Data Mining: Clustering

cs4821-cs5831

Some slides adapted from P. Smyth; A. Moore, D. Klein Han, Kamber, Pei; Tan, Steinbach, Kumar; L. Kaebling; R. Tibshirani; T. Taylor; and L. Hannah

Outline

Unsupervised Learning

Clustering

- K-means Clustering
- Hierarchical Clustering
- *Other Types of Clustering*
  Density-based, *Grid-based*, *Model-based*, *Frequent pattern-based*, *Constraint-based*, *Link or Graph-based*

Clustering

What is Clustering?

Task of dividing up data into groups (clusters), so that points in any one group are more "similar" to each other than to points outside the group

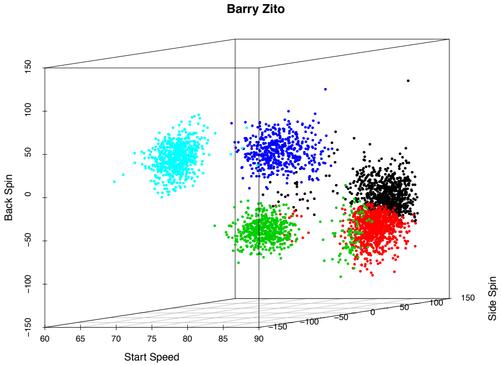
- Finds natural groupings among objects
- The number of groups (classes) is not known a priori, determined directly from the data

Clustering

Why Cluster?

- Summary - derive a reduced representation of the full data set
- Discovery - insights into the structure of the data, e.g., finding groups of songs that sound alike, chemicals that have similar properties, ...
- Other uses - help with prediction for classification, preprocessing step for other methods, check pre-existing group assignments

Example of Clustering



Inferred meaning of clusters: black - fastball, red - sinker, green - changeup, blue - slider, light blue - curveball

Example from R. Tibshirani

General Issues with Clustering

- No gold-standard, no ground truth
- Often no best clustering for a data set
- Different clustering algorithms may provide different groupings
- How many clusters to form?

Clustering is Ambiguous

How many clusters?

Original Data

Clustering is Ambiguous

How many clusters?

Original Data

2 clusters

Clustering is Ambiguous

How many clusters?

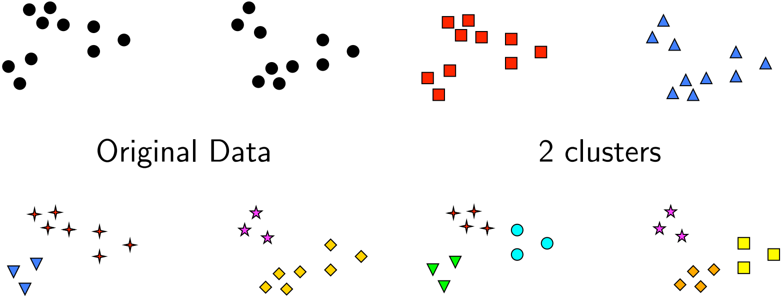
Original Data

2 clusters

4 clusters

Clustering is Ambiguous

How many clusters?



Original Data

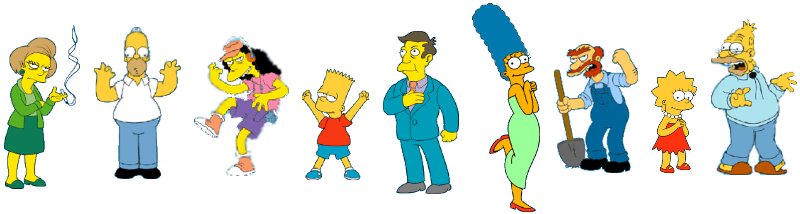
2 clusters

4 clusters

6 clusters

Clustering is Subjective

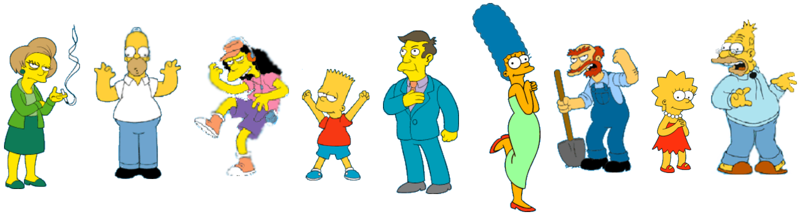
What is a natural grouping among these object?



slide from Eamonn Keogh

Clustering is Subjective

What is a natural grouping among these object?

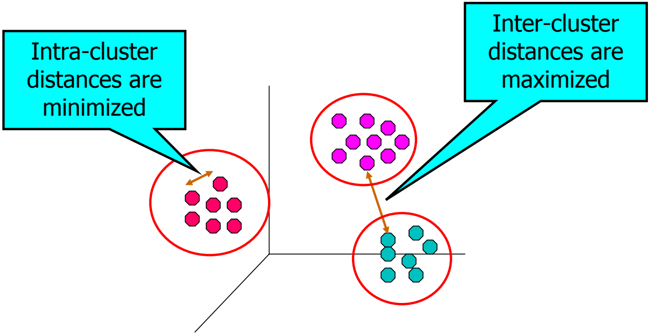


slide from Eamonn Keogh

What is Good Clustering?

- A good clustering method will produce high quality clusters
  - high intra-class similarity: cohesive within clusters
  - low inter-class similarity: distinctive between clusters
- The quality of a clustering method depends on:
  - the similarity/distance measure used (may be method dependent)
  - the method's implementation
  - its ability to discover some or all hidden patterns

What is Good Clustering?



Types of Clustering

- Partitional Clustering
  divide data into non-overlapping subsets (clusters) such that each data object is in exactly one subset
  Ex. k-means, k-medoids, CLARANS

- Hierarchical Clustering
  create a hierarchical decomposition of the set of data (hierarchical tree)
  Ex. Diana, Agnes, BIRCH, CHAMELION

- *Other Clustering Methods*
  density-based, grid-based, model-based, frequent pattern-based, constraint-based, link-based

Partitional Clustering

Problem

- Input:
  - Data set $\mathcal{D} = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n\}$ of n samples, where $\vec{x}_i \in \mathbb{R}^p$
  - A dissimilarity or distance measure $d(\vec{x}_i, \vec{x}_j)$, e.g., Euclidean distance
  - K the number of clusters
- Output:
  - K cluster centers, $c_1, \ldots, c_k$
  - a list of cluster assignments for each sample

Review of linear algebra operators, DMA 1.3

Clustering Definitions

Given a data set, $\mathcal{D} = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n\}$ and dissimilarity or distance measure $d_{ij} = d(\vec{x}_i, \vec{x}_j)$, e.g., let $d_{ij} = d(\vec{x}_i, \vec{x}_j) = \|\vec{x}_i - \vec{x}_j\|_2^2$

Let K be the number of clusterings. The clustering will return a function C that assigns each observation $\vec{x}_i$ to a group $k \in \{1, \ldots, K\}$.

Let $C(i) = k$ mean that $\vec{x}_i$ is assigned to group k. Let n_k be the number of samples in the group k

The within-cluster scatter is

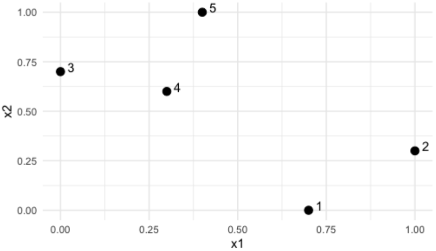
$$W = \frac{1}{2} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{C(i)=k, C(j)=k} d_{ij}$$

Example: Simple

R_cluster_simple
Let $n = 5$ and $K = 2$, where $x_i \in \mathbb{R}^2$ and $d_{ij} = \|x_i - x_j\|_2^2$

A dissimilarity matrix:

1	2	3	4	5
0.00	0.42	0.99	0.72	1.04
0.42	0.00	1.08	0.76	0.92
0.99	1.08	0.00	0.32	0.50
0.72	0.76	0.32	0.00	0.41
1.04	0.92	0.50	0.41	0.00



x1	x2
0.7	0.0
1.0	0.3
0.0	0.7
0.3	0.6
0.4	1.0

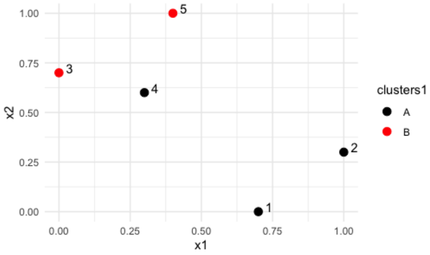
Example: Simple

R_cluster_simple

Let $n = 5$ and $K = 2$, where $\vec{x}_i \in \mathbb{R}^2$ and $d_{ij} = \|\vec{x}_i - \vec{x}_j\|_2^2$

A dissimilarity matrix:

1	2	3	4	5
0.00	0.42	0.99	0.72	1.04
0.42	0.00	1.08	0.76	0.92
0.99	1.08	0.00	0.32	0.50
0.72	0.76	0.32	0.00	0.41
1.04	0.92	0.50	0.41	0.00



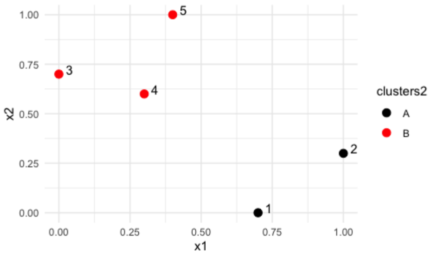
Clusters 1: $\{1, 2, 4\}, \{3, 5\}$

$$W_1 = (0.42 + 0.72 + 0.76)/3 + (0.5)/2 = 0.88$$

Example: Simple

R_cluster_simple

Let $n = 5$ and $K = 2$, where $\vec{x}_i \in \mathbb{R}^2$ and $d_{ij} = \|\vec{x}_i - \vec{x}_j\|_2^2$



A dissimilarity matrix:

1	2	3	4	5
0.00	0.42	0.99	0.72	1.04
0.42	0.00	1.08	0.76	0.92
0.99	1.08	0.00	0.32	0.50
0.72	0.76	0.32	0.00	0.41
1.04	0.92	0.50	0.41	0.00

Clusters 1: $\{1, 2, 4\}, \{3, 5\}$

$$W_1 = (0.42 + 0.72 + 0.76)/3 + (0.5)/2 = 0.88$$

Clusters 2: $\{1, 2\}, \{3, 4, 5\}$

$$W_2 = (0.42/2) + (0.32 + 0.5 + 0.41)/3 = 0.62$$

Finding Best Clusters

- From the previous example, we have seen **smaller W** is better.
- Idea: Find clusters by minimizing W
  - problem: minimizing W requires trying all possible assignments of samples to K groups. The number of possible assignments is given the Stirling numbers of the second kind:

$$S(n, K) = \frac{1}{K!} \sum_{k=1}^{K} (-1)^{K-k} \binom{K}{k} k^n$$

For $S(10, 4) = 34,105$, for $S(25, 4) \sim 5 \times 10^{13}$

- Have to find an approximation

Redefine Within-Cluster Scatter

Consider rewriting within-cluster scatter as

$$\frac{1}{2}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{C(i)=k}\sum_{C(j)=k}\|x_i - x_j\|_2^2 = \sum_{k=1}^{K}\sum_{C(i)=k}\|x_i - \bar{x}_k\|_2^2$$

where $\bar{x}_k$ is the average of the points in group k,

$$\bar{x}_k = \frac{1}{n_k}\sum_{C(i)=k} x_i$$

This is also known as the within-cluster variation

notation adapted from ISLR Ch. 10

Redefining the Problem

We want to choose a clustering $\hat{C}$ to minimize

$$\sum_{k=1}^{K} \sum_{C(i)=k} \|x_i - \bar{x}_k\|_2^2$$

In other words, solve the following optimization problem:

$$\min_{C, \{c_k\}_1^K} \sum_{k=1}^{K} \sum_{C(i)=k} \|x_i - c_k\|_2^2$$

over the clusterings C and cluster centers $c_1, \ldots, c_K$

K-means Algorithm

The k-means clustering algorithm works to minimize the criterion by alternately minimizing over C and $c_1, \dots, c_K$

Method:

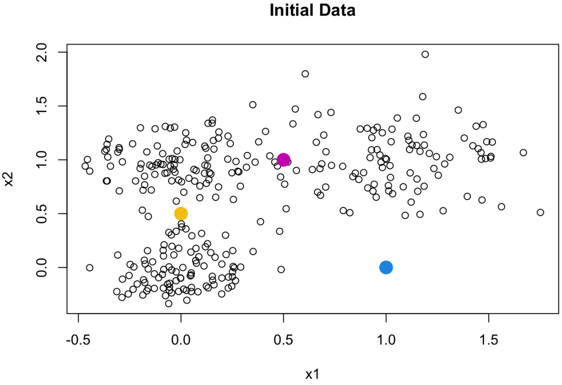
1. Start with an initial guess for $c_1, \dots, c_K$, then repeat:
2. Repeat until within-cluster variation doesn't change or cluster assignments stop changing:

   A. *Cluster Assignment Step*, Minimize over C:
   for each $i = 1, \dots, n$, find the cluster center c_k closest to x_i
   assign $C(i) = k$

   B. *Centroid Update Step*, Minimize over $c_1, \dots, c_k$:
   for each $k = 1, \dots, K$,
   assign $c_k = \bar{x}_k$, the average points in group k

{MacQueen '67, Lloyd '57/'82}

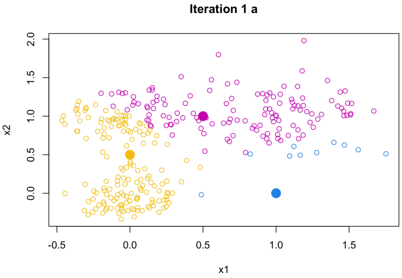
Example: K-means



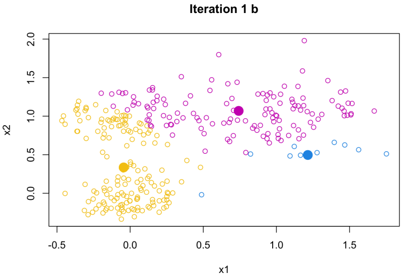
Data and Initial Centers

Example given in: {R, Python}_cluster_kmeans

Example: K-means



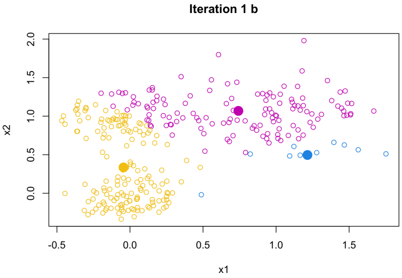
Cluster Assignment Step



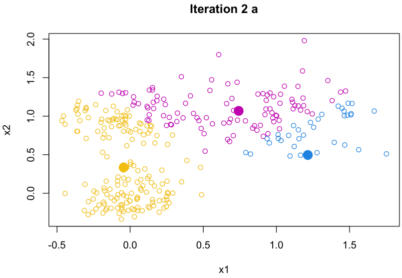
Centroid Update Step

Example given in: {R, Python}_cluster_kmeans

Example: K-means



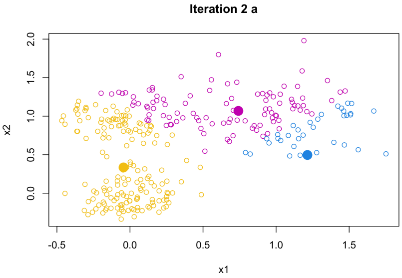
After Centroid Update



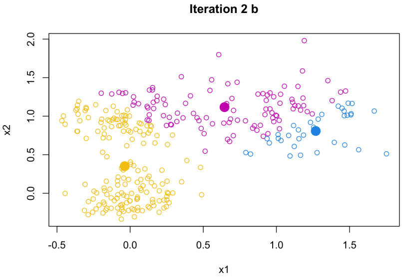
Cluster Assignment

Example given in: {R, Python}_cluster_kmeans

Example: K-means



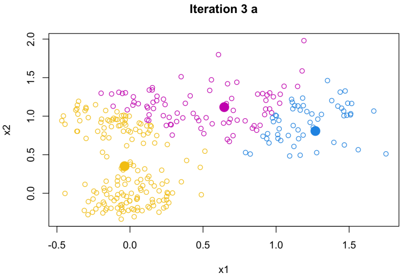
After Cluster Assignment



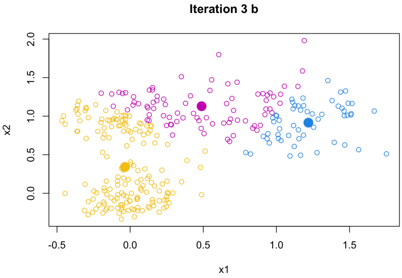
Centroid Update

Example given in: {R, Python}_cluster_kmeans

Example: K-means



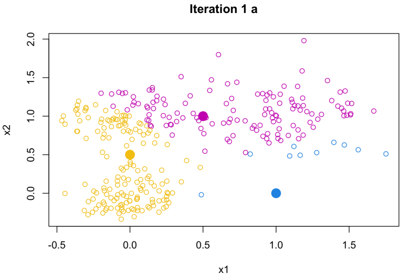
Cluster Assignment



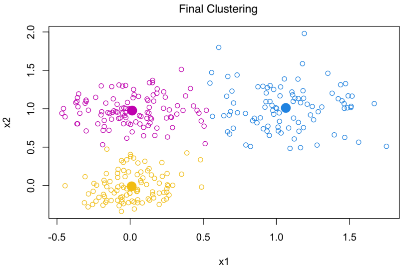
Centroid Update

Example given in: {R, Python}_cluster_kmeans

Example: K-means



Initial Centers



Final Assigned Clusters

Example given in: {R, Python}_cluster_kmeans

K-means Properties

- Efficiency: $O(tkn)$, where n is number of samples, k is the number of clusters, and t is the number of iterations
- The within-cluster variation decreases with each iteration
- The algorithm always converges to "some" solution, but not necessarily the best solution
- The final clustering depends on the initial cluster centers
- The value of K needs to be specified in advance
- The method can be sensitive to noisy data and outliers
- The method is not suitable to discover clusters with non-convex shapes

Voronoi tessellation

Given cluster centers, we identify each point to its nearest center. This defines a Voronoi tessellation in $\mathbb{R}^p$

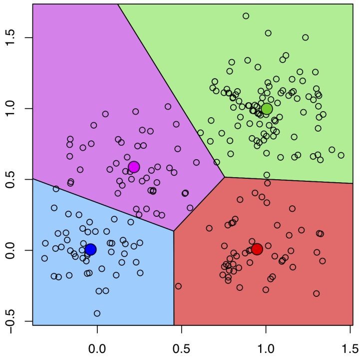


Image from R. Tibshirani

K-means - Choosing the initial points

The results of K-means with different initial centers (chosen randomly over the range of the x_i's)

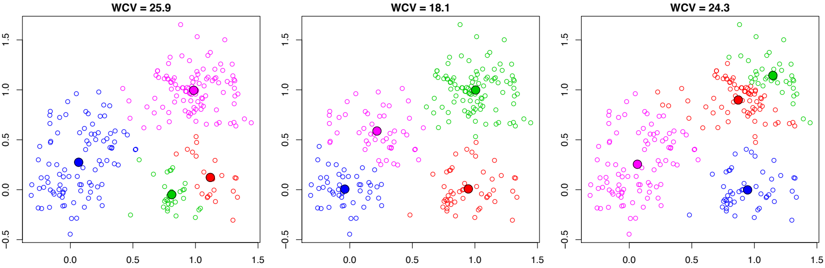


Image from R. Tibshirani

K-means - Choosing the initial points

- Multiple runs
    - Repeat problem multiple times to determine stable clusters over multiple runs
- Use hierarchical clustering to determine initial centroids
- Select more than K initial centroid and then select among these initial centroids
    - select most widely separated

What is the right number of clusters?

This is a hard problem!

- Why is it hard?
  Determining the number of clusters is a hard task for humans (unless data is low-dimensional). It is hard to explain what it is that we're looking for.
- Why is it important?
  - May have major ramifications in data domain (3 sub-types of a diseases vs. 4 sub-types of a disease)
- Methods
  - "elbow" or "knee" method
  - statistical measures

Choosing K - Approach 1

Focusing on K-means, the K-means algorithm approximately minimizes the within-cluster variation:

$$W = \sum_{k=1}^{K} \sum_{C(i)=k} \|x_i - \bar{x}_k\|_2^2$$

over clustering assignments C, where $\bar{x}_k$ is the average of points in group k.

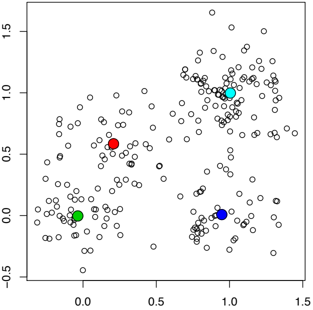
A lower value of W is better. So just run K-means for a number of different values of K and choose the value of K with the smallest W.

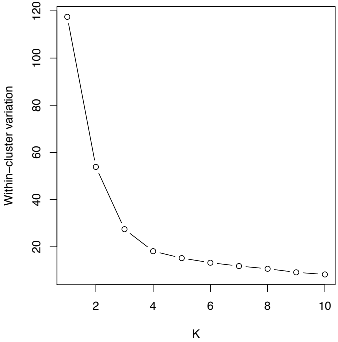
What is the problem?

Choosing K - Approach 1

Problem: within-cluster variation always decreases with large values of K

Example: n=250, p=2, $K = 1, \ldots, 10$





Between cluster variation

Within-cluster variation measures how tightly grouped the clusters are. As K increases, this values keeps going down. What else is needed?

Between-cluster variation measures how spread apart the groups are from each other:

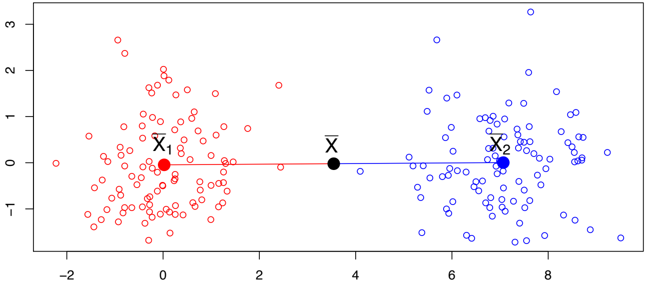
$$B = \sum_{k=1}^{K} n_k \|\bar{x}_k - \bar{x}\|_2^2$$

where $\bar{x}_k$ is the average point in group k, and $\bar{x}$ is the overall average

$$\bar{x}_k = \frac{1}{n_k} \sum_{C(i)=k} x_i \quad \text{and} \quad \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Example: Between cluster variation

Example: $n = 100$, $p = 2$, $K = 2$



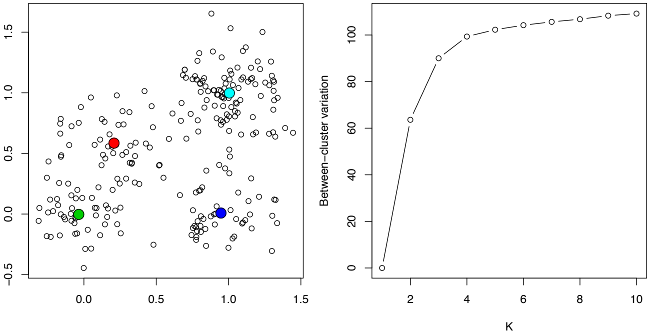
$$B = n_1 \|\bar{x}_1 - \bar{x}\|_2^2 + n_2 \|\bar{x}_2 - \bar{x}\|_2^2$$

$$W = \sum_{C(i)=1} \|x_i - \bar{x}_1\|_2^2 + \sum_{C(i)=2} \|x_i - \bar{x}_2\|_2^2$$

Choosing K - Approach 2

Larger values of B are better. So, can we just use B to choose the number of clusters?

No, between cluster variation keeps increasing



Choosing K - Approach 3 - CH index

Ideally, clustering assignments should have simultaneously a small W and a large B

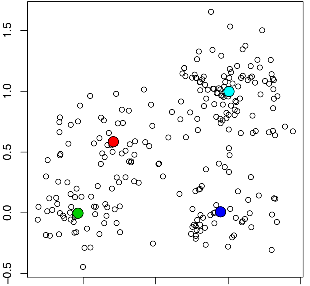
This is idea of CH index (Calinski and Harabasz, 1974). For clustering assignments coming from K clusters, we have the CH score:

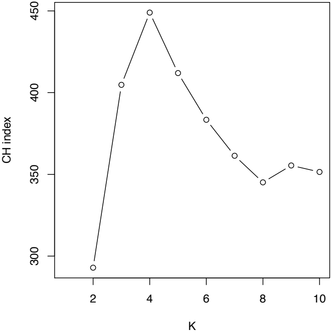
$$CH(K) = \frac{B(K)/(K-1)}{W(K)/(n-K)}$$

To choose K, pick a maximum number of clusters to consider K_{max}, and choose the value of K with the largest $CH(K)$, i.e.,

$$\hat{K} = \underset{K \in \{2,\ldots,K_{max}\}}{argmax} \; CH(K)$$

Example: CH index





Choose $K = 4$ clusters.

Choosing K - Approach 4 - Gap statistic

$W(K)$ always decreases, but how much it drops for any given K is informative.

The gap statistic is based on this idea (Tishirani et al., 2001). Compare the observed within-cluster variation $W(K)$ to $W_{unif}(K)$, the within-cluster variation if the data points were uniformly distributed. The gap is defined as

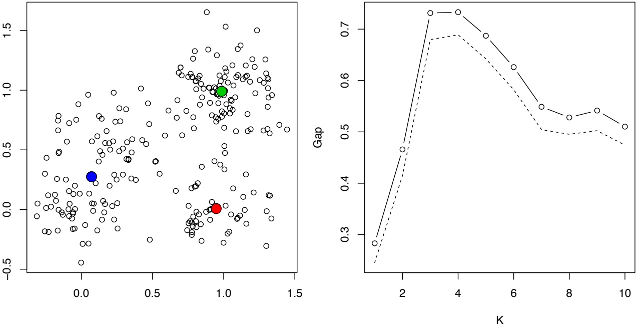
$$Gap(K) = \log W(k) - \log W_{unif}(K)$$

The value $\log W_{unif}(K)$ is computed by simulation; average log within-cluster variations over some number of simulated uniform data sets. Can also compute the standard error $s(K)$ of $\log W_{unif}(K)$.

Choose K as

$$\hat{K} = \underset{K \in \{1,\ldots,K_{max}\}}{argmax} \; Gap(K) \geq Gap(K+1) - s(K+1)$$

Example: Gap statistic



Choose $K = 3$ or $K = 4$ clusters.

K-means - Enhancements

- Handle empty clusters
  Basic k-means can result in empty clusters
- Several Strategies
  - choose the point that contributes most to the SSE
  - choose a point from the cluster with the highest SSE
  - if there are several empty clusters, the above can be repeated several times

K-means - Enhancements

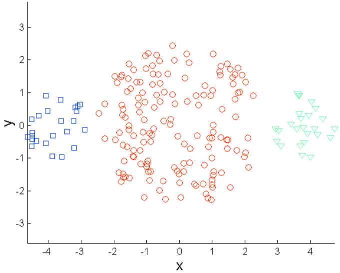
- Incremental Updating
  In basic k-means, centroids are updated after all points are assigned to a centroid
- An alternative is to update the centroid after each assignment (incremental updating)
  - each assignment updates zero or two centroids
  - more expensive
  - introduces order dependency
  - never get an empty cluster

K-means - Limitations

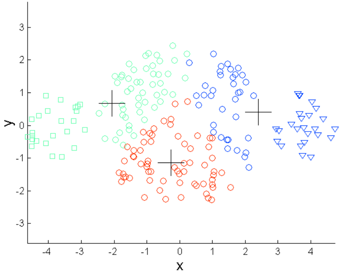
K-means has problems when clusters are of differing:

- sizes
- densities
- non-convex shapes
- has outliers

Limitation of K-means: sizes

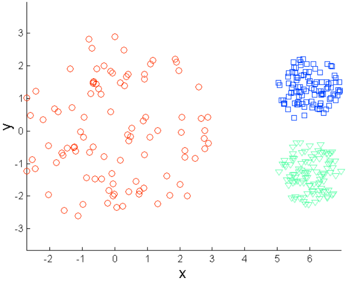


Original Points

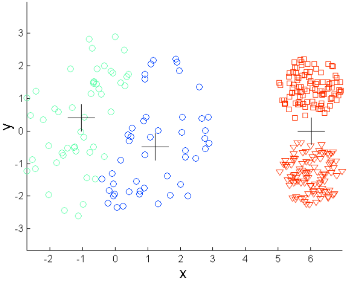


K-means (3 Clusters)

Limitation of K-means: densities

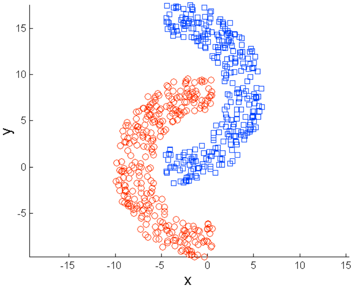


Original Points

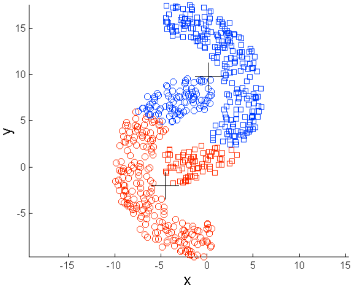


K-means (3 Clusters)

Limitation of K-means: non-convex shapes



Original Points



K-means (2 Clusters)

K-means and K-medoids

- k-means is sensitive to outliers
  - an object with an extremely large value may substantially distort the distribution of the data
- k-medoids instead of taking the mean values of the object in a cluster, *medoids* can be used, which is the most centrally located object in a cluster

K-medoids Clustering

- K-medoids algorithm is similar to k-means, except that the centroid is estimated not by the average, but by the observation having the minimum pairwise distance with the other cluster members.
- The advantage of this method is the centroid is an actual observation. The method also then allows to only keep track of the pairwise distances rather than the raw observations
- Method:
  - In `R`, `pam` implements k-medoids using Euclidean distance
  - In `Matlab`, `kmedoids` is available
  - In `Python`, `KMedoids` is in the `sklearn_extra` package

PAM - Kaufman & Rousseeuw '87, CLARA- Kaufman & Rousseeuw, '90, CLARANS - Ng & Han, '94

K-medoids Algorithm

The k-medoids clustering algorithm works similarly to k-means except the centers $c_1, \ldots, c_k$, come from the observations.

Method:

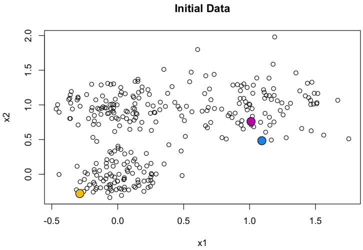
1. Start with an initial guess for $c_1, \ldots, c_k$ (select from n samples), then:

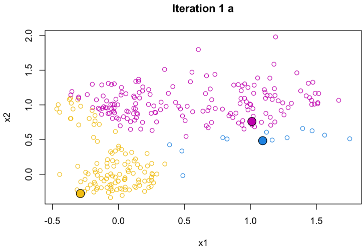
2. Repeat until within-cluster variation doesn't change or cluster assignments stop changing:

   A. *Cluster Update Step*, Minimize over C: for each $i = 1, \ldots, n$, find the cluster center c_k closest to x_i, and let $C(i) = k$

   B. *Medoid Update Step*, Minimize over $c_1, \ldots, c_k$: for each $k = 1, \ldots, K$, let $c_k = x_k^*$, the medoid of the points in cluster k, i.e., the point x_i in cluster k that minimizes $\sum_{C(j)=k} \|x_j - x_i\|_2^2$

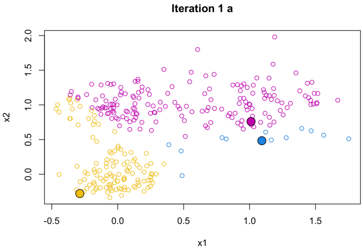
K-medoids Example

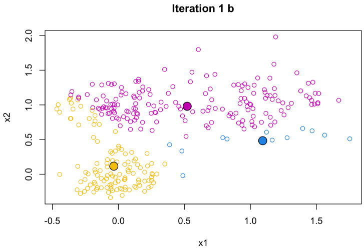




Example given in: {R, Python}_cluster_kmedoids

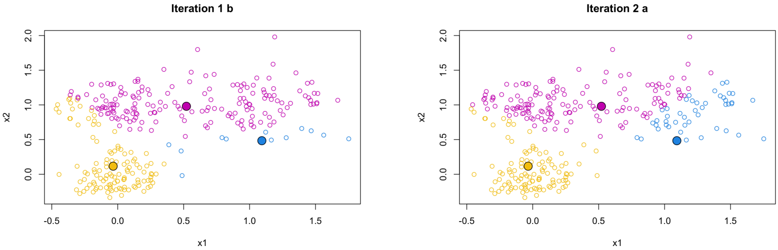
K-medoids Example





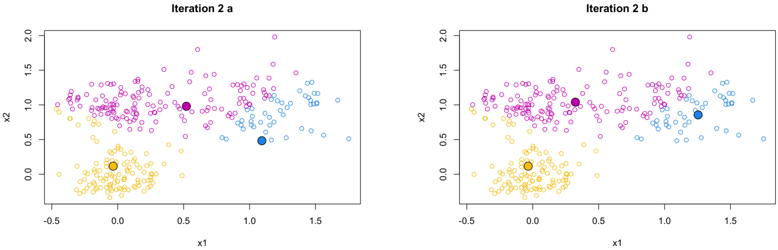
Example given in: {R, Python}_cluster_kmedoids

K-medoids Example



Example given in: {R, Python}_cluster_kmedoids

K-medoids Example

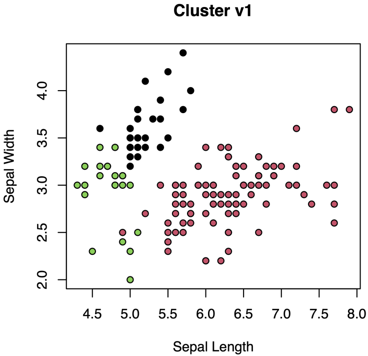


Example given in: {R, Python}_cluster_kmedoids

K-medoids Example 2 - Iris data

Instability / Stability of Kmeans vs. K-medoids Algorithm running from different initial centers results in similar clusterings

Kmeans

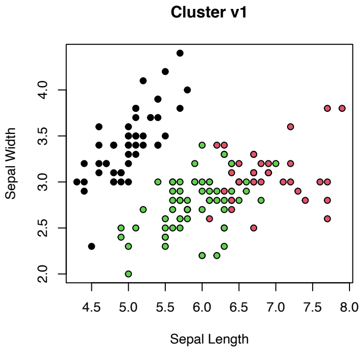


Example given in: {R, Python}_cluster_initial_centers

K-medoids Example 2 - Iris data

Instability / Stability of Kmeans vs. K-medoids Algorithm running from different initial centers results in similar clusterings

Kmedoids



Example given in: {R, Python}_cluster_initial_centers

Properties of K-medoids

The k-medoids algorithm shares many of the same properties as the k-means algorithm

- the method always converges
- different starts produce different final answers
- does not achieve the global minimum

Additionally, k-medoids is computationally more expensive than k-means (it is harder to compute the medoid than the average)