Machine Learning

CSE 142

Xin (Eric) Wang

Friday, November 12, 2021

Today

- Clustering (k-means, k-medoids)

Guest Lecture next Monday

- **Predicting behavior of road users**, by Vihan Jain
- Bio: Vihan Jain is a Staff Software Engineer at Waymo in Mountain View where he is tech-leading a team which develops and deploys Machine Learned models that predict the behavior of road users as the autonomous driving vehicle navigates through a scene. Previously, he was at Google Research where he worked on multi-modal learning, natural language compositionality and video semantic understanding. He has also worked on long-term value modeling in recommendation systems, designing configurable simulation platforms for studying recommendations, wide and deep learning and TensorFlow infrastructure. Prior to moving to the US, he worked with Ads Infrastructure at Google Canada and did an internship at Google India during the summer of 2012. He graduated from Indian Institute of Technology Roorkee as a gold medalist in 2013. In his free time, he likes to travel and play/follow several sports.

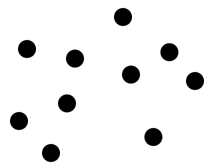
Clustering vs. classification

- Classification vs. clustering
  - In a classifier, possible class labels are provided
    - { dog, cat, elephant, mouse, …}, { spam, ham }, etc.
    - Given in the training data (for supervised classification)
  - In a clustering problem, possible labels are the cluster labels learned from the training set
    - { cluster 1, cluster 2, cluster 3, …}
    - Not given in the training data
- Terminology: In both cases, people often refer to the assigning of labels or clusters to data points (during the learning/training process, or afterwards in testing) as classification
  - Even if it's a clustering problem!

Clustering

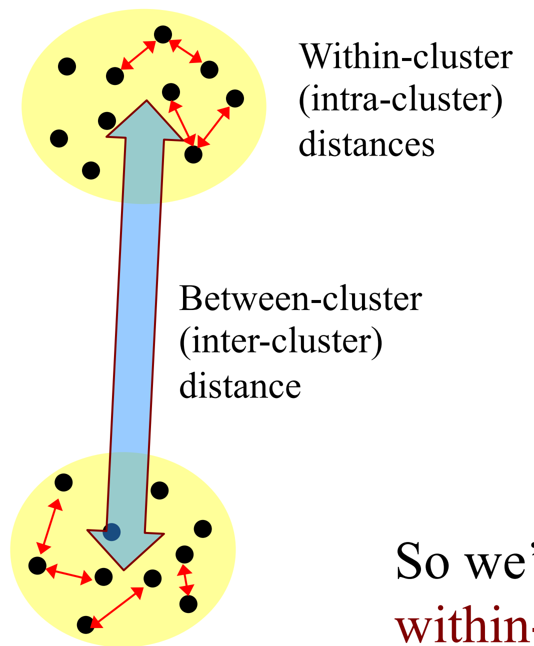
- The goal of clustering is to find clusters (groupings) that are compact with respect to the distance metric
- What do we mean by *compactness*?

Is this cluster compact?

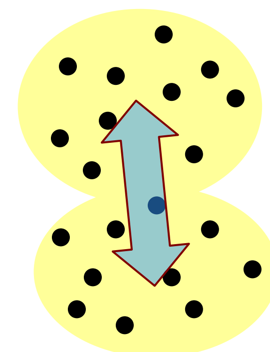


It depends….

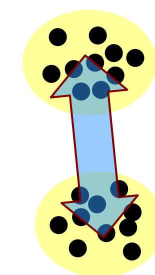
Yes



No



Yes



So we'd like to have a measure of within-cluster and between-cluster distribution or ***scatter***

Scatter matrix

This is a widely-used concept in ML!

Covariance matrix:

Sample covariance: $\hat{\Sigma}_{ij} = \frac{1}{k}\sum_k (x_{ik} - \hat{\mu}_i)(x_{jk} - \hat{\mu}_j) = \frac{1}{k} S_{ij}$

If $\boldsymbol{X_z}$ is a matrix that holds all the zero-centered samples as <u>column</u> vectors, then

$$\hat{\Sigma} = \frac{1}{k}\boxed{\boldsymbol{X_z}\boldsymbol{X_z}^T} = \frac{1}{k}\boldsymbol{S}$$

S is the Scatter matrix

Alternatively, if $\boldsymbol{X_z}$ is a matrix that holds all the zero-centered samples as <u>row</u> vectors, then

$$\boldsymbol{S} = \boxed{\boldsymbol{X_z}^T\boldsymbol{X_z}}$$

It depends on how we define $\boldsymbol{X_z}$!

For the scatter matrix (and thus the covariance matrix), $\boldsymbol{X_z}$ is zero-mean

– That is, the mean data point $\bar{\boldsymbol{x}}$ (or $\boldsymbol{\mu}$ or $\boldsymbol{\mu_x}$) is first subtracted from every data point $\boldsymbol{x}_i$

By the way, the Gram matrix is not zero-mean…

Scatter matrix

- If the data D is partitioned into K subsets $\{D_1, D_2, \ldots D_K\}$ then the scatter matrix can be written as

$$\boldsymbol{S} = \left(\sum_{j=1}^{K} \boldsymbol{S}_j\right) + \boldsymbol{B}$$

Scatter matrix of the partition means

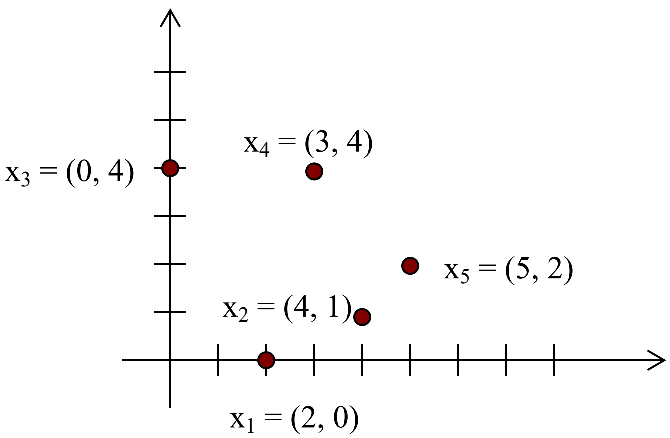
Between-cluster scatter matrix

Subset scatter matrices

Within-cluster scatter matrices

To compute B, replace every point in D with the mean of its partition D_i and compute the scatter matrix

Example: Scatter matrix



$$\boldsymbol{X} = [\boldsymbol{x}_1 \quad \boldsymbol{x}_2 \quad \boldsymbol{x}_3 \quad \boldsymbol{x}_4 \quad \boldsymbol{x}_5] = \begin{bmatrix} 2 & 4 & 0 & 3 & 5 \\ 0 & 1 & 4 & 4 & 2 \end{bmatrix}$$

The Gram matrix is …

$$\boldsymbol{G} = \boldsymbol{X}^T\boldsymbol{X} = \begin{bmatrix} 4 & 8 & 0 & 6 & 10 \\ 8 & 17 & 4 & 16 & 22 \\ 0 & 4 & 16 & 16 & 8 \\ 6 & 16 & 16 & 25 & 23 \\ 10 & 22 & 8 & 23 & 29 \end{bmatrix}$$

($k \times k$), where k is the number of data points

$$\bar{\boldsymbol{x}} = \frac{1}{5}\sum_{i=1}^{5} \boldsymbol{x}_i = \frac{1}{5}\begin{bmatrix} 14 \\ 11 \end{bmatrix} = \begin{bmatrix} 2.8 \\ 2.2 \end{bmatrix}$$

$$\boldsymbol{X_z} = [\boldsymbol{x}_1 - \bar{\boldsymbol{x}} \quad \boldsymbol{x}_2 - \bar{\boldsymbol{x}} \quad \boldsymbol{x}_3 - \bar{\boldsymbol{x}} \quad \boldsymbol{x}_4 - \bar{\boldsymbol{x}} \quad \boldsymbol{x}_5 - \bar{\boldsymbol{x}}] = \begin{bmatrix} -0.8 & 1.2 & -2.8 & 0.2 & 2.2 \\ -2.2 & -1.2 & 1.8 & 1.8 & -0.2 \end{bmatrix}$$

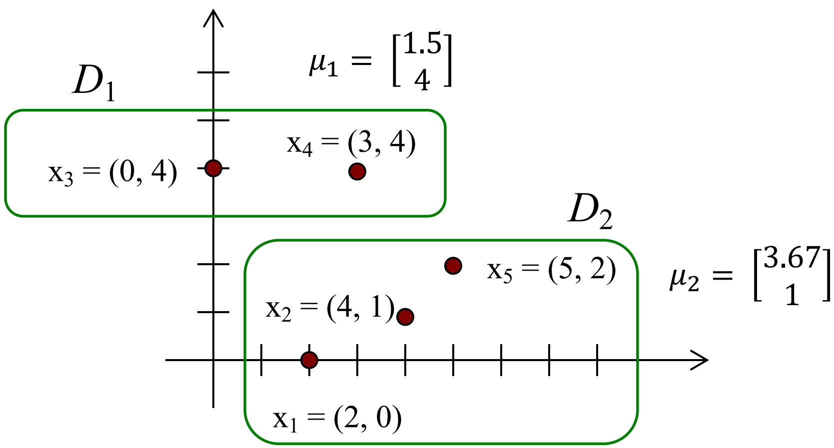
The Scatter matrix is …

$$\boldsymbol{S} = \boldsymbol{X}_z\boldsymbol{X}_z{}^T = \begin{bmatrix} 14.8 & -4.8 \\ -4.8 & 12.8 \end{bmatrix}$$

(N x N), where N is the dimensionality of the data points

Example: Scatter matrix via partitions

$$\boldsymbol{S} = \left(\sum_{j=1}^{K} \boldsymbol{S}_j\right) + \boldsymbol{B}$$



Partition means

$$\begin{bmatrix}1.5 & 1.5 & 3.67 & 3.67 & 3.67\\ 4 & 4 & 1 & 1 & 1\end{bmatrix} \qquad \mu_B = \begin{bmatrix}2.8\\2.2\end{bmatrix}$$

Zero-mean partition means

$$B_z = \begin{bmatrix}-1.3 & -1.3 & 13/15 & 13/15 & 13/15\\ 1.8 & 1.8 & -1.2 & -1.2 & -1.2\end{bmatrix}$$

Between-cluster scatter matrix

$$\boldsymbol{B} = \boldsymbol{B}_z \boldsymbol{B}_z{}^T = \begin{bmatrix}5.633 & -7.8\\ -7.8 & 10.8\end{bmatrix}$$

Scatter matrix of D_1

$$\boldsymbol{S}_1 = \left[\boldsymbol{x}_3 - \begin{bmatrix}1.5\\4\end{bmatrix} \quad \boldsymbol{x}_4 - \begin{bmatrix}1.5\\4\end{bmatrix}\right]\left[\boldsymbol{x}_3 - \begin{bmatrix}1.5\\4\end{bmatrix} \quad \boldsymbol{x}_4 - \begin{bmatrix}1.5\\4\end{bmatrix}\right]^T = \begin{bmatrix}-1.5 & 1.5\\ 0 & 0\end{bmatrix}\begin{bmatrix}-1.5 & 1.5\\ 0 & 0\end{bmatrix}^T = \begin{bmatrix}4.5 & 0\\ 0 & 0\end{bmatrix}$$

Scatter matrix of D_2

$$\boldsymbol{S}_2 = \begin{bmatrix}-5/3 & 1/3 & 4/3\\ -1 & 0 & 1\end{bmatrix}\begin{bmatrix}-5/3 & 1/3 & 4/3\\ -1 & 0 & 1\end{bmatrix}^T = \begin{bmatrix}4.67 & 3\\ 3 & 2\end{bmatrix}$$

$$\boldsymbol{S} = \boldsymbol{S}_1 + \boldsymbol{S}_1 + \boldsymbol{B} = \begin{bmatrix}4.5 & 0\\ 0 & 0\end{bmatrix} + \begin{bmatrix}4.67 & 3\\ 3 & 2\end{bmatrix} + \begin{bmatrix}5.633 & -7.8\\ -7.8 & 10.8\end{bmatrix} = \begin{bmatrix}14.8 & -4.8\\ -4.8 & 12.8\end{bmatrix}$$

Clustering

- The goal of clustering is to find clusters (groupings) that are compact with respect to the distance metric
- Good clustering is characterized by low within-cluster variance and high between-cluster variance
- The scatter matrix, $\boldsymbol{S} = \boldsymbol{X}_z \boldsymbol{X}_z^T$, gives us a measure of variance
- If the data D is partitioned into K subsets/partitions $\{D_1, D_2, \dots D_K\}$, then the scatter matrix can be written as

$$\boldsymbol{S} = \left(\sum_{j=1}^{K} \boldsymbol{S}_j\right) + \boldsymbol{B}$$

Between-cluster scatter matrix
(maximize!)

Within-cluster scatter matrices
(minimize!)

Scatter

- The scatter of D is defined as the trace of the scatter matrix
  - The *trace* is the sum of the diagonal elements of a square matrix

$$\mathrm{Scat}(D) = Tr(\boldsymbol{S}) = Tr(\boldsymbol{X}_z \boldsymbol{X}_z{}^T)$$

$$= Tr\left(\begin{bmatrix} 14.8 & -4.8 \\ -4.8 & 12.8 \end{bmatrix}\right) = 14.8 + 12.8 = 27.6$$

- Since $\boldsymbol{S}$ can be decomposed into partitions, so can Scat(D)

$$\mathrm{Scat}(D) = \underbrace{\sum_{j=1}^{K} \mathrm{Scat}(D_j)}_{} + \underbrace{\sum_{j=1}^{K} |D_j| \, ||\boldsymbol{\mu}_j - \boldsymbol{\mu}||^2}_{}$$

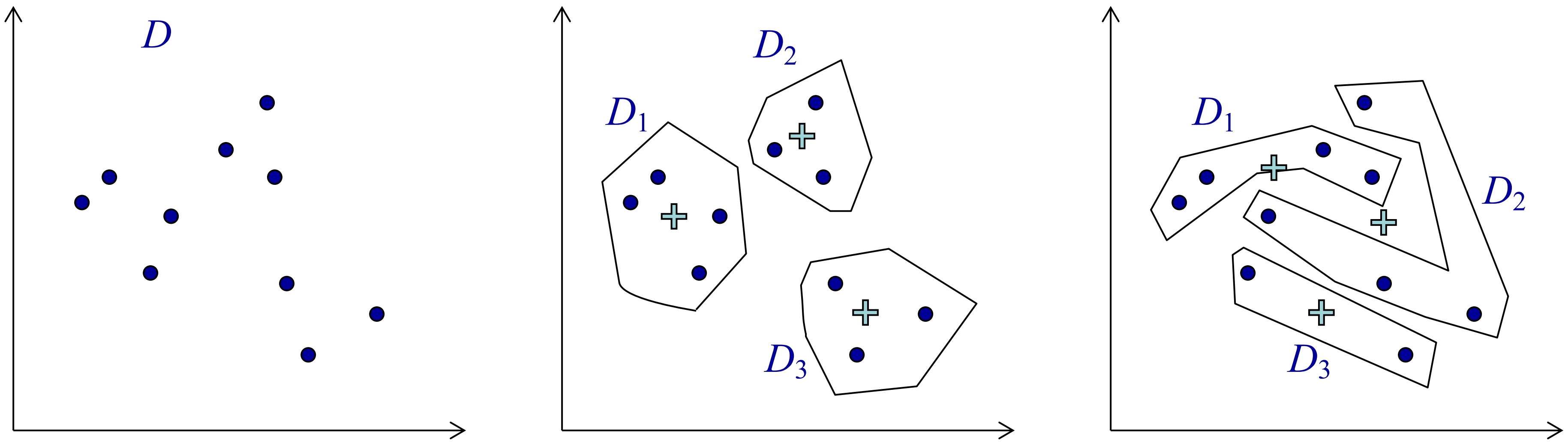
Fixed for a given data set

Want to choose partitions that minimize this

Equivalent to maximizing this

This is the goal of *k*-means clustering

Scatter



$$\text{Scat}(D) = \sum_{j=1}^{K} \text{Scat}(D_j) + \sum_{j=1}^{K} |D_j| \, ||\boldsymbol{\mu}_j - \boldsymbol{\mu}||^2$$

K-means clustering

- The general K-means clustering problem is NP-complete, so there is no efficient solution to find the optimal clustering (data partition)
- A widely-used heuristic algorithm for clustering is also known as the K-means algorithm, but it is not optimal
  - It will converge to a solution, but there is no guarantee that the solution is the best one (the global minimum of scatter)
  - But it works quite well in most cases!
- Typically, the K-means algorithm would be run several times (with a random starting point) and then the best solution is selected
  - I.e., the solution with the smallest within-cluster scatter

K-means algorithm

K is an input parameter

Algorithm KMeans(D, K) – K-means clustering using Euclidean distance Dis_2.

Input : data $D \subseteq \mathbb{R}^d$; number of clusters $K \in \mathbb{N}$.

Output : K cluster means $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K \in \mathbb{R}^d$.

randomly initialise K vectors $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K \in \mathbb{R}^d$;

repeat

  assign each $\mathbf{x} \in D$ to $\arg\min_j \mathrm{Dis}_2(\mathbf{x}, \boldsymbol{\mu}_j)$; ⟵ *1-Nearest neighbor assignment*

  for $j = 1$ to K **do**

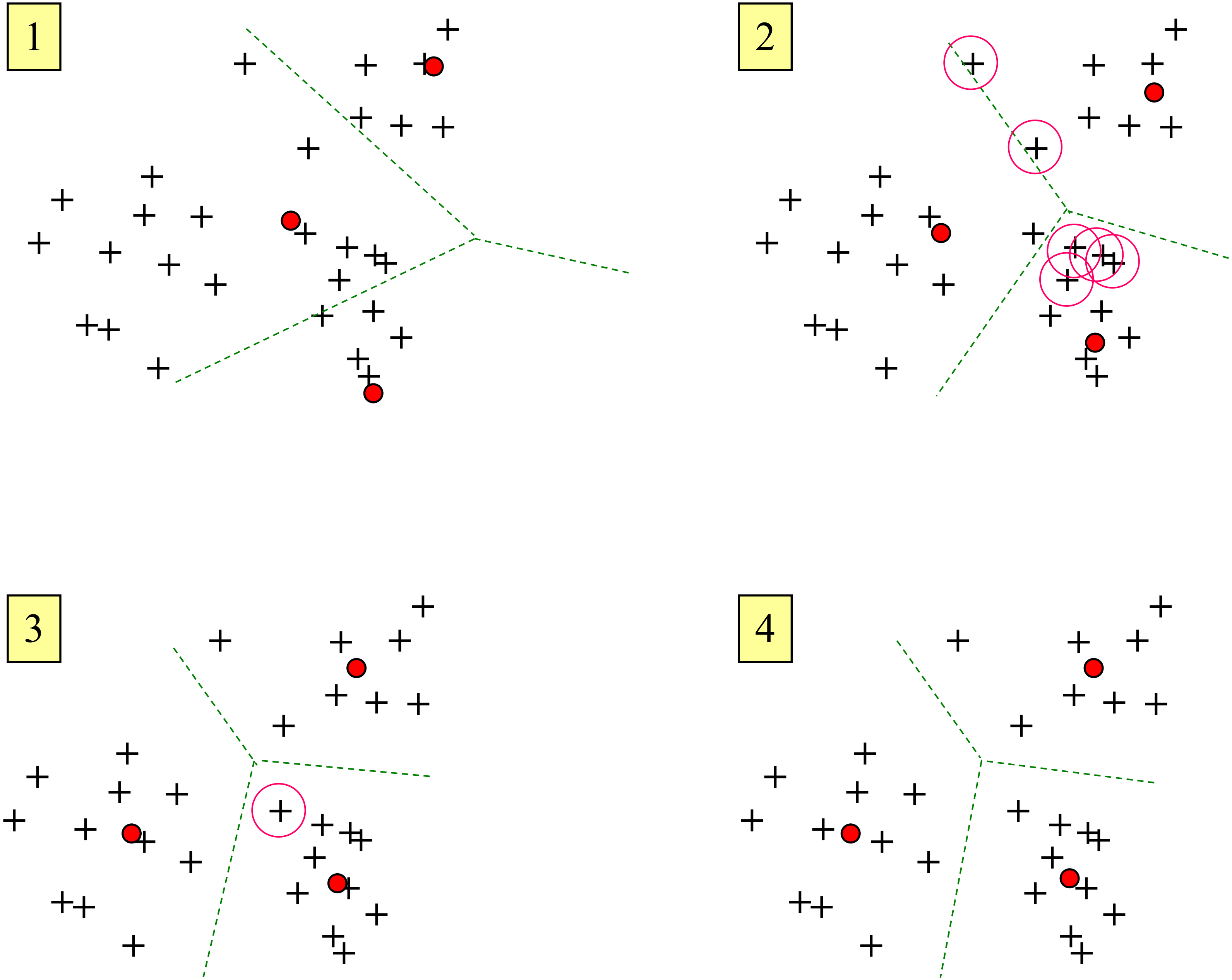
    $D_j \leftarrow \{\mathbf{x} \in D | \mathbf{x}$ assigned to cluster $j\}$; ⟵ *Partition defined by assignment*

    $\boldsymbol{\mu}_j = \frac{1}{|D_j|} \sum_{\mathbf{x} \in D_j} \mathbf{x}$; ⟵ *Re-compute the cluster mean*

  end

until no change in $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$;

return $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$;



No change – finished!

K-means demo

Demo: https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

K-means applications

In K-means, we simply take a cluster centroid (exemplar) to be the mean of points in the cluster.

- Document clustering
- House clustering based on price, square footage, #bedrooms, etc.
- Image segmentation (pixel clustering based on RGB values and location)
- … …

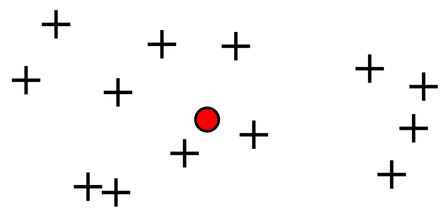
?

K-medoids algorithm

- In some problems, the cluster exemplars are required to be data points (from the training data)
  - As opposed to using the mean of the cluster points, for example, since the mean is most likely not a point in the data set
- The concept of *medoid* is useful here – the medoid of a set of points is the point with the minimal average dissimilarity (distance) to all other points in the set
  - Using some distance metric: Euclidian, L1, etc.
  - This is a generalization of the concept of median to multiple dimensions
- K-means can be modified to use data points as exemplars rather than means, by instead computing the cluster medoids

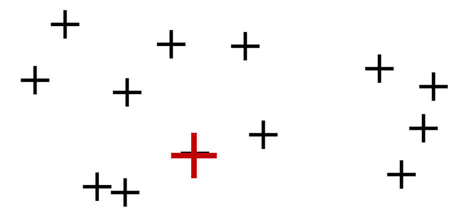
K-medoids algorithm

Cluster mean



Location that minimizes the sum of squared distances to points

Cluster medoid



Point that minimizes the sum of squared distances to points

K-medoids algorithm

Algorithm KMedoids(D, K, Dis) – K-medoids clustering using arbitrary distance metric Dis.

Input : data $D \subseteq \mathscr{X}$; number of clusters $K \in \mathbb{N}$;
distance metric $\mathrm{Dis} : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$.

Output : K medoids $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K \in D$, representing a predictive clustering of $\mathscr{X}$.

randomly pick K data points $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K \in D$;

repeat

  assign each $\mathbf{x} \in D$ to $\arg\min_j \mathrm{Dis}(\mathbf{x}, \boldsymbol{\mu}_j)$;

  for $j = 1$ to K **do**

    $D_j \leftarrow \{\mathbf{x} \in D | \mathbf{x}$ assigned to cluster $j\}$;

    $\boldsymbol{\mu}_j = \arg\min_{\mathbf{x} \in D_j} \sum_{\mathbf{x}' \in D_j} \mathrm{Dis}(\mathbf{x}, \mathbf{x}')$; $\longleftarrow$ *Re-compute the cluster medoid*

  end

until no change in $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$;

return $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$;

Summary: Distance methods and clustering

- Similarity is a function of distance
- Euclidian distance may not always be the right choice of distance metric
- Nearest neighbor methods assign classes/clusters based on distances to points or exemplars, not based on computed boundaries
- For good clustering, we want high within-class (intra-class) similarity and low between-class (inter-class) similarity
- The scatter matrix is an important structure in clustering
- The K-means algorithm (and variations) is widely used