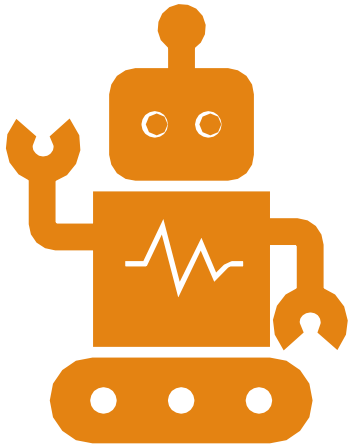


CS341
Artificial Intelligence

Lecture 10

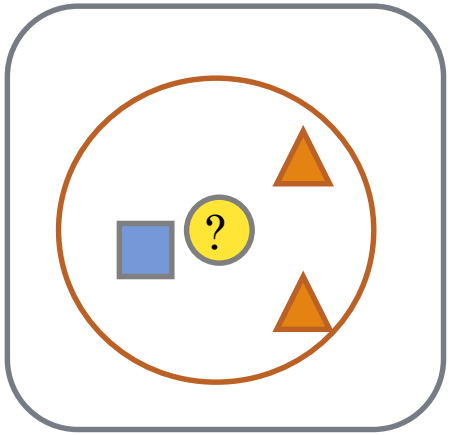
DR. HEBA MOHSEN

Supervised Learning

- A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples.
- An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way.
- **Supervised learning algorithms:**
  - ID3 Decision Tree
  - K-Nearest Neighbor (K-NN)

K-Nearest Neighbor (K-NN)

K nearest neighbors stores all available cases and classifies new cases based on a similarity measure (e.g distance function)



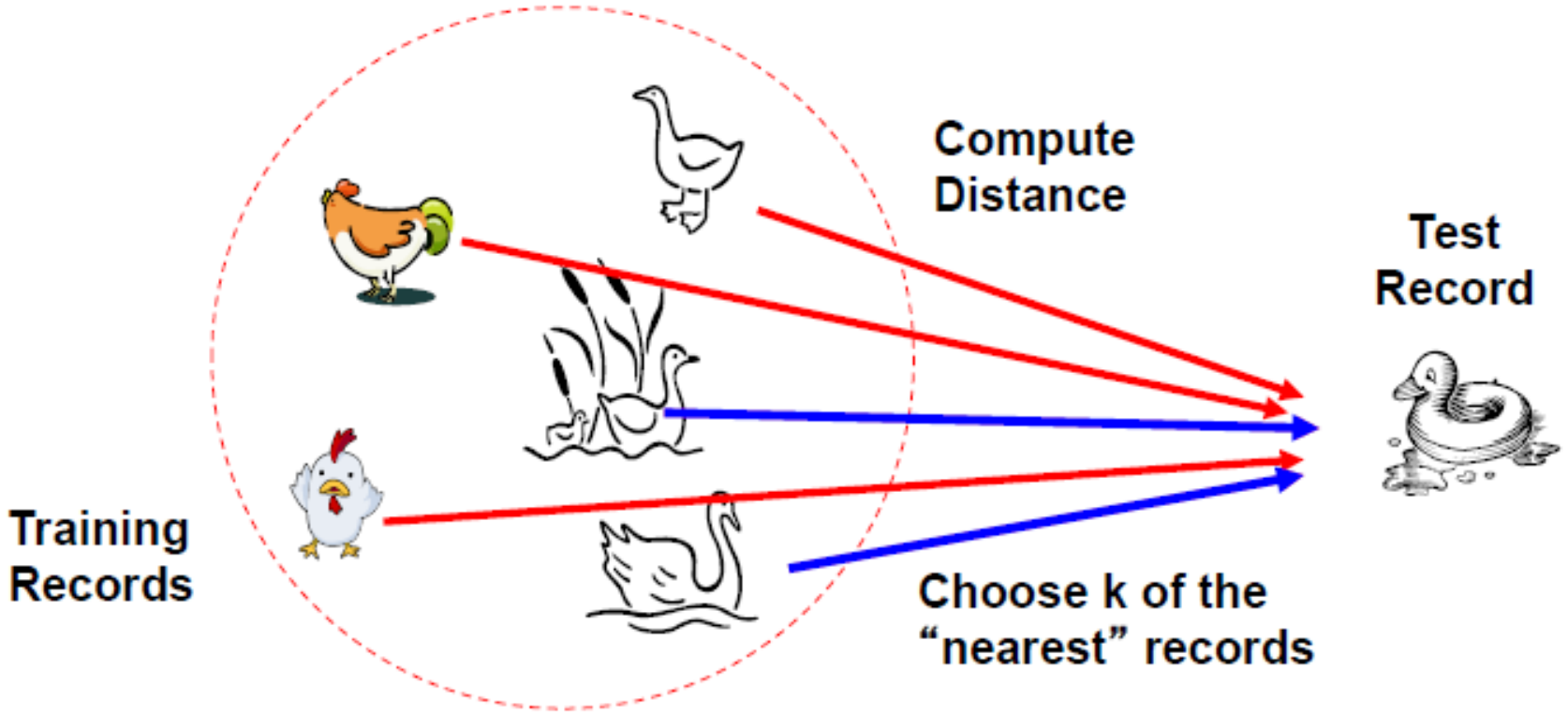
Requires 3 things:

- **Training Data**
- **Distance metric:** to compute distance between records
- **The value of *k***
  - the number of nearest neighbors to retrieve from which to get majority class

To classify an unknown object:

- An object (a new instance) is classified by a majority votes for its neighbor classes
- The object is assigned to the most common class amongst its K nearest neighbors. (measured by a distance function)

Distance Measure



Common Distance Metrics

There are different ways to compute the distance between records

The proper metric to use is always going to be determined by the data-set and the classification task.

Popular ones:

1. **Euclidean distance**
2. **Hamming distance**

1. Euclidean Distance

- Euclidean distance is the most common use of distance.
- In most cases when people said about distance, they will refer to Euclidean distance. It is also known as simply distance.
- It is the root of square differences between coordinates of a pair of objects.

$$d_{ij} = \sqrt{\sum_{k=1}^{n} \left(x_{ik} - x_{jk} \right)^2}$$

Features	Coord1	Coord2	Coord3	Coord4
Object A	0	3	4	5
Object B	7	6	3	-1

$$d_{BA} = \sqrt{(0-7)^2 + (3-6)^2 + (4-3)^2 + (5+1)^2}$$

$$= \sqrt{49+9+1+36} = 9.747$$

2. Hamming Distance

Hamming distance is a string metric used to compare strings

The Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different.

It measures the minimum number of substitutions required to change one string into the other, or the minimum number of errors that could have transformed one string into the other.

Calculating the Hamming Distance

The Hamming distance between:

- **"karolin"** and **"kathrin"** is 3.
- **"karolin"** and **"kerstin"** is 3.
- **1011101** and **1001001** is 2.
- **2173896** and **2233796** is 3.

Example with K=3

Customer	Age	Income	No. credit cards	Class
George	35	35K	3	No
Rachel	22	50K	2	Yes
Steve	63	200K	1	No
Tom	59	170K	1	No
Anne	25	40K	4	Yes
John	37	50K	2	YES

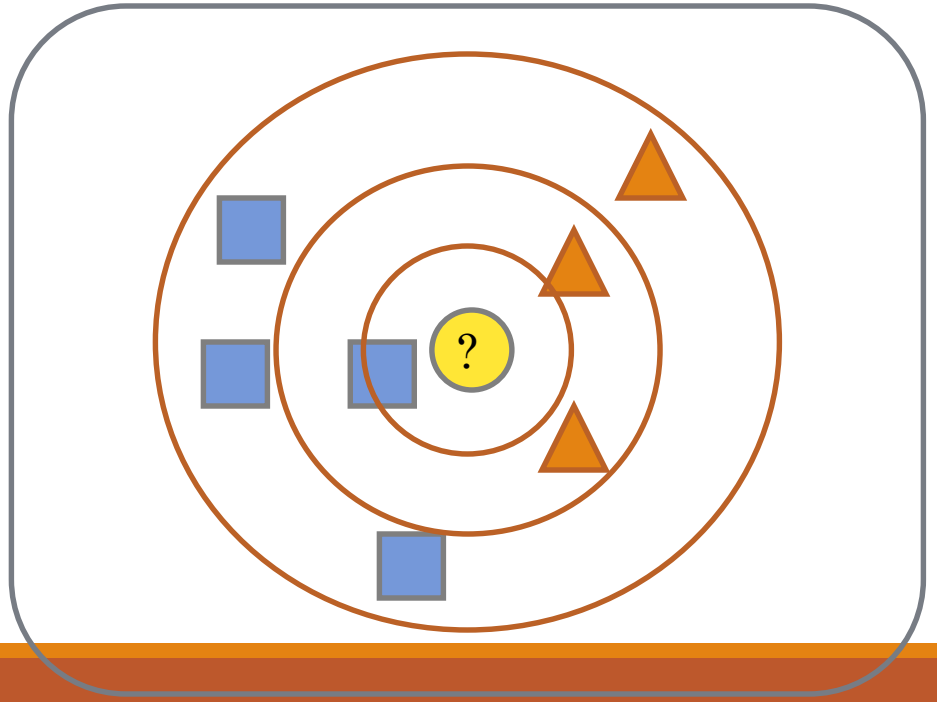
Distance from John
sqrt $[(35-37)^2+(35-50)^2+(3-2)^2]=15.16$
sqrt $[(22-37)^2+(50-50)^2+(2-2)^2]=15$
sqrt $[(63-37)^2+(200-50)^2+(1-2)^2]=152.23$
sqrt $[(59-37)^2+(170-50)^2+(1-2)^2]=122$
sqrt $[(25-37)^2+(40-50)^2+(4-2)^2]=15.74$

The Value of K

Choose an **odd value** for *k*, to eliminate ties.

However, if k is too small, sensitive to noise points.

If k is too large, neighborhood may include points from other classes.



$K = 1$:

- Belongs to square class

$K = 3$:

- Belongs to triangle class

$K = 7$:

- Belongs to square class

K Nearest Neighbor Issues

Classifying unknown records are relatively expensive:

- Requires distance computation of k-nearest neighbors.
- Computationally intensive, especially when the size of the training set grows.

Accuracy can be severely degraded by the presence of noisy or irrelevant features.

Unsupervised Learning

Unsupervised learning eliminates the teacher and requires that the learner form and evaluate concepts on its own.

In machine learning, the problem of unsupervised learning is that of trying to find hidden structure in unlabeled data.

The data have no target attribute. We want to explore the data to find some intrinsic structures in them.

Clustering

Clustering is the classification of objects into different groups so that the data in each subset (ideally) share some common trait - often according to some defined distance measure.

- The objects that are similar to (near) each other in one cluster and objects that are very different (far away) from each other into different clusters.

- Example: groups people of similar sizes together to make "small", "medium" and "large" T-Shirts.

Clustering algorithms: K-Means

K-means Clustering

The *k*-means algorithm is an algorithm to cluster *n* objects based on attributes into *k* partitions, where $k < n$.

- Each cluster has a cluster **center**, called **centroid**.
- *k* is a positive value specified by the user based on the final application. For example we might set k=26 in a handwritten English letter application.

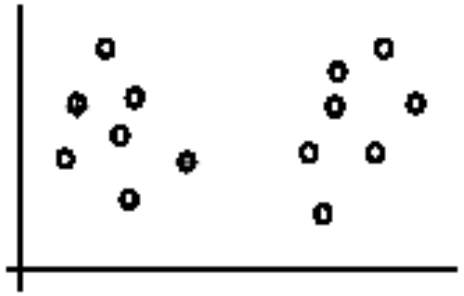
K-means Algorithm

Given *k*, the *k-means* algorithm works as follows:

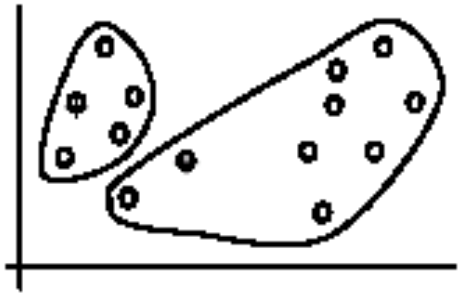
1. Randomly choose *k* data points (seeds) to be the initial centroids, cluster centers
2. Assign each data point to the closest centroid
3. Re-compute the centroids using the current cluster memberships.
4. If a convergence criterion is not met, go to step (2).

Stopping/Convergence Criterion

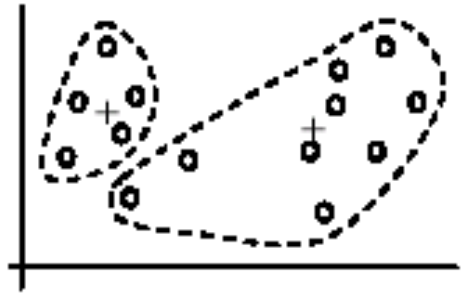
1. No (or minimum) re-assignments of data points to different clusters.
2. No (or minimum) change of centroids.



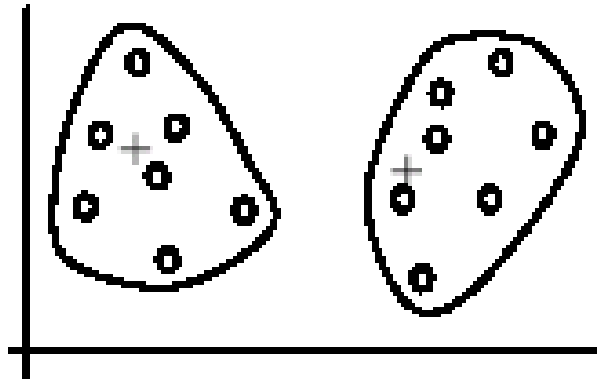
(A). Random selection of k centers



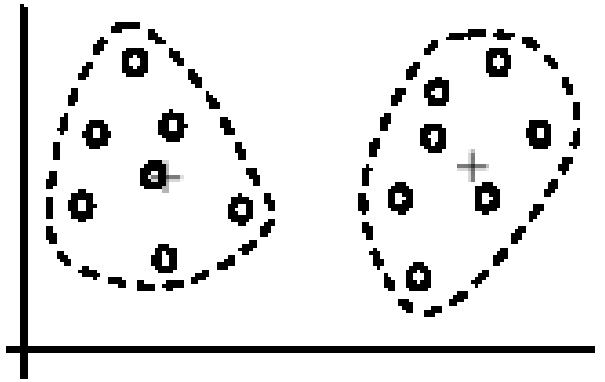
Iteration 1: (B). Cluster assignment



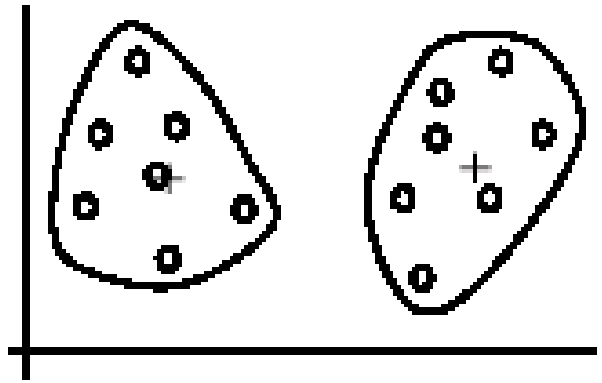
(C). Re-compute centroids



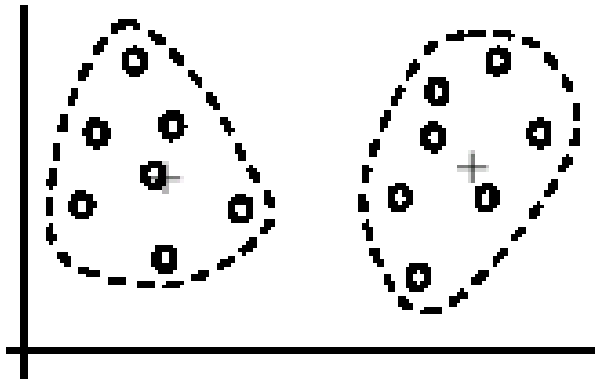
Iteration 2: (D). Cluster assignment



(E). Re-compute centroids



Iteration 3: (F). Cluster assignment



(G). Re-compute centroids

Calculating the distance

The *k-means* algorithm **can be used for any application data set** where **the mean can be defined and computed**. In the Euclidean space, the mean of a cluster is computed with:

$$m_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$$

Where $|C_j|$ is the number of data points in cluster C_j

Calculating the distance

The distance from one data point X_i to a mean (centroid) m_j is computed with:

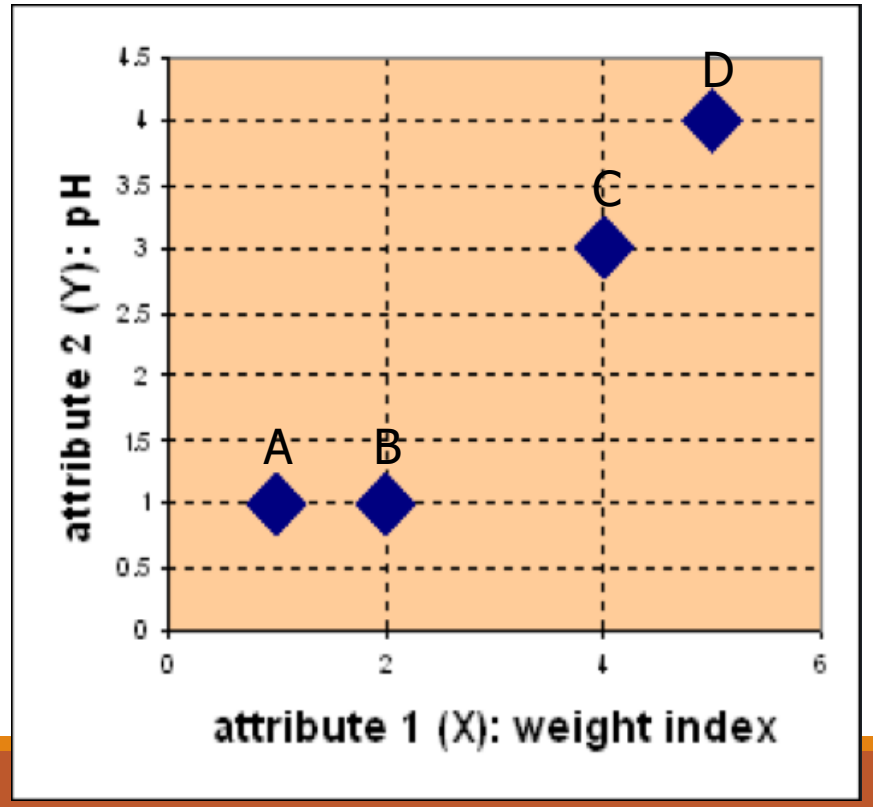
$$dist(\mathbf{x}_i, \mathbf{m}_j) = \| \mathbf{x}_i - \mathbf{m}_j \|$$

$$= \sqrt{(x_{i1} - m_{j1})^2 + (x_{i2} - m_{j2})^2 + \ldots + (x_{ir} - m_{jr})^2}$$

Example

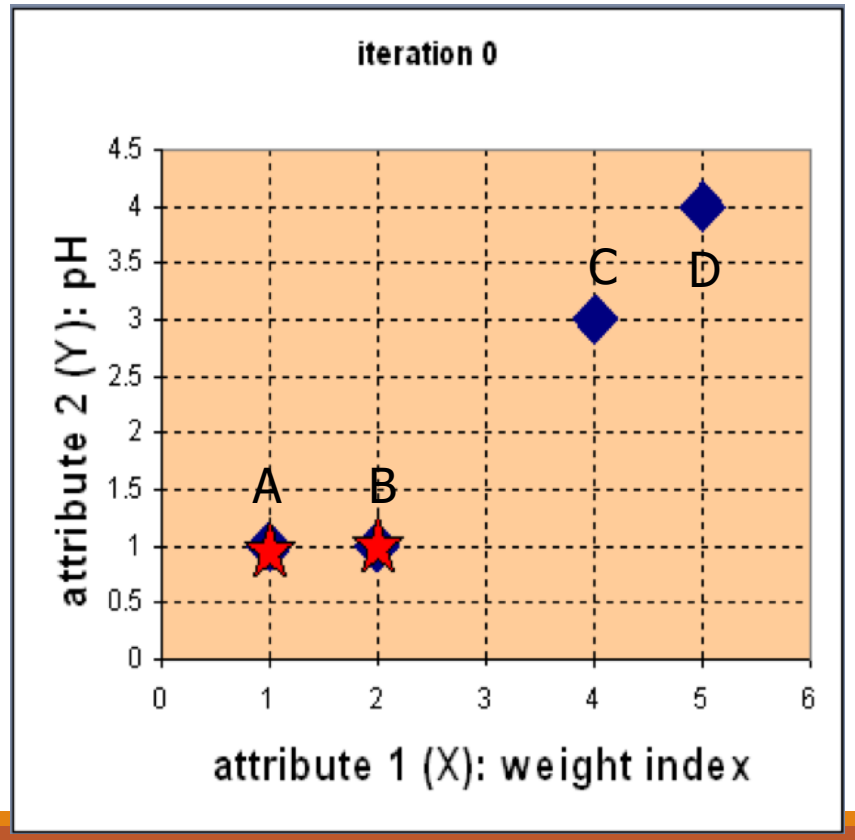
Suppose we have **4 types of medicines** and each has two attributes (weight index and pH). Our goal is to group these objects into ***two*** groups of medicine.

Medicine	Weight Index	pH
A	1	1
B	2	1
C	4	3
D	5	4



Example

Step 1: Use initial seed points for partitioning



c_1= A= (1,1) and c_2= B = (2,1)

Euclidean distance

$$\mathbf{D}^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \quad \begin{matrix} \mathbf{c}_1 = (1,1) & group-1 \\ \mathbf{c}_2 = (2,1) & group-2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \quad \begin{matrix} X \\ Y \end{matrix}$$

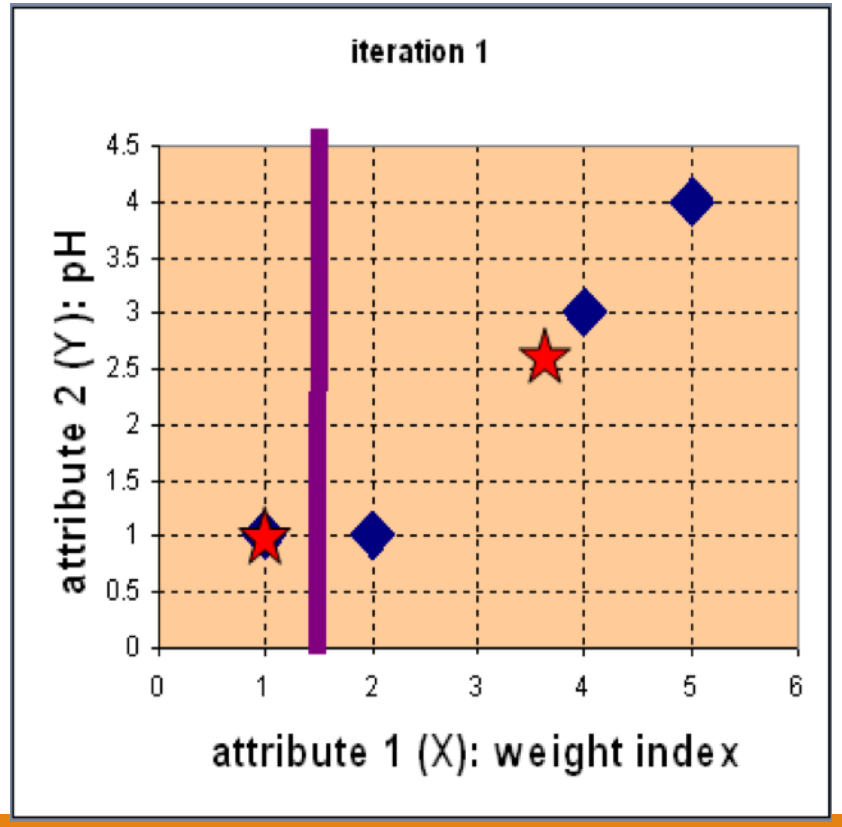
$$d(D,c_1) = \sqrt{(5-1)^2 + (4-1)^2} = 5$$

$$d(D,c_2) = \sqrt{(5-2)^2 + (4-1)^2} = 4.24$$

Assign each object to the cluster with the nearest seed point

Example

Step 2: Compute new centroids of the current partition



Knowing the members of each cluster, now we compute the new centroid of each group based on these new memberships.

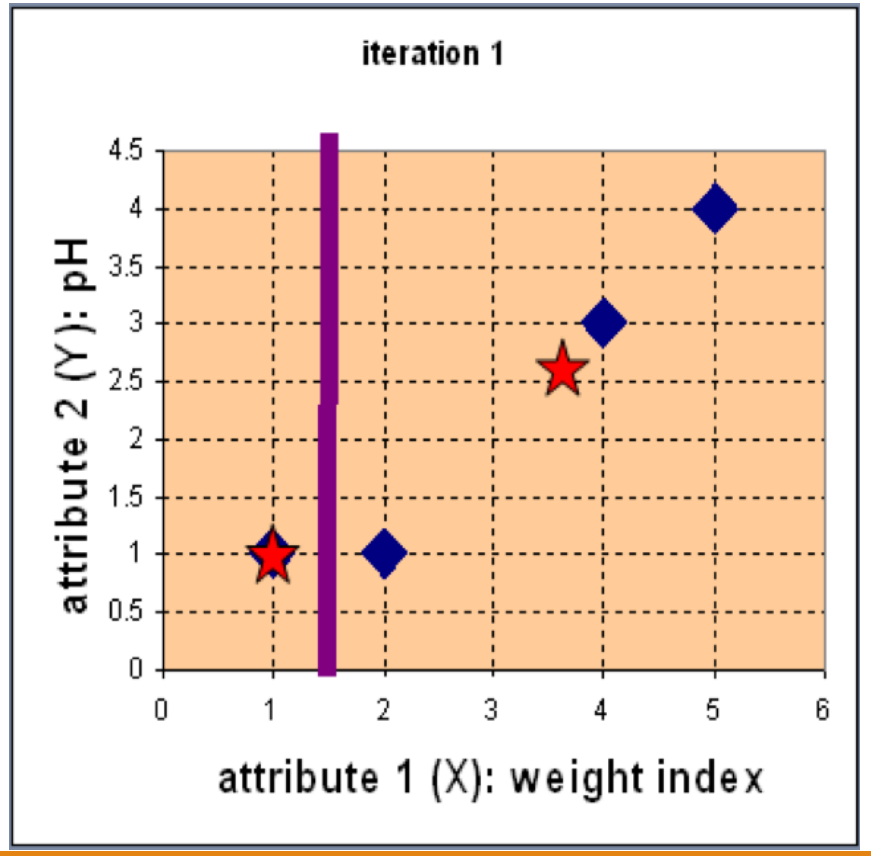
$$c_1 = (1,\ 1)$$

$$c_2 = \left(\frac{2+4+5}{3},\ \frac{1+3+4}{3}\right)$$

$$= (\frac{11}{3},\ \frac{8}{3})$$

Example

Step 2: Renew membership based on new centroids



Compute the distance of all objects to the new centroids

$$D^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \quad \begin{matrix} c_1 = (1,1) & group-1 \\ c_2 = (\frac{11}{3}, \frac{8}{3}) & group-2 \end{matrix}$$

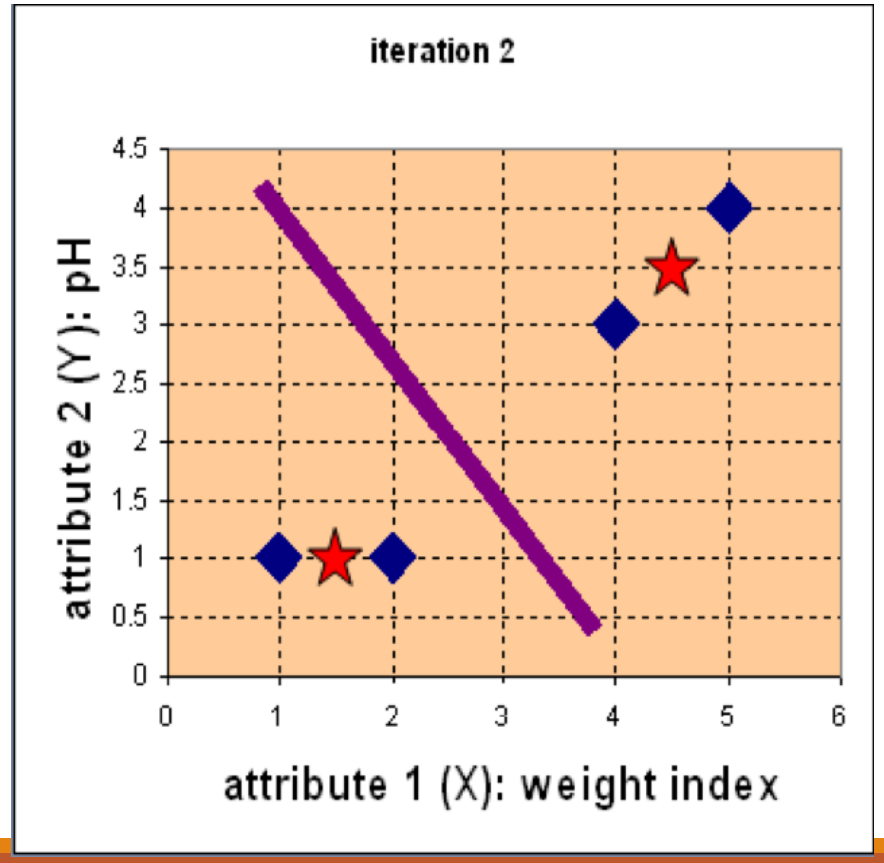
$$\begin{matrix} A & B & C & D \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \begin{matrix} X \\ Y \end{matrix}$$

Assign the membership to objects

Example

Step 3: Repeat the first two steps until its convergence

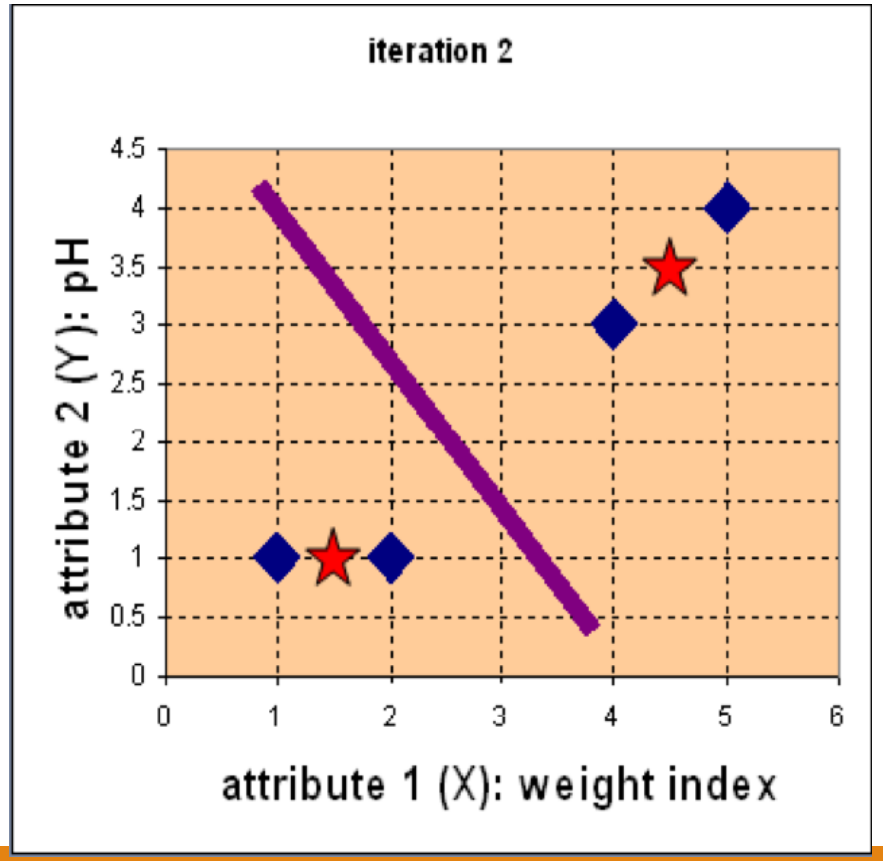


Knowing the members of each cluster, now we compute the new centroid of each group based on these new memberships.

$$c_1 = \left(\frac{1+2}{2}, \frac{1+1}{2}\right) = (1\frac{1}{2}, 1)$$

$$c_2 = \left(\frac{4+5}{2}, \frac{3+4}{2}\right) = (4\frac{1}{2}, 3\frac{1}{2})$$

Step 3: Repeat the first two steps until its convergence



Compute the distance of all objects to the new centroids

$$\mathbf{D}^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \quad \begin{matrix} \mathbf{c}_1 = (1\frac{1}{2}, 1) & group-1 \\ \mathbf{c}_2 = (4\frac{1}{2}, 3\frac{1}{2}) & group-2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \begin{matrix} X \\ Y \end{matrix}$$

Stop due to no new assignment. Membership in each cluster no longer change.

Exercise

Use the **K-means** algorithm to cluster the following 8 data point into **3 clusters**:

A1=(2,10),
A2=(2,5),
A3=(8,4),
A4=(5,8),
A5=(7,5),
A6=(6,4),
A7=(1,2),
A8=(4,9).