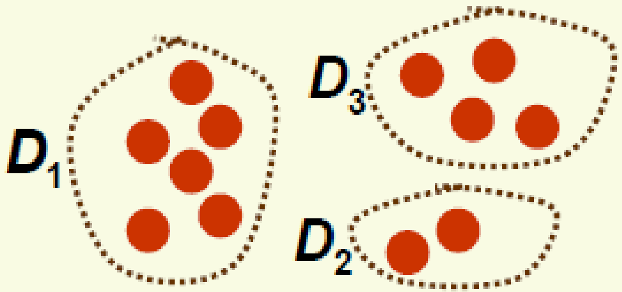
Cluster Analysis – Iterative ML Algorithms

Iterative Clustering Algorithms

Criterion Functions for Clustering

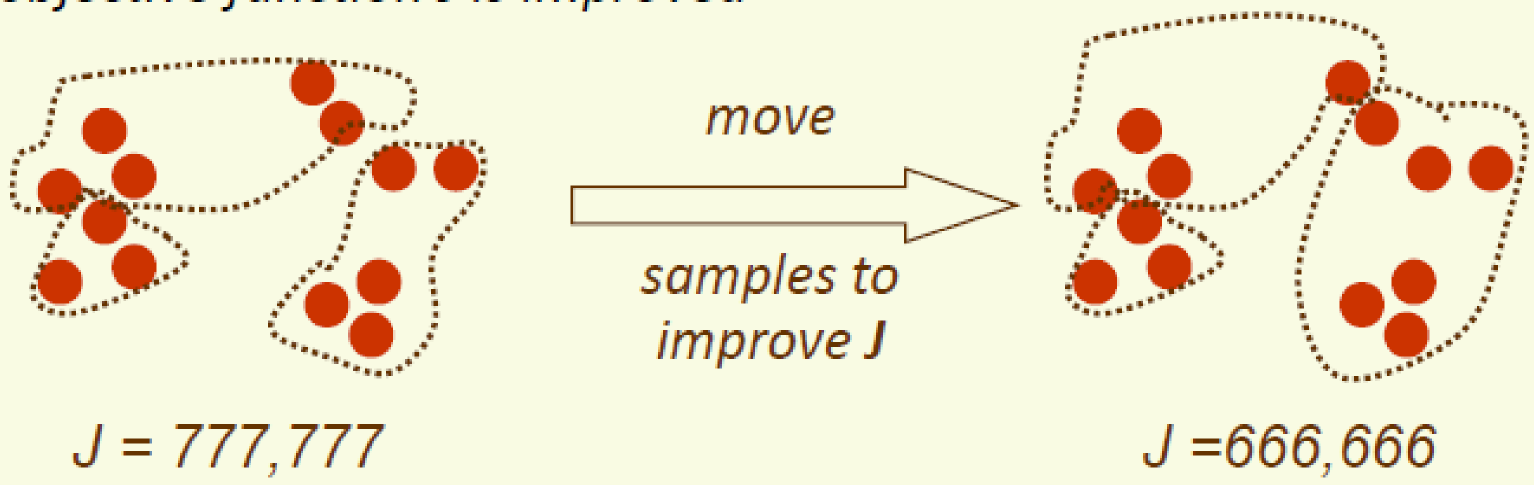
- Have samples $\mathbf{x}_1,\ldots,\mathbf{x}_n$
- Suppose partitioned samples into c subsets $D_1,\ldots,D_c$



- There are approximately $c^n/c!$ distinct partitions
- Can define a criterion function $J(D_1,\ldots,D_c)$ which measures the quality of a partitioning $D_1,\ldots,D_c$
- Then the clustering problem is well defined
  - optimal clustering is partition which optimizes criterion function

Iterative Optimization Algorithms

- Now have both proximity measure and criterion function, need algorithm to find the optimal clustering
- Exhaustive search is impossible, since there are approximately $c^n/c!$ possible partitions
- Usually some iterative algorithm is used
  1. find a reasonable initial partition
  2. repeat: *move samples from one group to another s.t. the objective function **J** is improved*



K-means Clustering

- Probably the most famous clustering algorithm
- J_{SSE} **objective** function:

$$J_{SSE} = \sum_{i=1}^{k} \sum_{x \in D_i} || \boldsymbol{x} - \mu_i ||^2$$

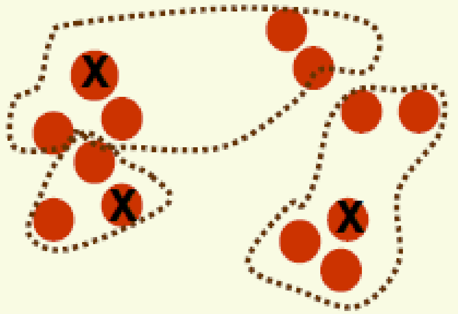
  - can work for some other objective functions, but not proven to work as well

- Fix number of clusters to $\boldsymbol{k}$ ($\boldsymbol{c} = \boldsymbol{k}$)
- Iterative clustering algorithm
- Moves many samples from one cluster to anther in a smart way

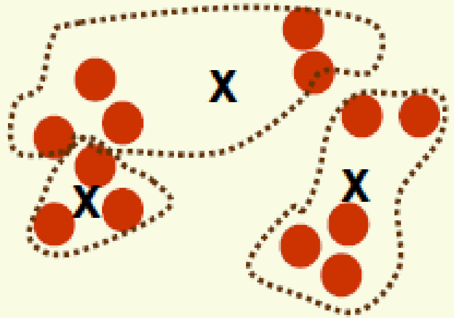
K-means Clustering

k = 3

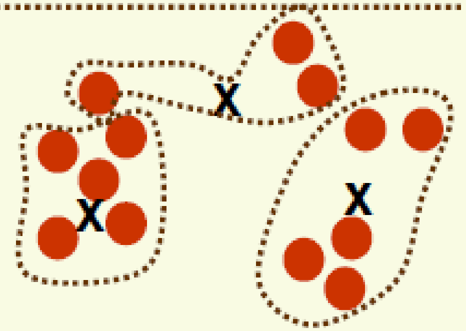
1. Initialize
   - pick ***k*** cluster centers arbitrary
   - assign each example to closest center



2. compute sample means
   for each cluster



3. reassign all samples to the
   closest mean

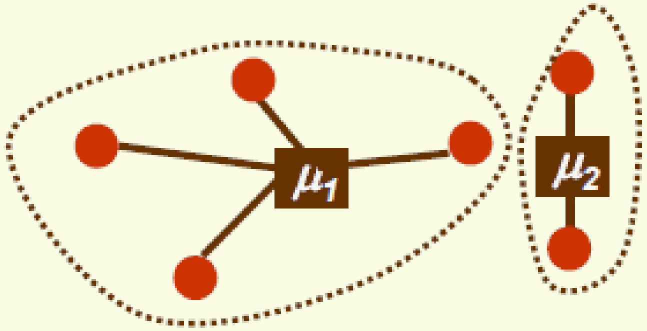


4. if clusters changed at step 3, go to step 2

K-means Clustering

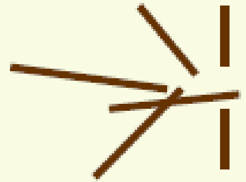
- Consider steps *2* and *3* of the algorithm

2. compute sample means for each cluster

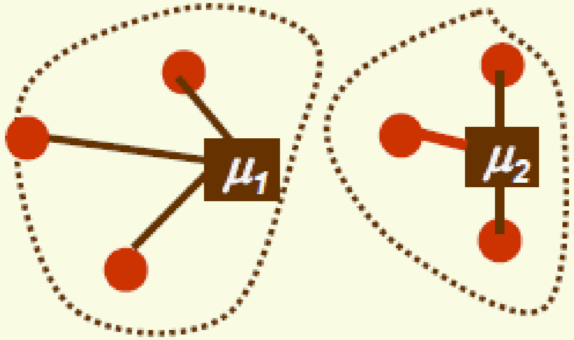


$$J_{SSE} = \sum_{i=1}^{k} \sum_{x \in D_i} \| \boldsymbol{x} - \mu_i \|^2$$

= sum of



3. reassign all samples to the closest mean



If we represent clusters by their old means, the error has gotten smaller

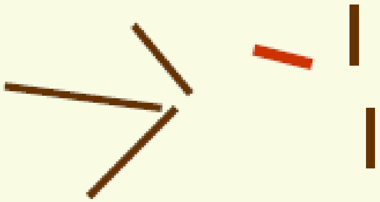
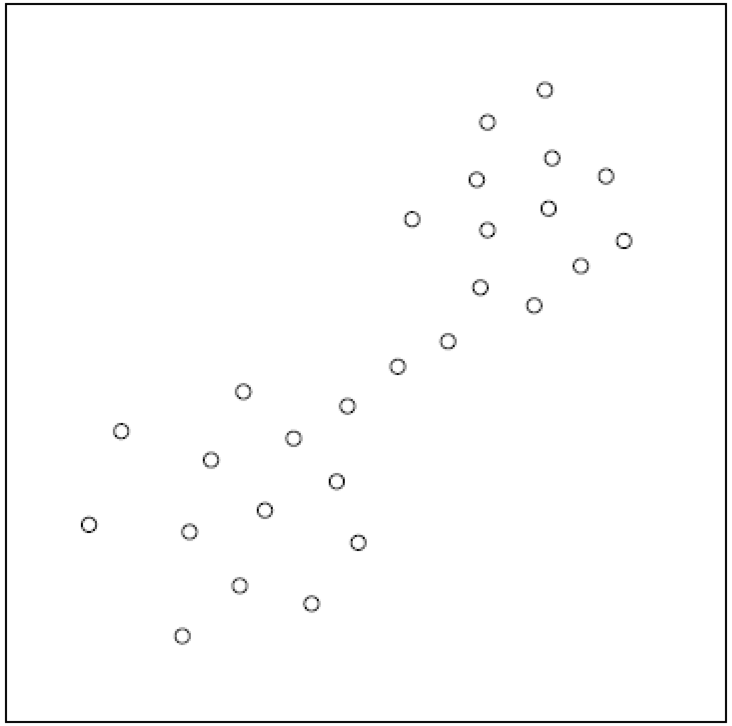


Illustration of K-means Algorithm

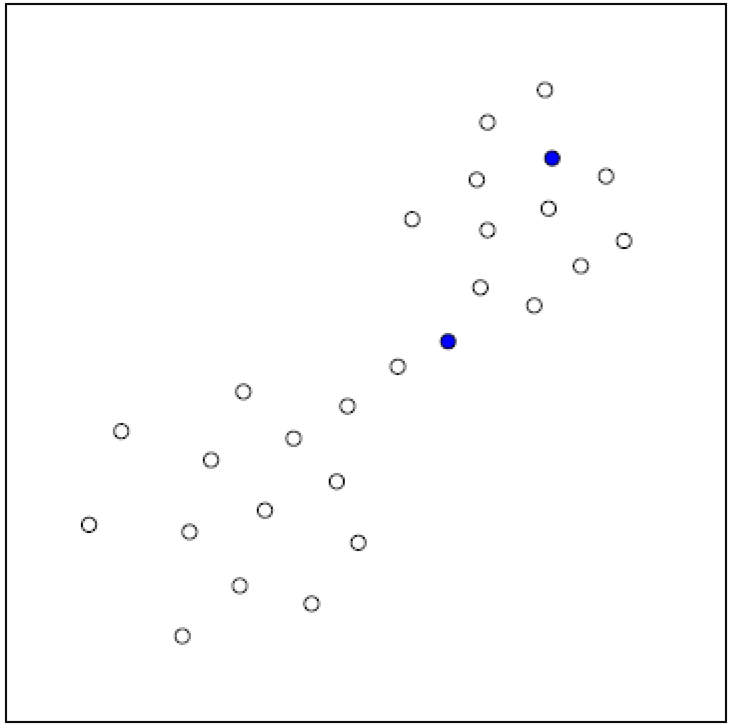
Iterative Cluster Analysis

k-Means with Minimization Criterion e = Variance



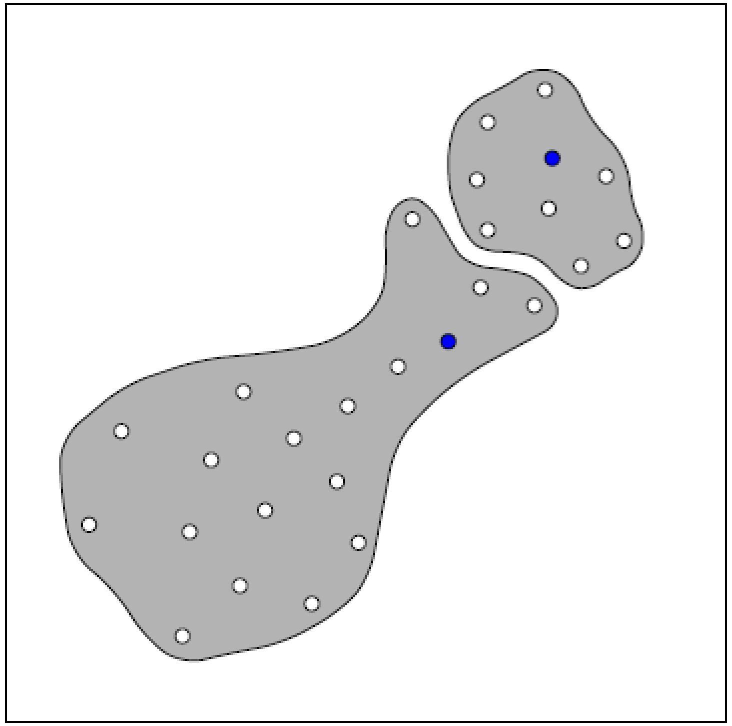
Iterative Cluster Analysis

k-Means with Minimization Criterion e = Variance



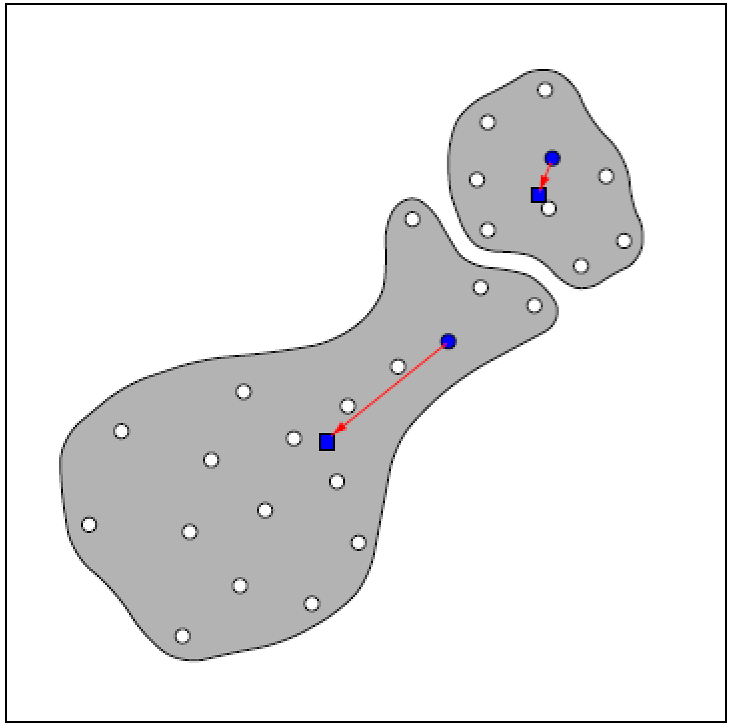
Iterative Cluster Analysis

k-Means with Minimization Criterion e = Variance



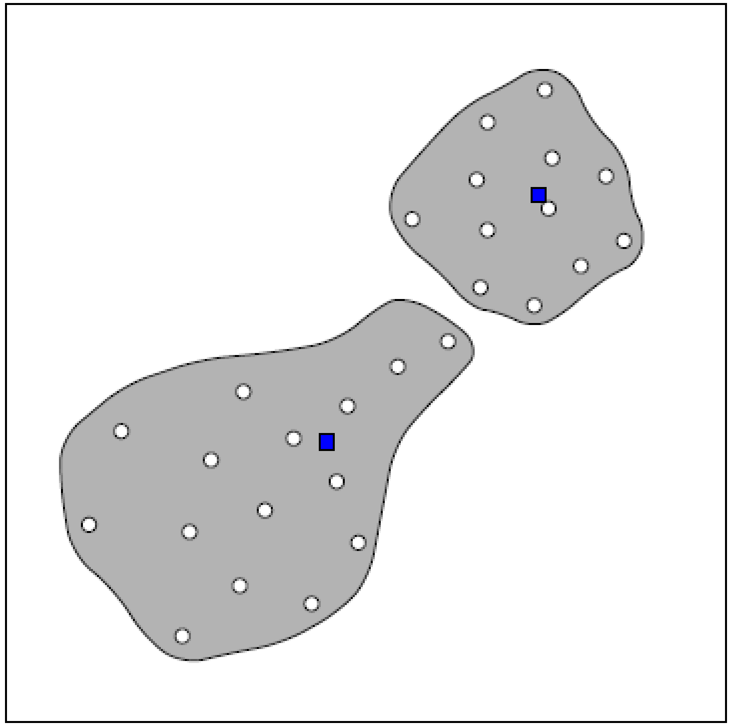
Iterative Cluster Analysis

k-Means with Minimization Criterion e = Variance



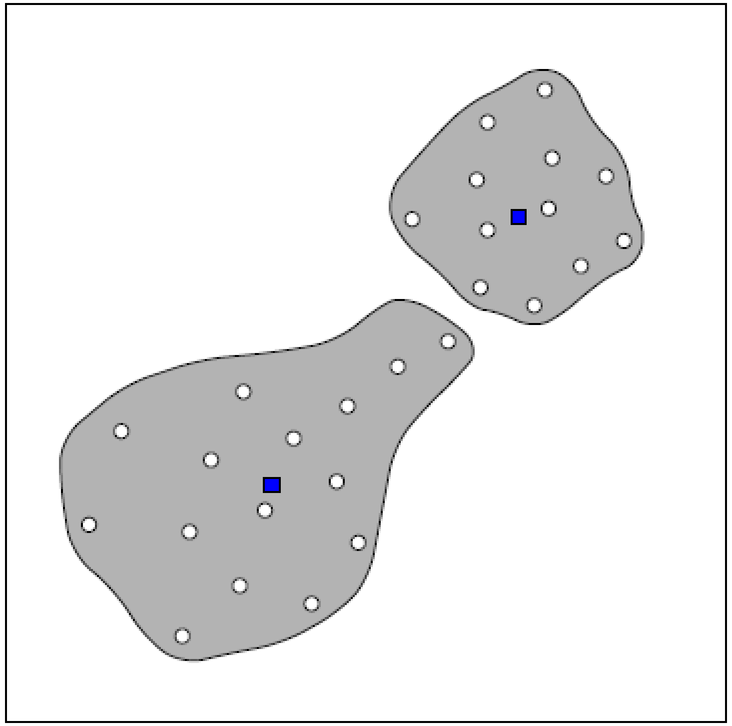
Iterative Cluster Analysis

k-Means with Minimization Criterion e = Variance



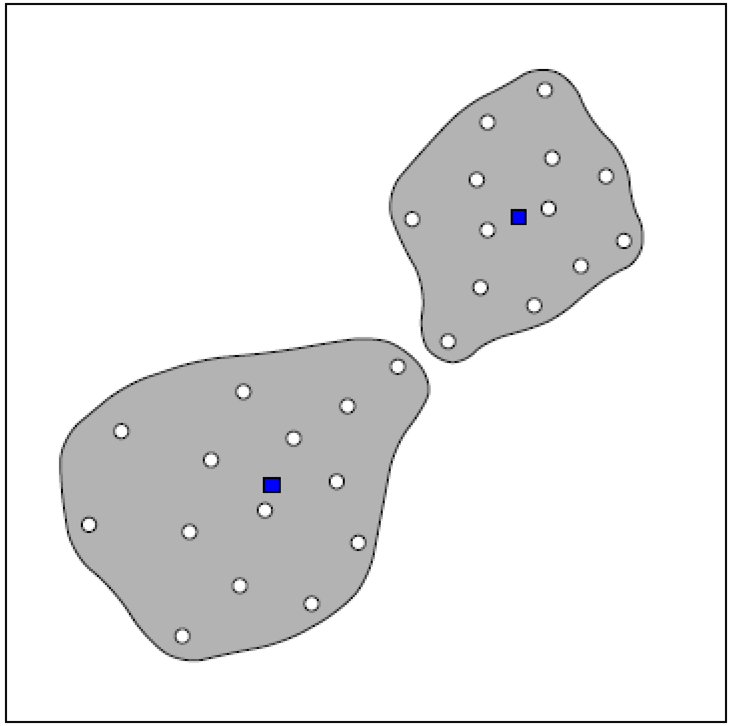
Iterative Cluster Analysis

k-Means with Minimization Criterion e = Variance



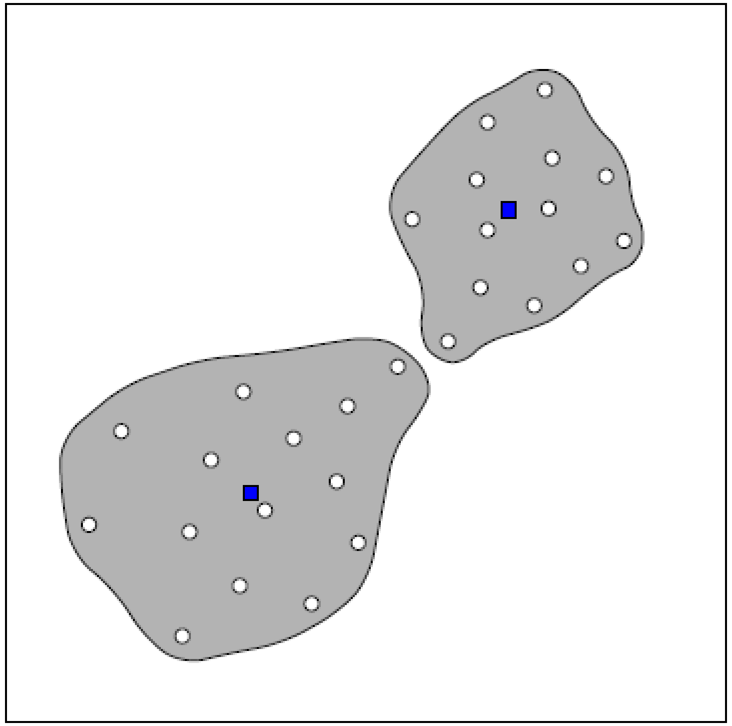
Iterative Cluster Analysis

k-Means with Minimization Criterion e = Variance



Iterative Cluster Analysis

k-Means with Minimization Criterion e = Variance



Iterative Cluster Analysis

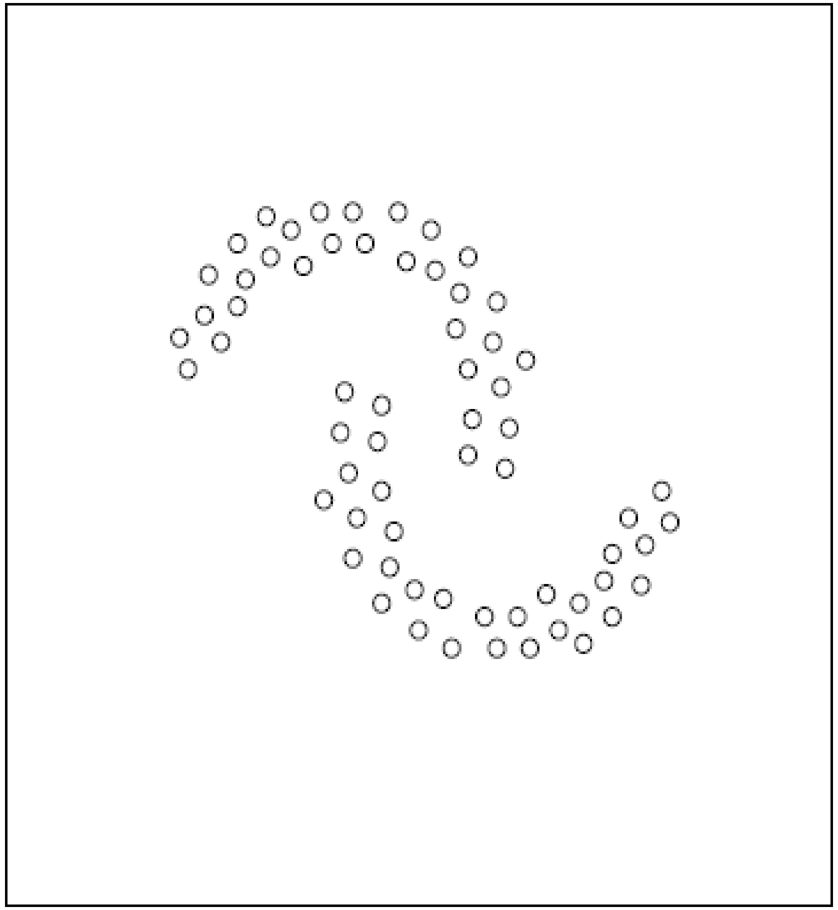
Minimization Criteria of Exemplar-Based Algorithms [algorithm]

$e(C_i) = \sum_{v \in C_i} (v - r_i)^2$	$r_i = \bar{v}_{C_i}$	centroid computation via variance minimization (k-means)
$e(C_i) = \sum_{v \in C_i} \lvert v - r_i \rvert$	$r_i \in C_i$	medoid computation (k-medoid)
$e(C_i) = \max_{v \in C_i} \lvert v - r_i \rvert$	$r_i \in C_i$	k-center
$e(C_i) = \sum_{v \in V} (\mu_i(v))^2 \cdot (v - r_i)^2$	$r_i = \frac{\sum_{v \in V} (\mu_i(v))^2 \cdot v}{\sum_{v \in V} (\mu_i(v))^2}$	Fuzzy k-means

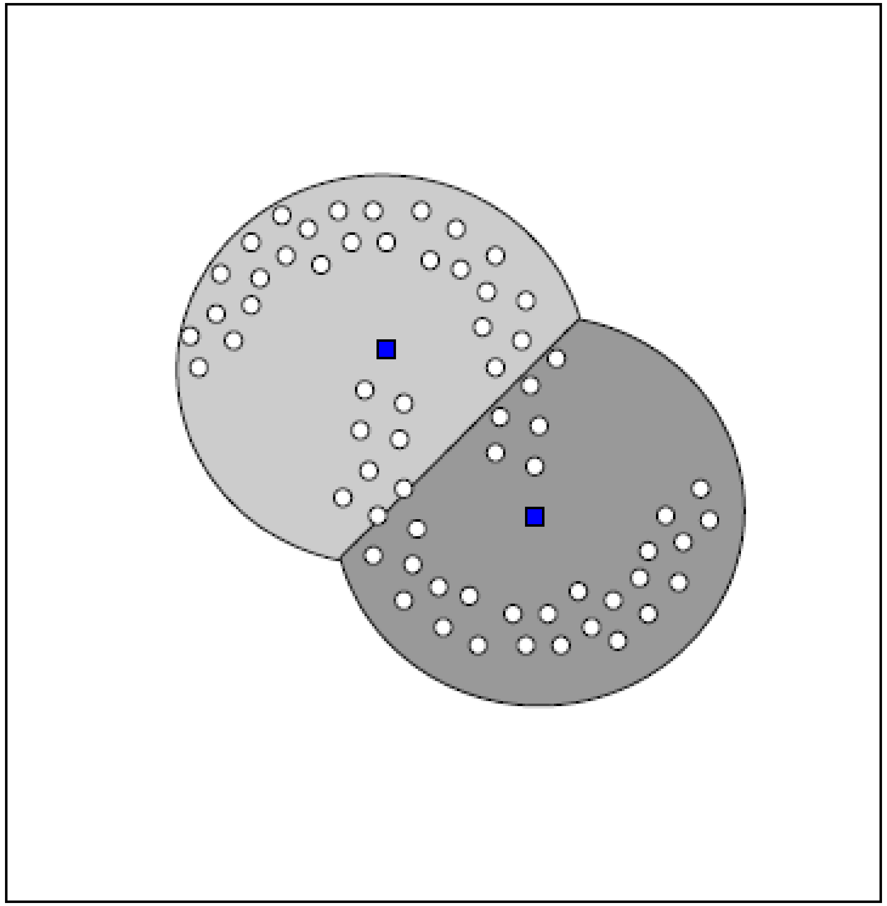
The sum of the squared distances to a cluster representative r_i becomes minimum, if r_i is the arithmetic mean of the points in C_i. Hence, the computation of the centroid in k-means corresponds to a local—i.e., cluster-specific—minimization of the variance.

The medoid or central element of a cluster denotes a point $r_i \in C_i$ that minimizes the sum of the distances from r_i to all other points in C_i. An advantage of medoids compared to centroids is their robustness with respect to outliers and, as a consequence, an improved convergence behavior (= smaller number of iterations).

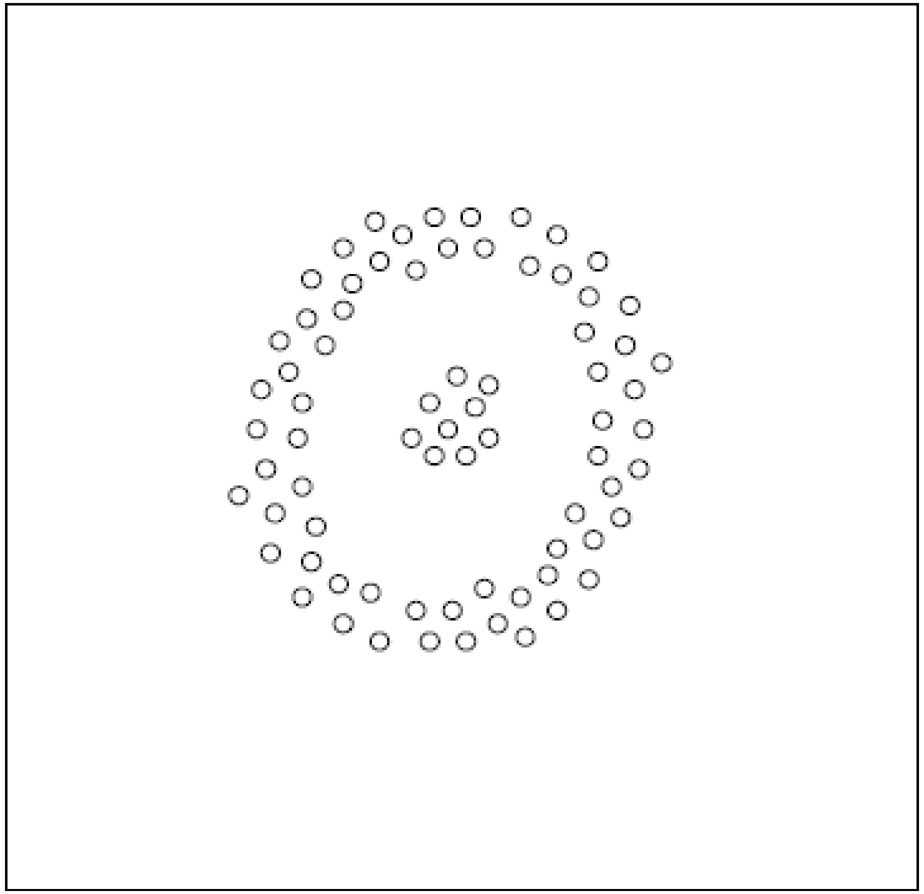
Issues



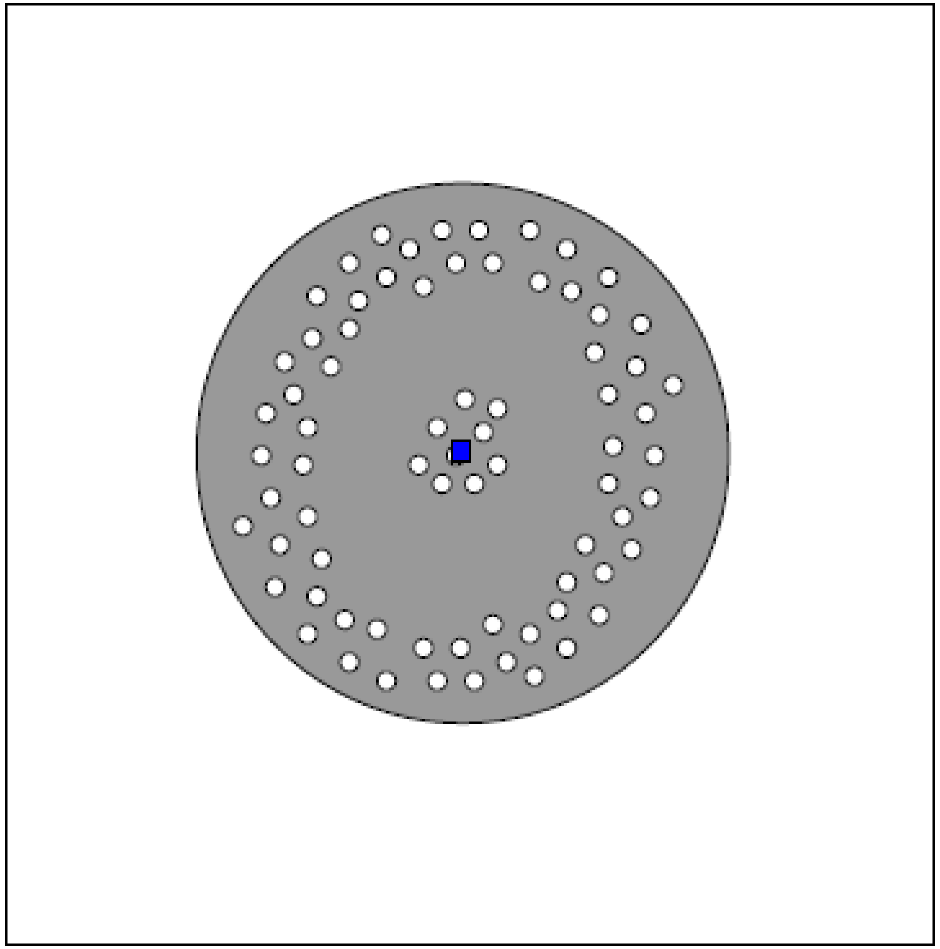
Exemplar-based algorithms fail to detect nested clusters.



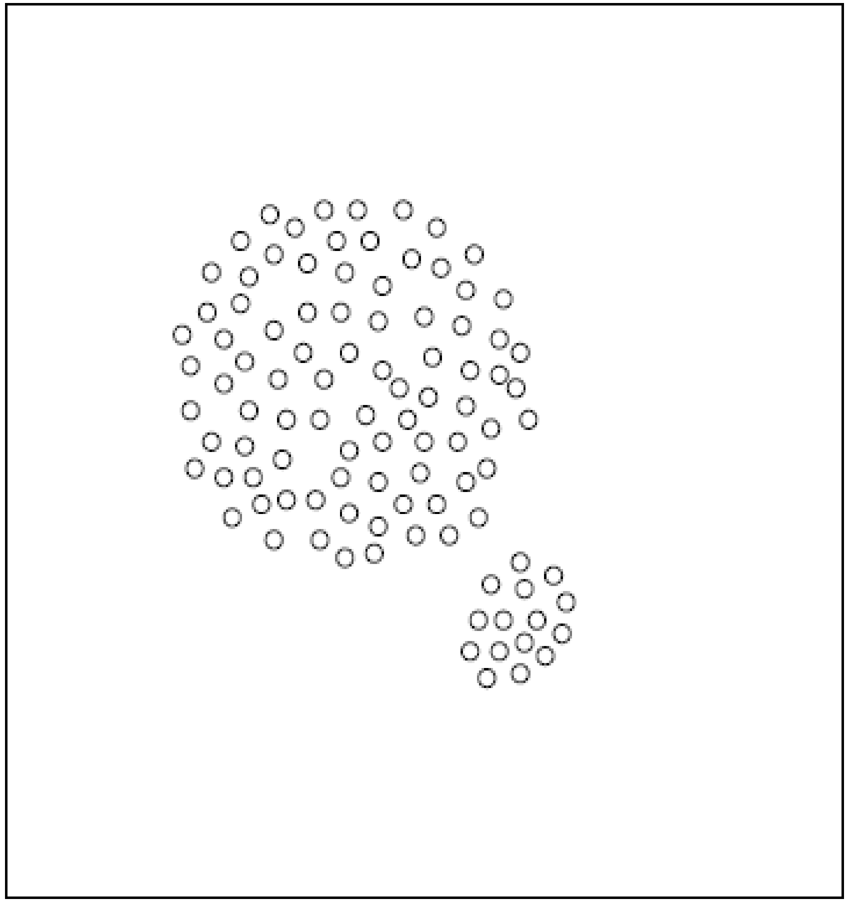
Exemplar-based algorithms fail to detect nested clusters.



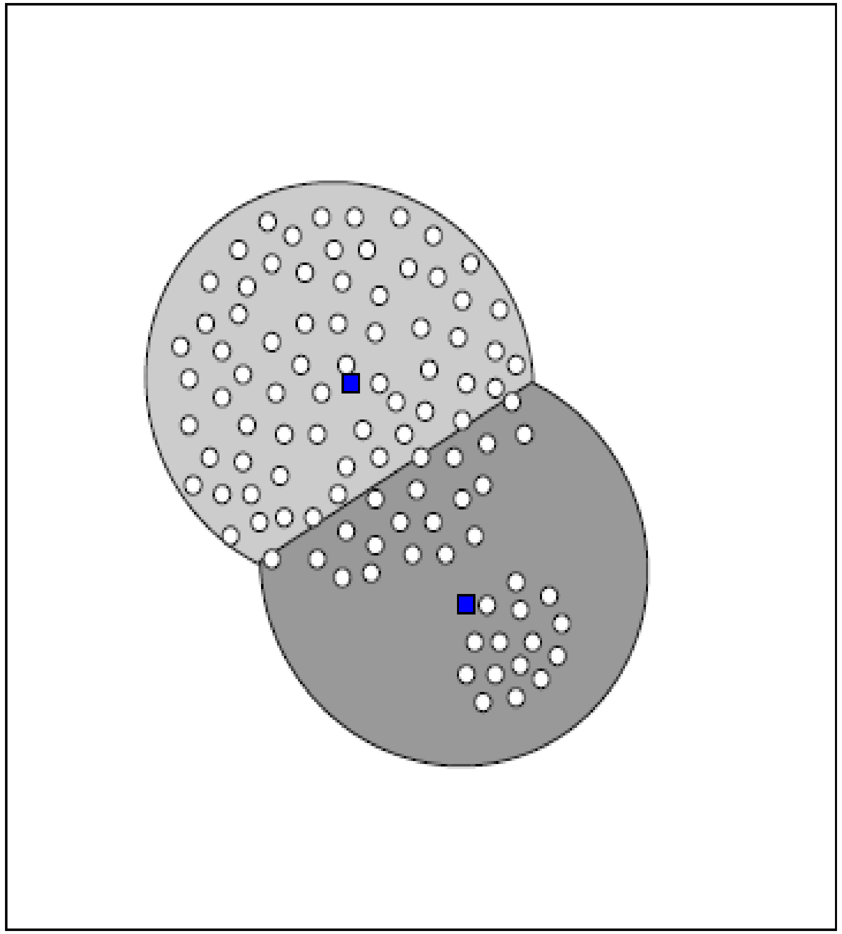
Exemplar-based algorithms fail to detect nested clusters.



Exemplar-based algorithms fail to detect nested clusters.



Exemplar-based algorithms fail to detect clusters with large difference in size.



Exemplar-based algorithms fail to detect clusters with large difference in size.