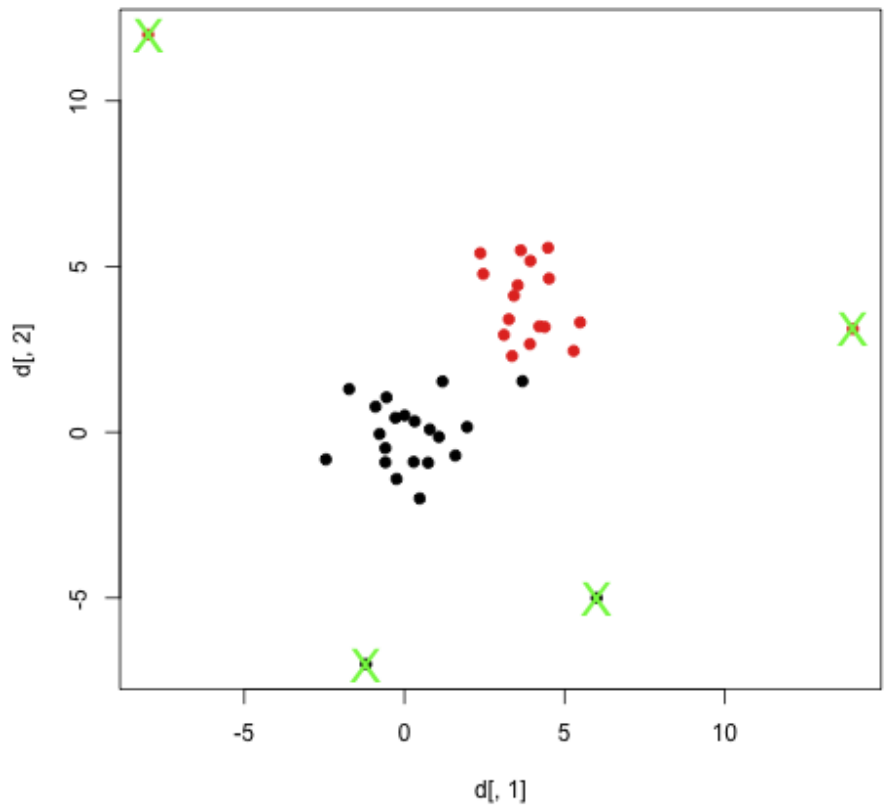


Figure: The partition from robust sparse 2-means

The simulation studies of robust sparse K-means and other clustering methods

Datasets generated from the simulation model

A dataset generated from the simulation model, containing 60 cases $\boldsymbol{x}_i = [x_{i1}, \cdots, x_{i500}]^T$ ($i = 1, \cdots, 60$). The cases are generated from 3 clusters. Note that the first 50 features are clustering features and the rest of 450 features are noise features. That is:

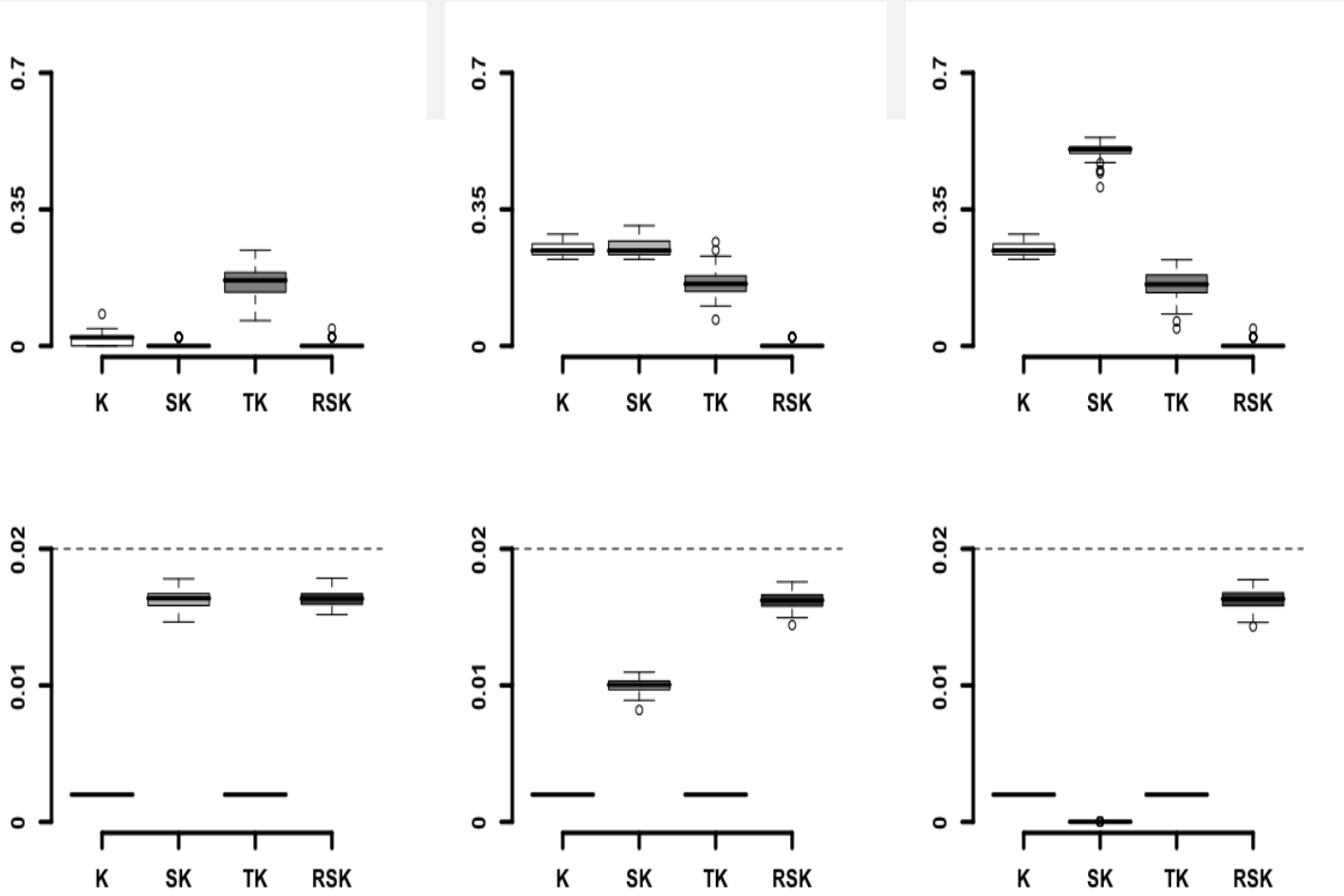
- for $j = 1, \cdots, 50$, $x_{ij} = \mu_i + \epsilon_{ij}$, with $\epsilon_{ij} \overset{i.i.d}{\sim} N(0, 1)$ where:

$$\mu_i = \begin{cases} \mu & \text{if } 1 \leq i \leq 20, \\ 0 & \text{if } 21 \leq i \leq 40, \\ -\mu & \text{if } 41 \leq i \leq 60. \end{cases}$$

- for $j = 51, \cdots, 500$ and $i = 1, \cdots, 50$, $x_{ij} = \epsilon_{ij}$.

Model 1:

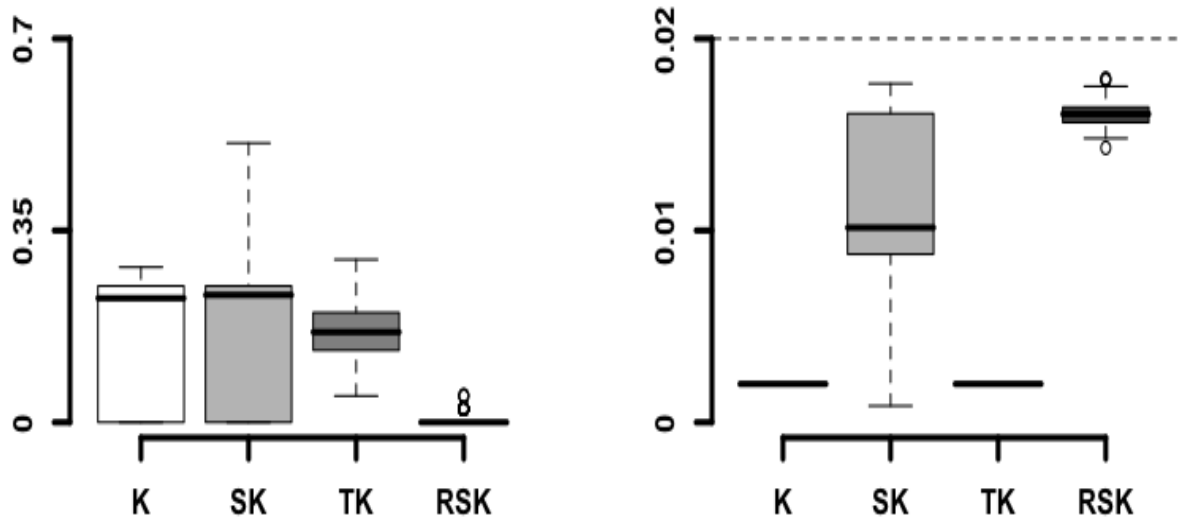
- 100 datasets are generated from the simulation model ($\mu = 1$).
- The value of a single noise feature for a single case, $x_{1,500}$, is replaced by the value, *out* for each dataset.



out=15 *out*=25 *out*=500

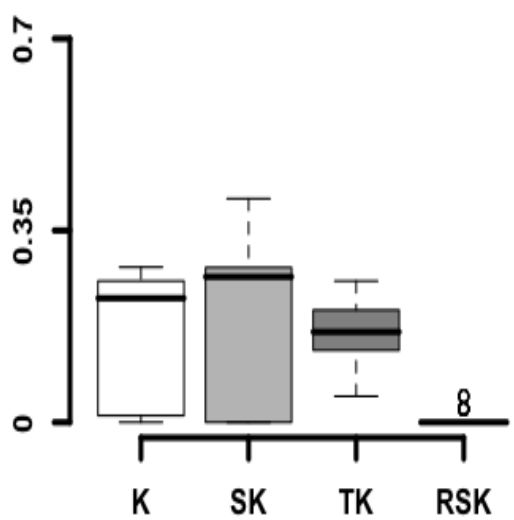
Model 2:

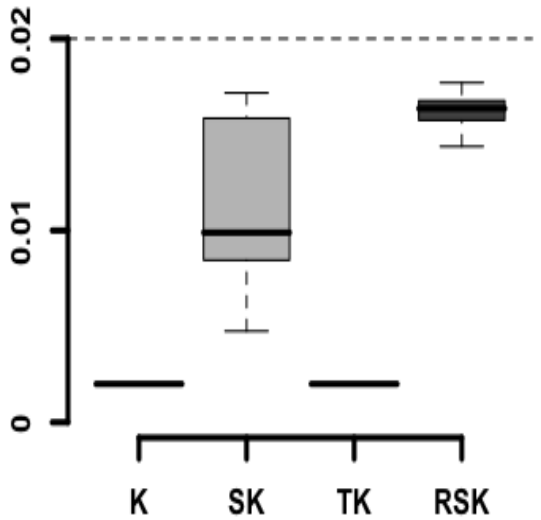
- 100 datasets are generated from simulation model ($\mu = 1$).
- Two cases from each cluster are contaminated in a single noise feature by adding a noise with large variance. In total 6 cases (10% of cases are contaminated) and 6 noise features are contaminated. Specifically, $x_{ij} \sim N(0, 15^2)$ for (i, j)=$(1, 51)$, $(2, 52)$, $(21, 53)$, $(22, 54)$, $(41, 55)$, $(42, 56)$.



Model 3:

- 100 datasets are generated from simulation model ($\mu = 1$).
- Two cases from each cluster are contaminated in a single clustering feature by adding a noise with large variance. In total 6 cases (10% of cases are contaminated) and 6 clustering features are contaminated. Specifically, $x_{ij} \sim N(0, 15^2)$ for (i, j)=$(3, 1)$, $(4, 2)$, $(23, 3)$, $(24, 4)$, $(43, 5)$, $(44, 6)$.





The robust sparse K-means with missing values

- Let $MF(\boldsymbol{x})$ denote the set of feature indices which do not have missing values in the vector $\boldsymbol{x}$.
- The k^{th} cluster center along the j^{th} feature $\bar{x}_{k,j}$ is calculated without missing values.

All the squared Euclidean distance between $\boldsymbol{x}_i$ and $\bar{\boldsymbol{x}}_k$ are calculated as:

$$d_{AE}(\boldsymbol{x}_i, \bar{\boldsymbol{x}}_k) = \frac{p}{|MF(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)|} \sum_{j \in MF(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)} (x_{ij} - \bar{x}_{k,j})^2.$$

All the weighted squared Euclidean distance between $\boldsymbol{x}_i$ and $\bar{\boldsymbol{x}}_k$ are calculated as:

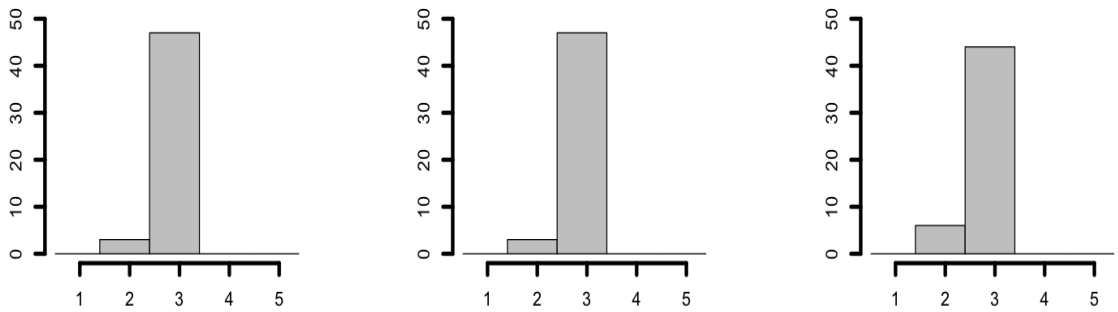
$$d_{AW}(\boldsymbol{x}_i, \bar{\boldsymbol{x}}_k) = S_i \sum_{j \in MF(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)} w_j (x_{ij} - \bar{x}_{k,j})^2,$$

where:

$$S_i = \frac{\sum_{j=1}^{p} w_j}{\sum_{j \in MF(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)} w_j}.$$

Clest with robust sparse *K*-means

We adapted Clest (S. Dudoit and J.Fridlyand, 2002), prediction based resampling method, to robust sparse *K*-means.

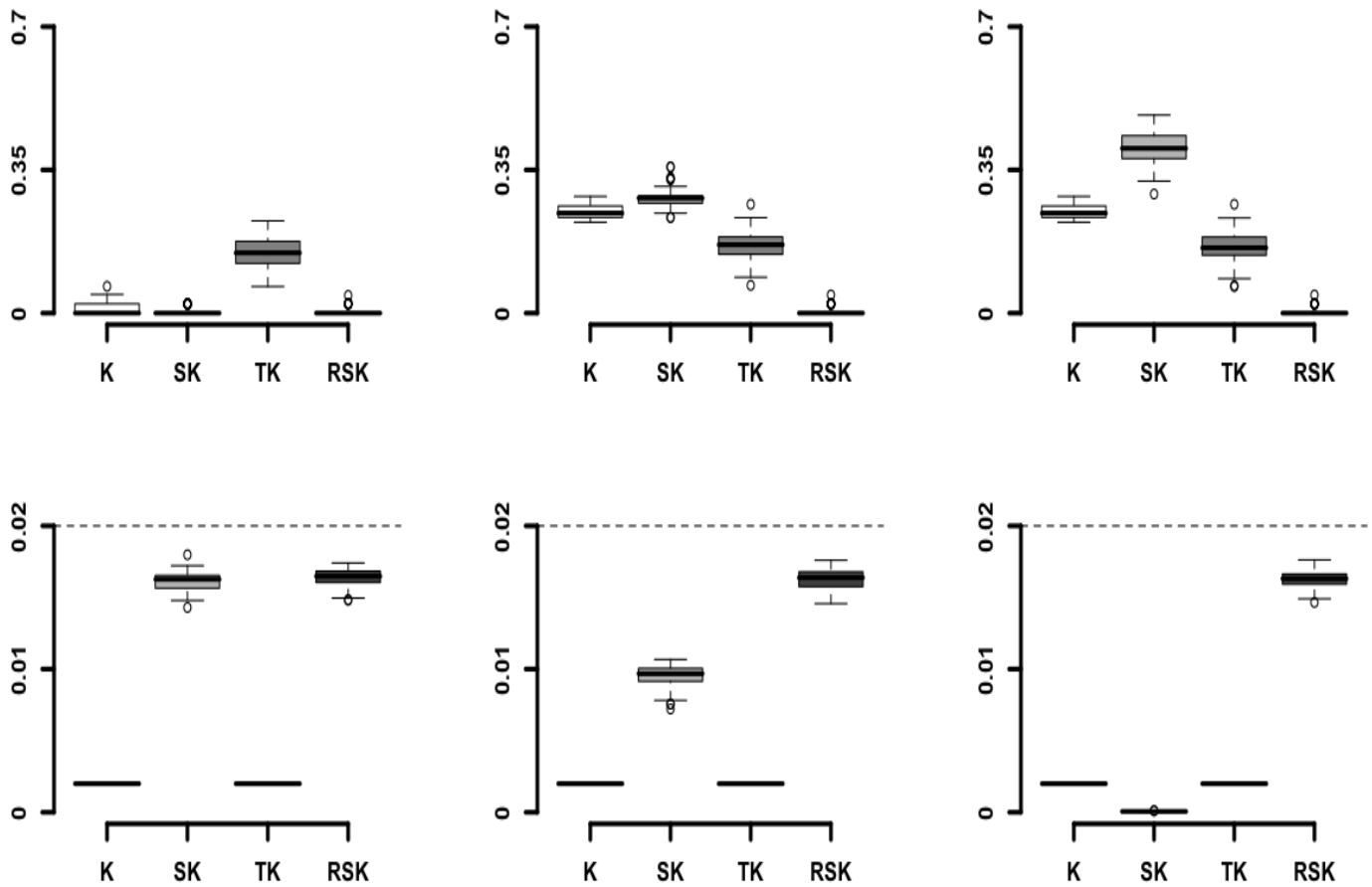


Model 1 *out*=500 Model 2 Model 3

Figure: The results of Clest on datasets generated from the three models with $\mu = 2$. The histogram of the estimated number of clusters from Clest over 50 generated datasets for each model.

`R` package `RSKC` is publicly available.
Thank you!

The value of a single clustering feature for a single case, $x_{1,1}$, is replaced by the value, *out*.



out=15 *out*=25 *out*=500

Figure: Model 2

A single case, $\boldsymbol{x}_1$, is replaced by one generated from a multivariate normal distribution with $\boldsymbol{\mu} = (5, \cdots, 5)^T \in \mathbb{R}^{500}$ and an identity variance covariance matrix.

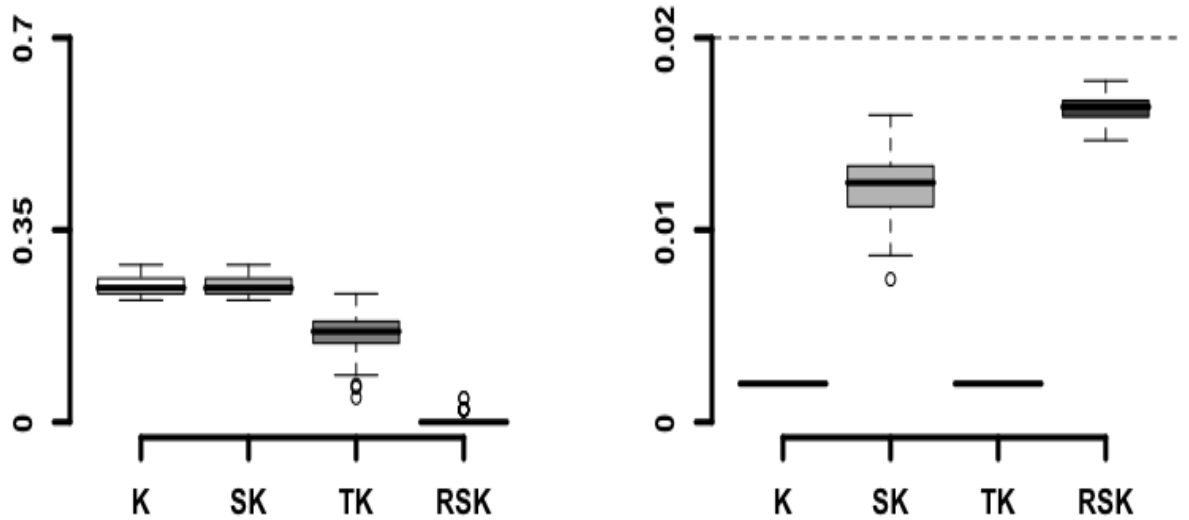


Figure: Model 3

- 100 datasets are generated from simulation model ($\mu = 1$).
- Outlier values are generated from both models 4 and 5. In total 12 cases (20% of cases are contaminated), 6 clustering features and 6 noise features are contaminated.

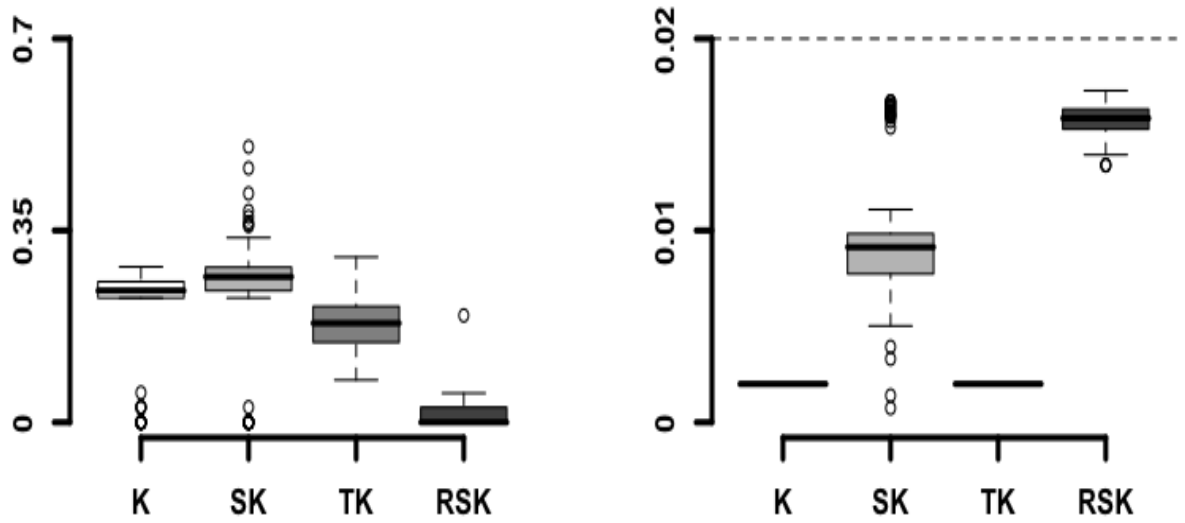


Figure: Model 6

- 100 datasets are generated from simulation model ($\mu = 1$).
- Two cases from each cluster are replaced by ones generated from a multivariate normal distribution with large variance. Specifically, $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_{21}$, $\boldsymbol{x}_{22}$, $\boldsymbol{x}_{41}$ and $\boldsymbol{x}_{42}$ are generated from $\boldsymbol{N}_{500}(\boldsymbol{0}, 5^2\boldsymbol{I})$. In total 6 cases (10% of cases are contaminated) and all the features are contaminated.

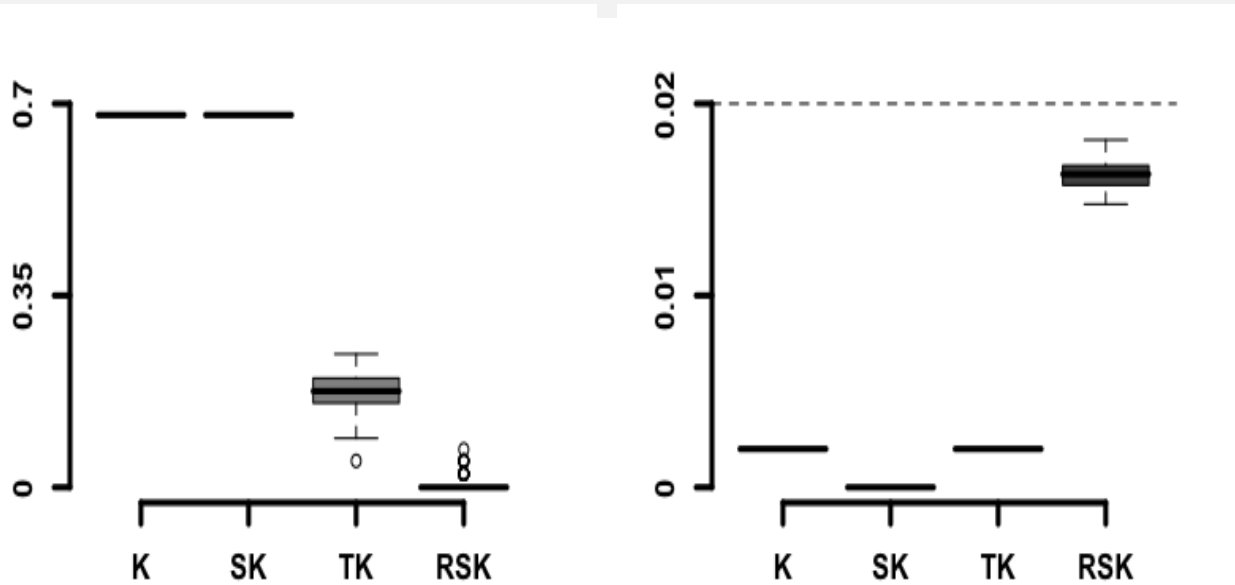


Figure: Model 7

- 100 datasets are generated from simulation model ($\mu = 1$).
- Twenty-five clustering features of the 1^{st} case are replaced with 25 clustering features of the 60^{th} case. (Outliers are 'hidden'.)

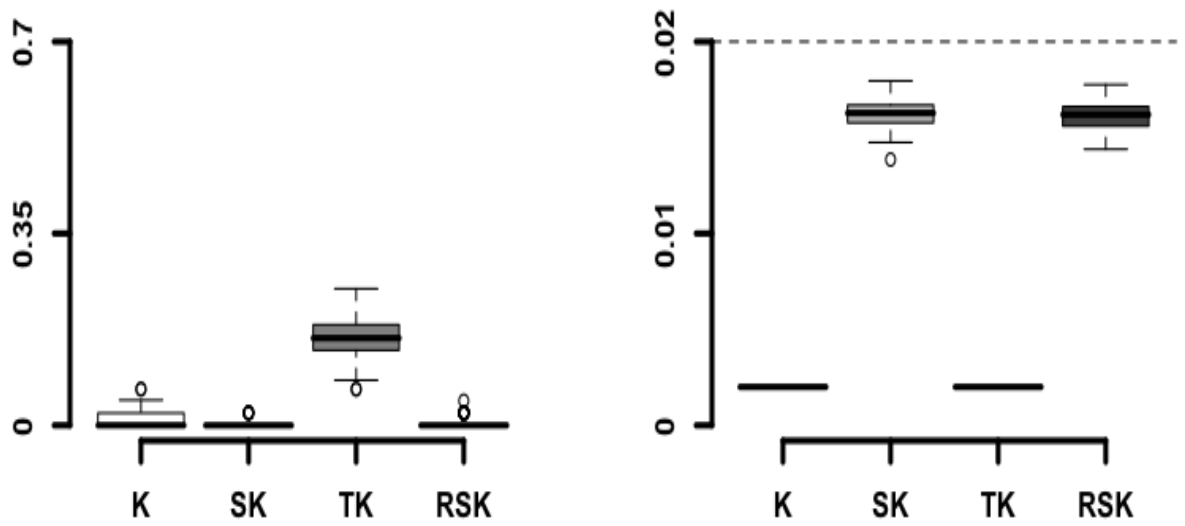
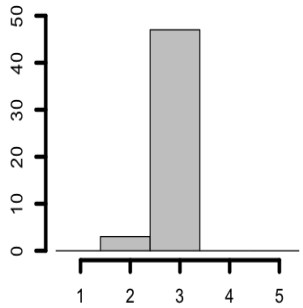
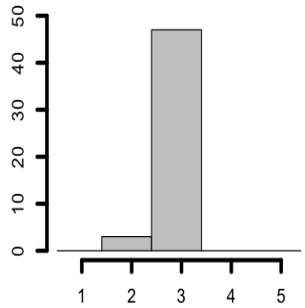


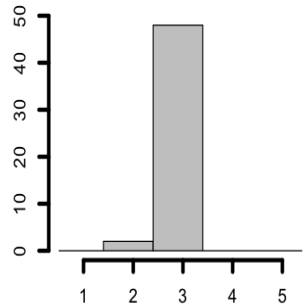
Figure: Model 8



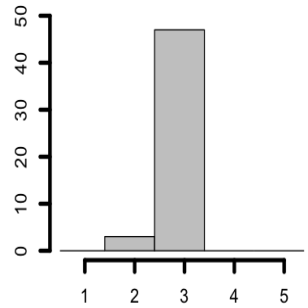
Model 1 *out*=500



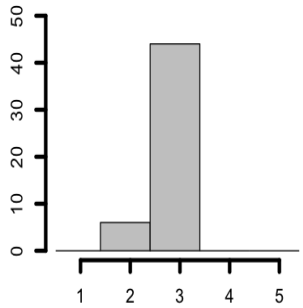
Model 2 *out*=500



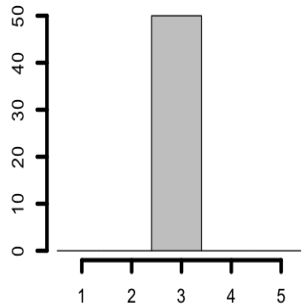
Model 3



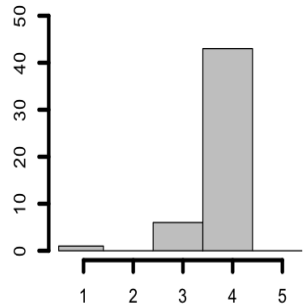
Model 4



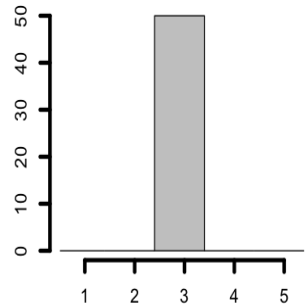
Model 5



Model 6

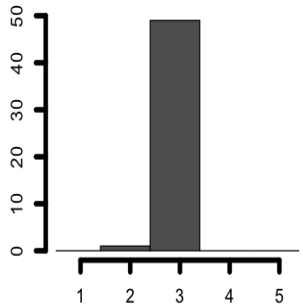


Model 7

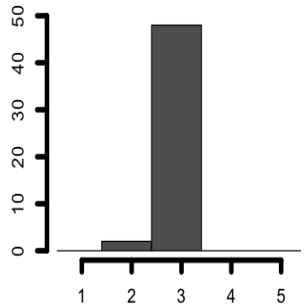


Model 8

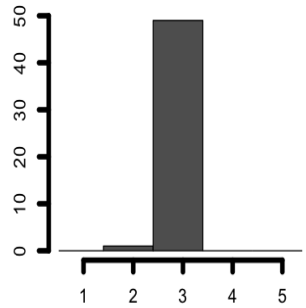
Figure: The results of Clest on datasets generated from the three models with $\mu = 2$. The histogram of the estimated number of clusters from Clest over 50 generated datasets for each model.



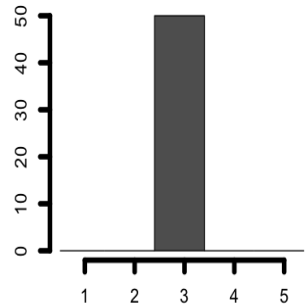
Model 1 *out*=500



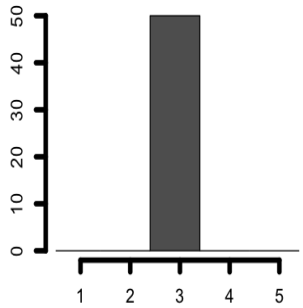
Model 2 *out*=500



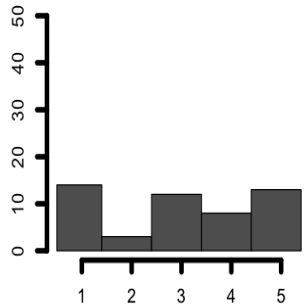
Model 3



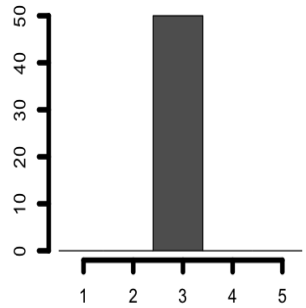
Model 4



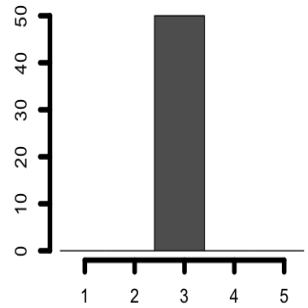
Model 5



Model 6

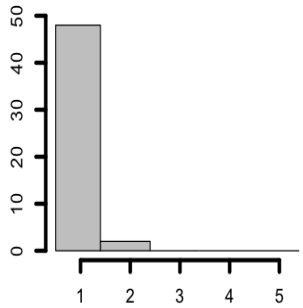


Model 7

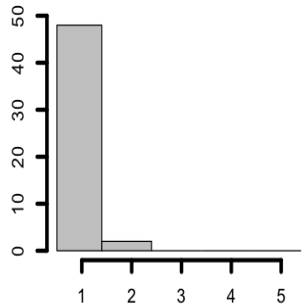


Model 8

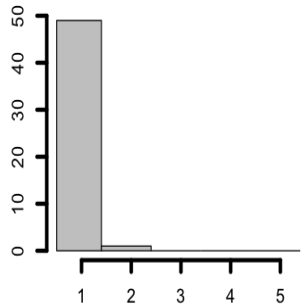
Figure: The results of Clest on datasets generated from the three models with $\mu = 2$. The histogram of the estimated number of clusters from Clest over 50 generated datasets for each model.



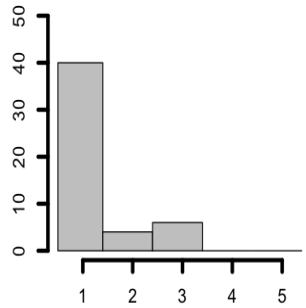
Model 1 *out*=500



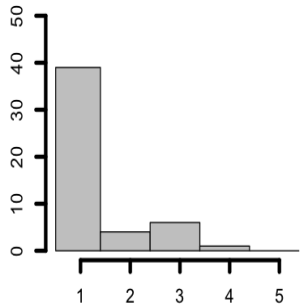
Model 2 *out*=500



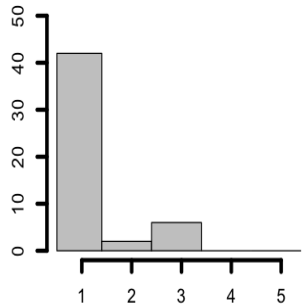
Model 3



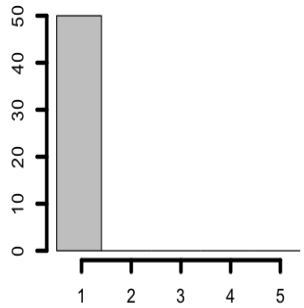
Model 4



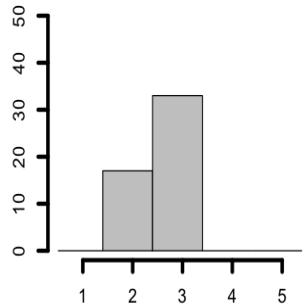
Model 5



Model 6

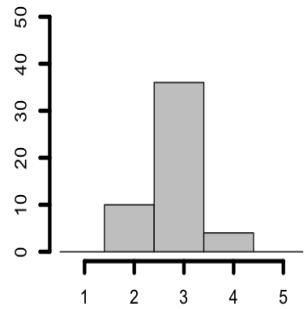


Model 7

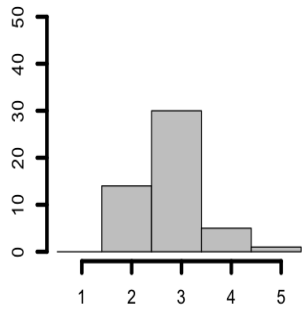


Model 8

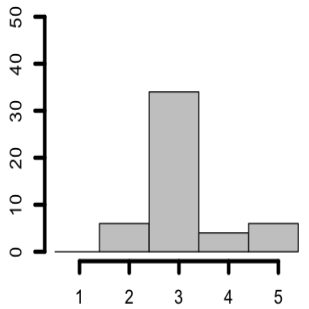
Figure: The results of Clest on datasets generated from the three models with $\mu = 1$. The histogram of the estimated number of clusters from Clest over 50 generated datasets for each model.



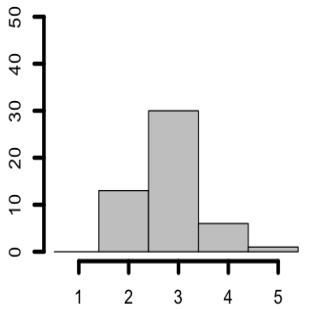
Model 1 *out*=500



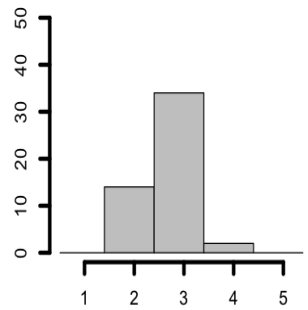
Model 2 *out*=500



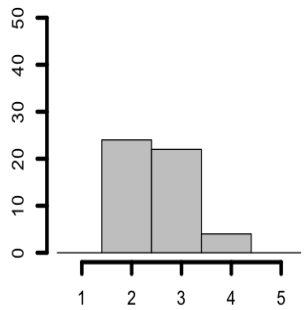
Model 3



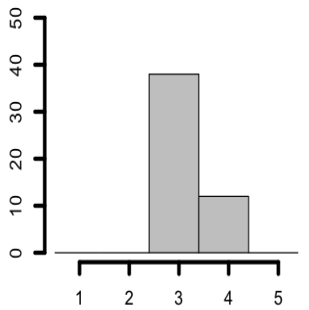
Model 4



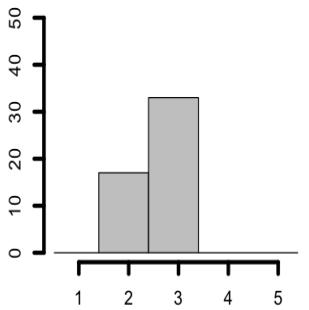
Model 5



Model 6



Model 7



Model 8

Figure: The results of Clest on datasets generated from the three models with $\mu = 1$. The histogram of the estimated number of clusters from Clest over 50 generated datasets for each model. The significance level is set to 1.