Clustering

JELENA JOVANOVIĆ

Email: jeljov@gmail.com

Web: http://jelenajovanovic.net

OUTLINE

- What is clustering?
- Application domains
- K-Means clustering
  - Understanding it through an example
  - The K-Means algorithm
  - Some challenging issues
  - An example in WEKA

WHAT IS CLUSTERING?

Clustering is an unsupervised learning task

- its input is a set of instances to be grouped based on their similarity
- there is no data about the desired/correct group for any of the input instances

What is clustering?

It is about grouping objects in such a manner that for each object the following is true:

- the object is more *similar* to the objects from its group (cluster), than to objects from other groups (clusters)

Similarity between objects is computed using certain

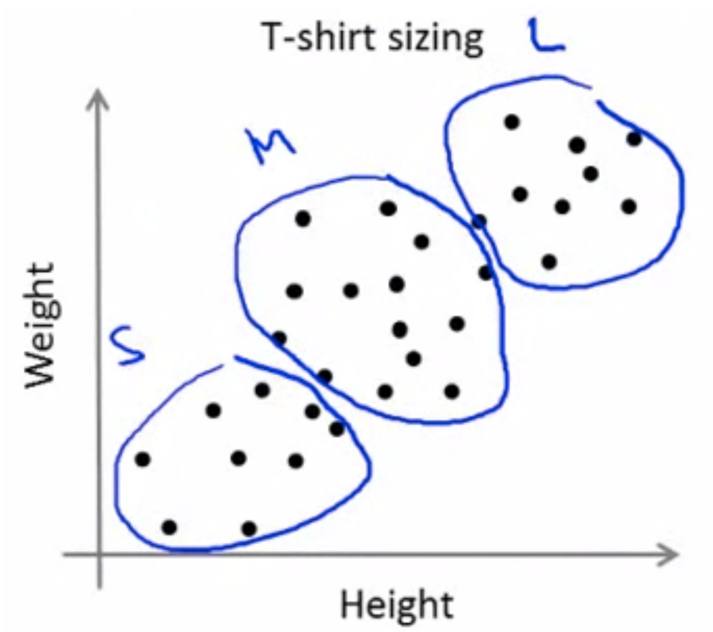
- similarity measure (e.g., Cosine similarity), or
- distance measure (e.g., Euclidian distance)

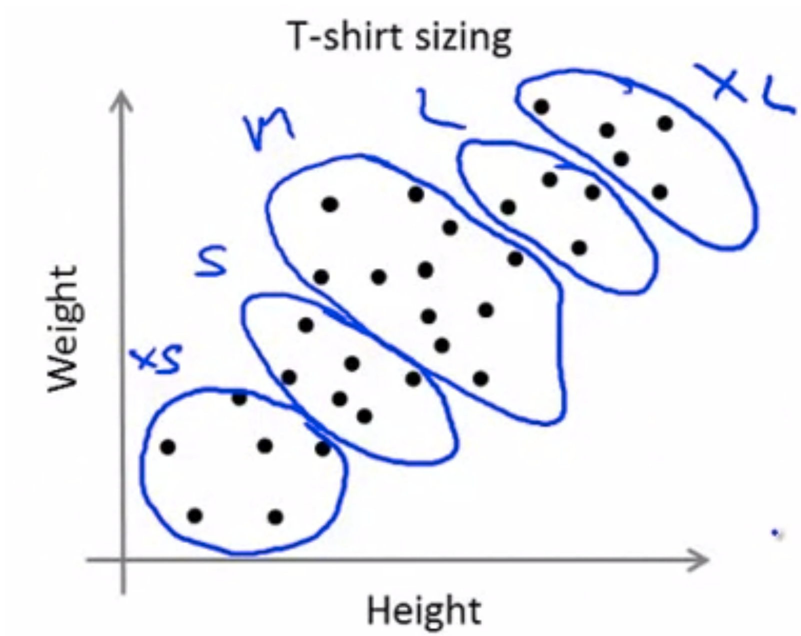
What is clustering?

Unlike the classification task, for this task, there is no unique “correct” solution

- how good/suitable a solution is, depends upon the specific domain and application case – the same solution might be differently evaluated in different application cases
- if it is to be done properly, domain experts need to evaluate the solution(s) produced by the model

An example illustrating different valid solutions for the same input dataset





APPLICATION DOMAINS

- Market segmentation
- Detection of groups/communities in social networks
- Pattern mining in the user tracking data
- Grouping of objects (e.g., images or documents) based on their common characteristics
- …

K-MEANS ALGORITHM

K-MEANS

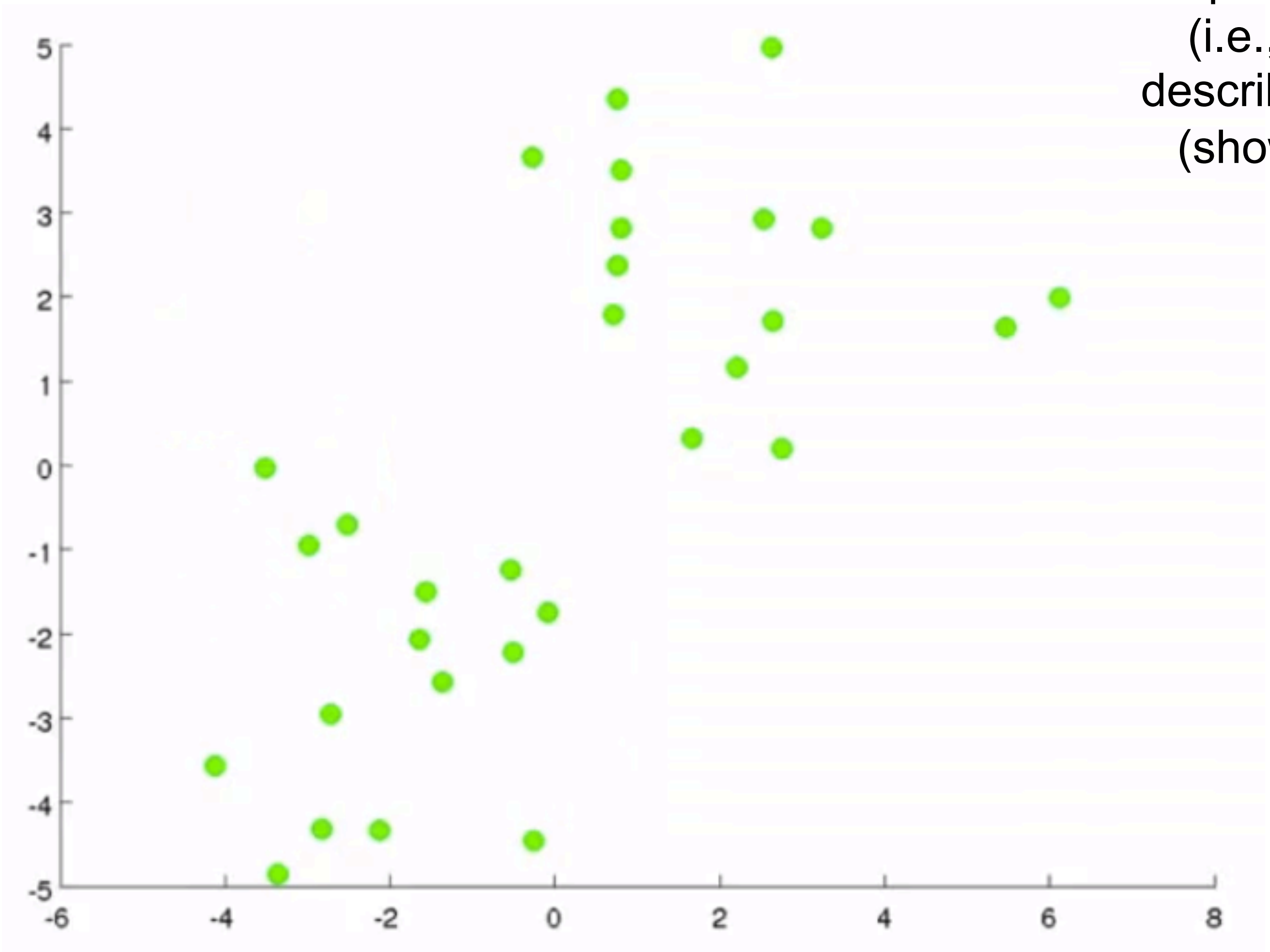
One of the simplest and most widely known and used clustering algorithm

It can be best understood through examples, so we will first have a look at an example

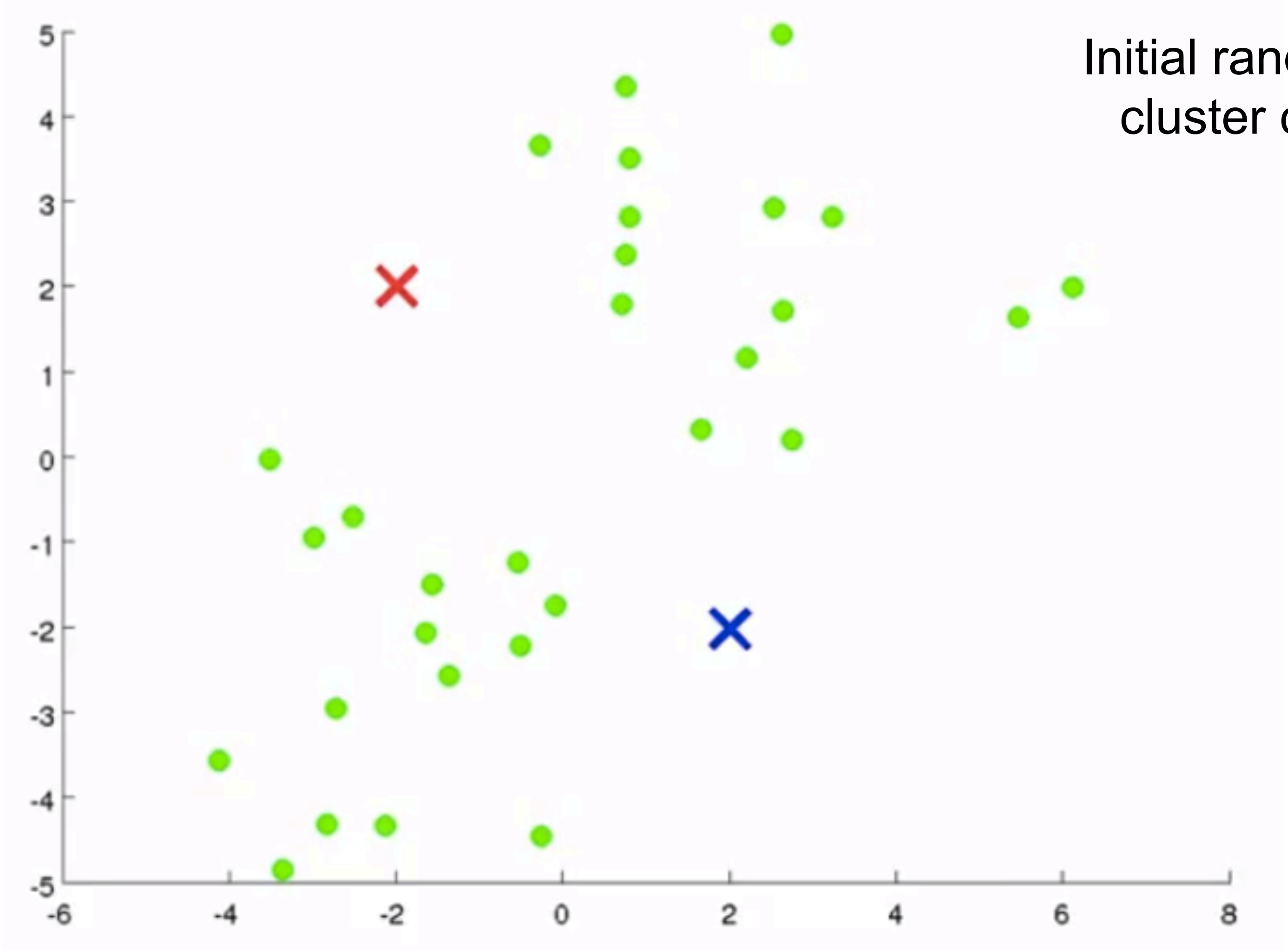
The example is taken from the course: https://www.coursera.org/course/ml

K-MEANS: AN EXAMPLE

Let’s suppose the diagram presents the input data (i.e., a set of instances), described with 2 attributes (shown on x and y axes)

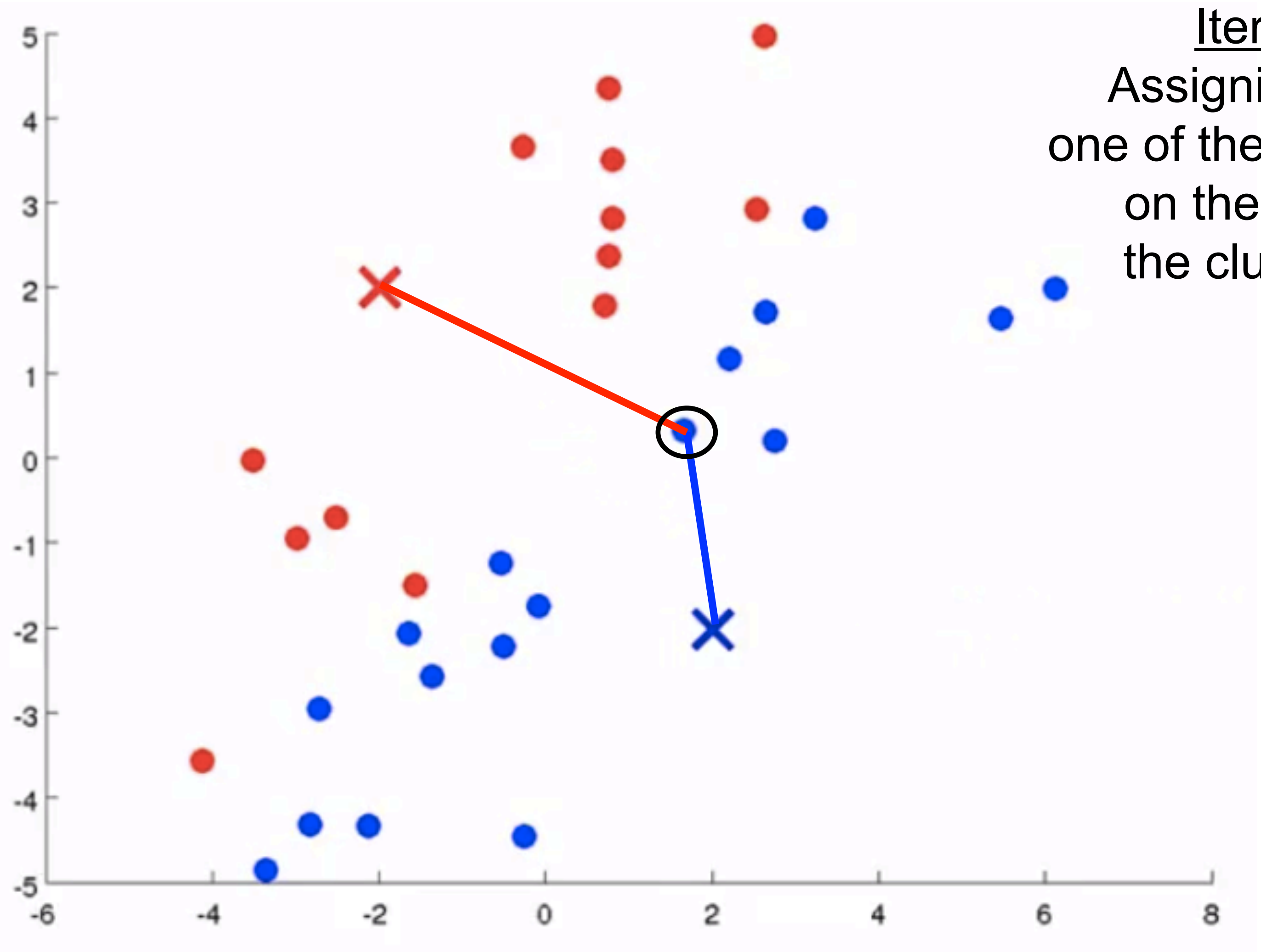


K-MEANS: AN EXAMPLE



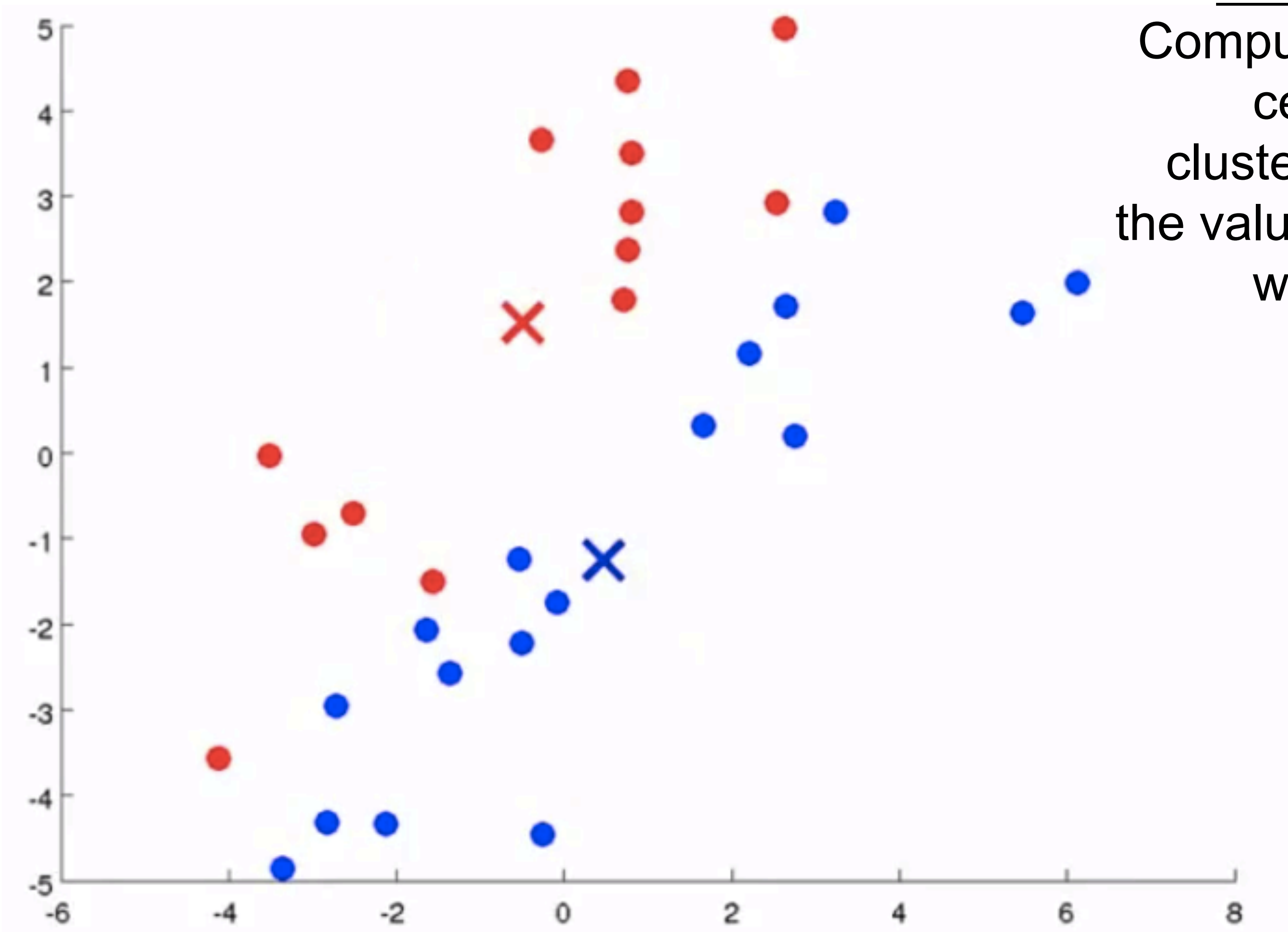
<u>Initialization:</u>
Initial random selection of cluster centroids (K = 2)

K-MEANS: AN EXAMPLE



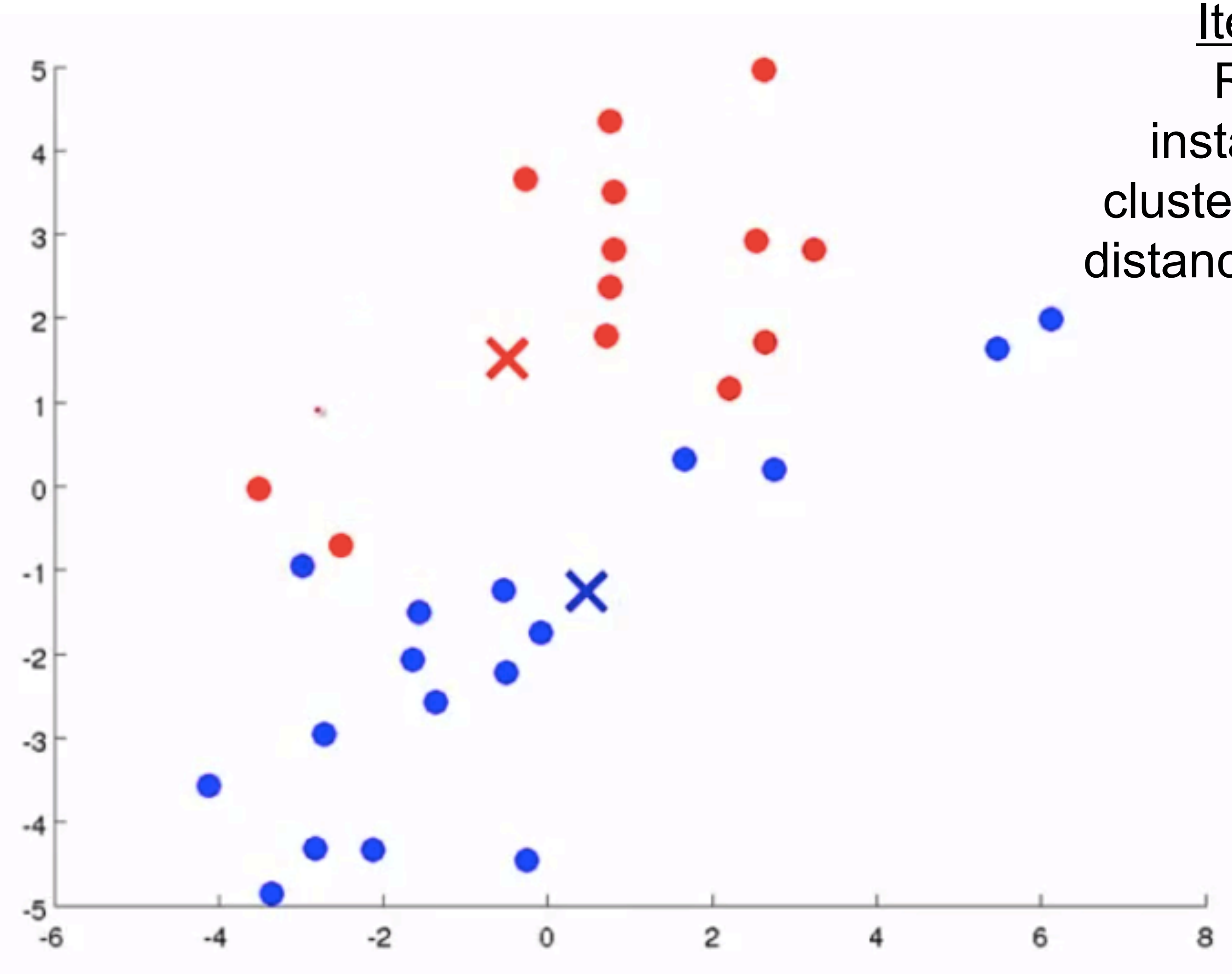
<u>Iteration 1, Step 1</u>:
Assigning instances to one of the clusters based on their distance from the clusters' centroids

K-MEANS: AN EXAMPLE



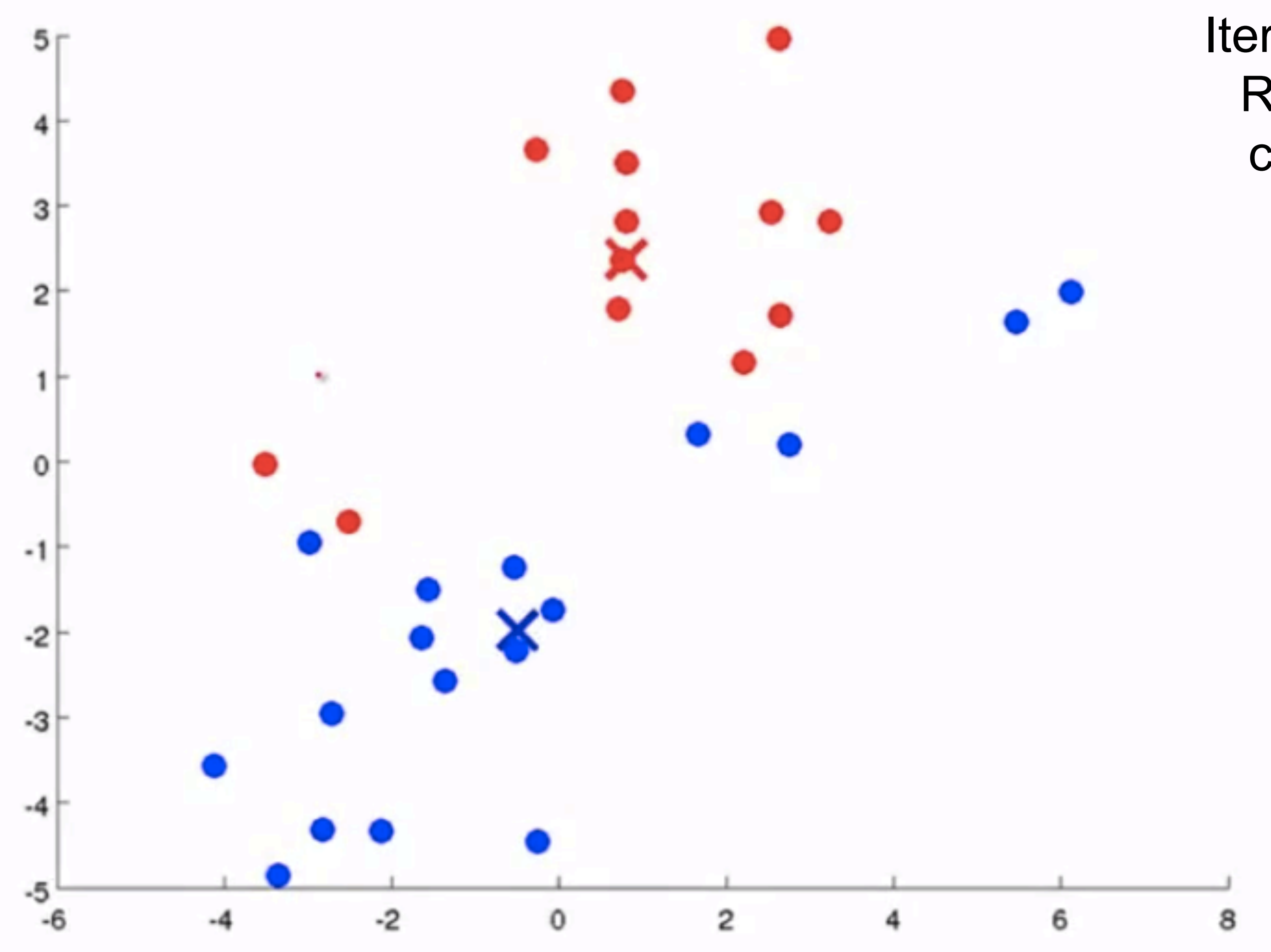
<u>Iteration 1, Step 2:</u>
Computation of a new centroid for each cluster, by averaging the values of instances within the cluster

K-MEANS: AN EXAMPLE



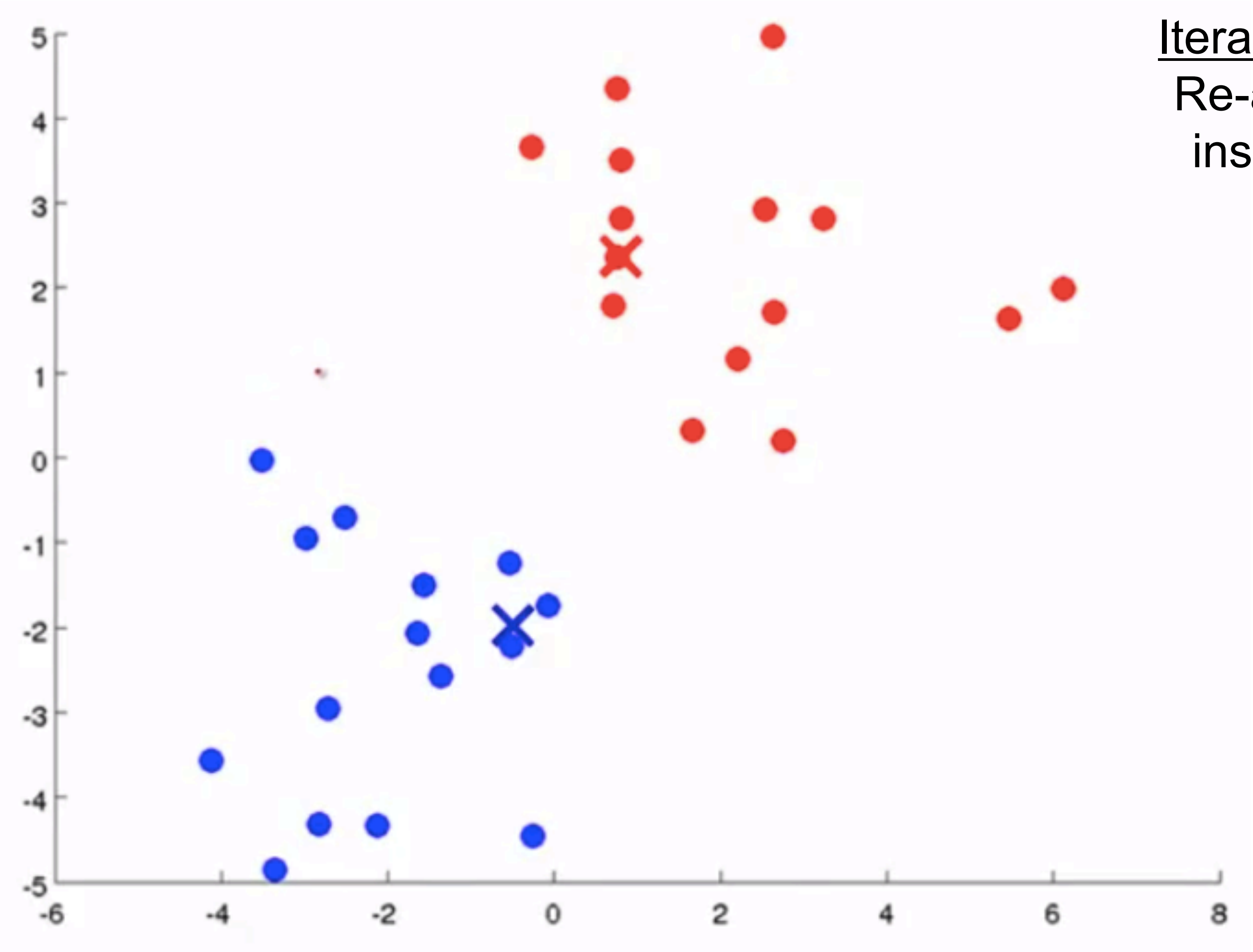
<u>Iteration 2, Step 1:</u>
Re-assignment of instances across the clusters based on their distance from the (new) cluster centroids

K-MEANS: AN EXAMPLE



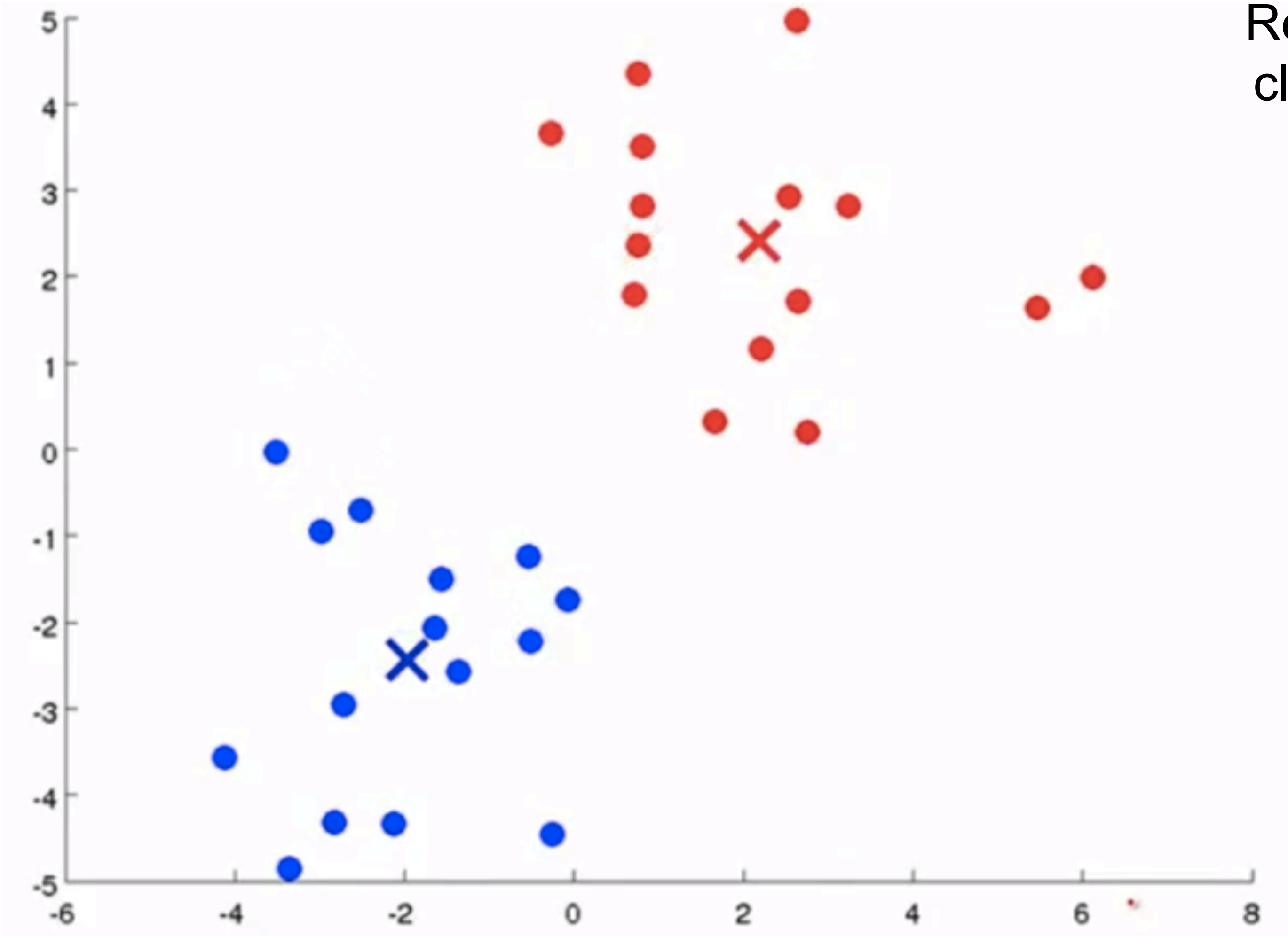
Iteration 2, Step 2:
Re-calculation of
cluster centroids

K-MEANS: AN EXAMPLE



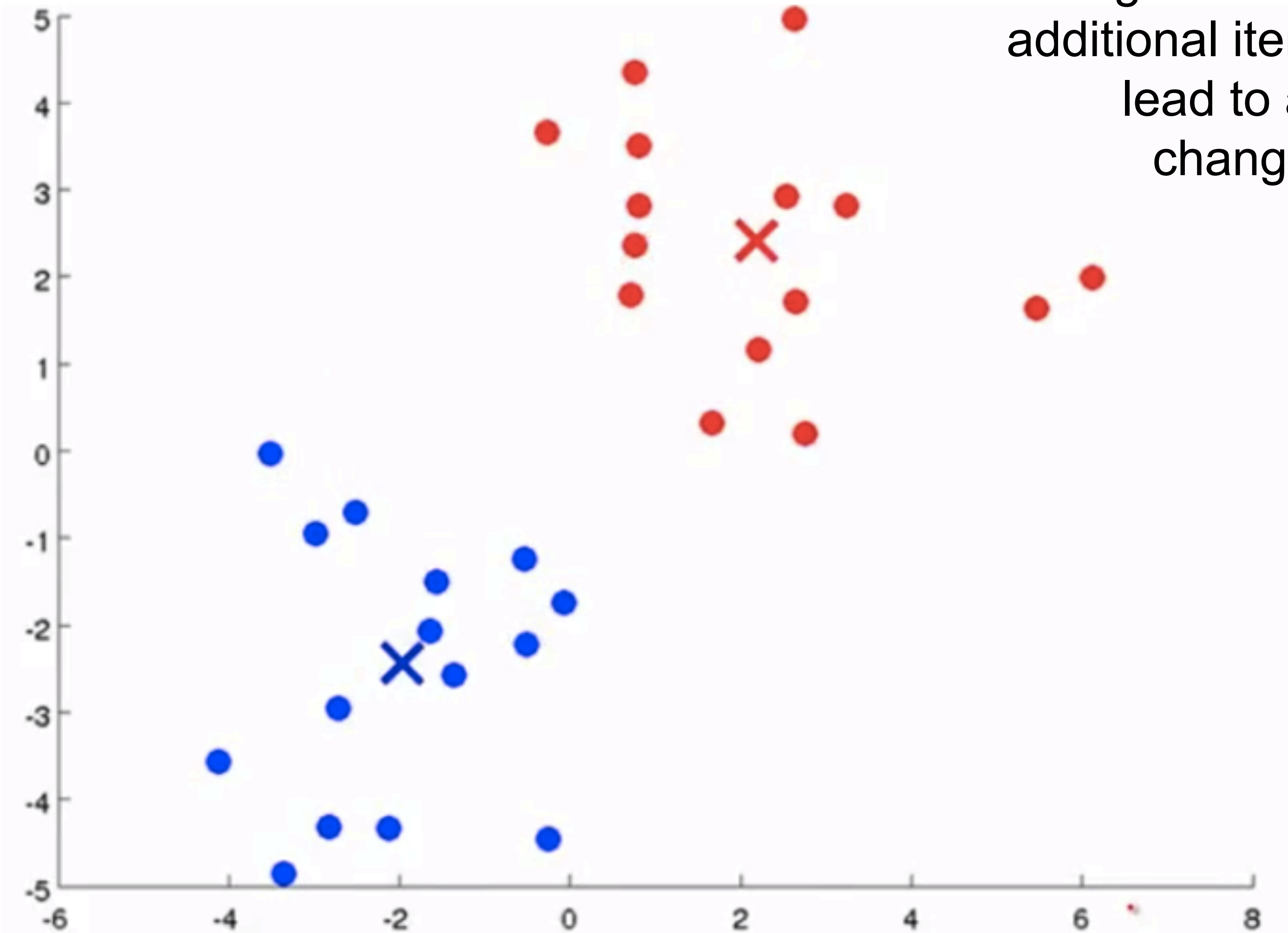
Iteration 3, Step 1: Re-assignment of instances across the clusters

K-MEANS: AN EXAMPLE



<u>Iteration 3, Step 2:</u>
Re-calculation of cluster centroids

K-MEANS: AN EXAMPLE



The algorithm is converging: additional iterations will not lead to any significant change; the process terminates

K-MEANS: THE ALGORITHM

Input:

- *K* – the number of clusters
- (unlabeled) training set with *m* instances; each instance in this set is a vector described with *n* attributes (x_1, x_2, …, x_n)
- *max* - max number of iterations (optional parameter)

K-MEANS: THE ALGORITHM

Steps:

1) Initial, random selection of a centroid for each cluster
   - centroids are chosen from the training set, i.e., *K* instances are randomly taken from the training set and set as centroids

2) Repeat until the algorithm starts converging or the number of iterations reaches *max*:
   1) *Cluster assignment*: for each instance *i* from the training set, *i* = 1,m, identify the closest centroid and assign the instance to the corresponding cluster
   2) *Repositioning of centroids*: for each cluster, compute a new centroid by averaging the values of instances assigned to that cluster

K-MEANS: THE COST FUNCTION

The objective of the K-means algorithm is to *minimize the cost function **J***:

$$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K) = \frac{1}{m} \sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2$$

$x^{(i)}$ – *i*-th instance in the training dataset, i=1,m

$c^{(i)}$ – index of the cluster to which the instance $x^{(i)}$ is currently assigned

μ_j – centroid of the cluster *j*, j=1,K

$\mu_{c(i)}$ – centroid of the cluster to which the instance $x^{(i)}$ has been assigned

This function is also known as *distortion function*

K-MEANS: THE COST FUNCTION

$$\min_{\substack{c^{(1)},\dots,c^{(m)},\\ \mu_1,\dots,\mu_K}} J(c^{(1)},\dots,c^{(m)},\mu_1,\dots,\mu_K)$$

K-means algorithm minimizes the cost function ***J*** in the following manner:

- the *Cluster assignment* phase minimizes ***J*** with respect to $c^{(1)}, \dots, c^{(m)}$, holding $\mu_1, \dots, \mu_K$ fixed
- the *Repositioning of centroids* phase minimizes ***J*** with respect to $\mu_1, \dots, \mu_K$, holding $c^{(1)}, \dots, c^{(m)}$ fixed

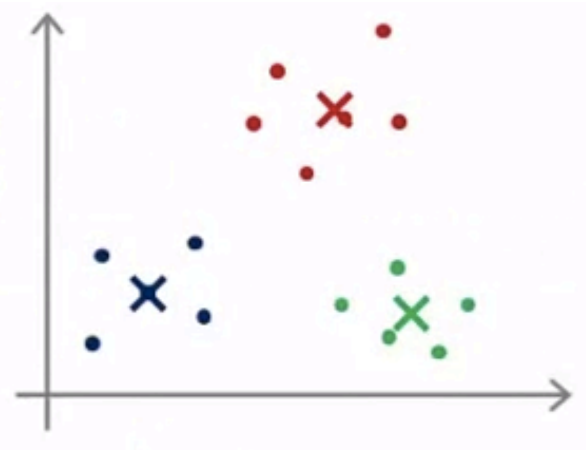
K-MEANS: EVALUATION

Criteria for evaluating the “quality” of the resulting clusters:

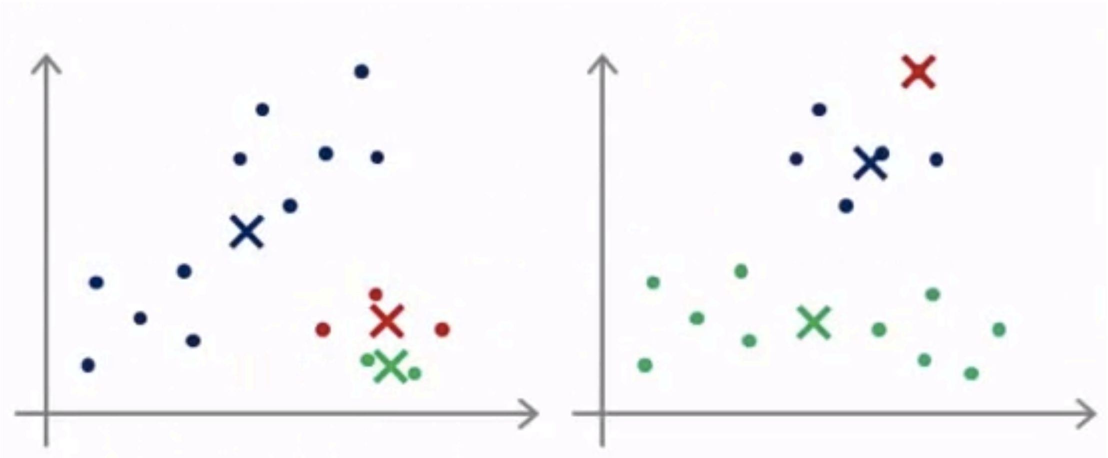
- Distance between the centroids
  - the more distant the centroids are, the lower is the overlap between the clusters, and thus their quality is higher
- St. deviation of instances from the centroid
  - the lower the st. deviation, the more tightly grouped are the instances, and thus, the clusters are considered better
- Within cluster sum of squared errors
  - a quantitative measure for estimating the quality of the clusters
  - we will consider it through an example (slide 23)

K-MEANS:
INITIAL SELECTION OF CENTROIDS

- Depending on how initial cluster centroids are chosen, the K-means algorithm would converge quicker or slower
- "Unlucky" selection of initial centroids may lead K-Means to get stuck in the so called *local optima* and produce poor results
  - this is a local minimum of the *cost function*



"Lucky" initialization



"Unlucky" initialization that leads to a local minimum

K-MEANS:
MULTIPLE RANDOM INITIALIZATIONS

It allows for avoiding situations that lead K-means in a local minimum

Consists of the following:

```
for i = 1 to n {//n is often in the range 50-1000
     Randomly select the initial set of centroids;
     Apply the K-Means algorithm;
     Compute the cost function
}
Choose the instance of the algorithm that produces the
lowest value of the cost function
```

This approach gives good results if the number of clusters is relatively low (2 - 10); should not be used if the number of clusters is higher

K-MEANS: HOW TO CHOOSE K ?

How to determine the number of clusters K?

- In case we have domain knowledge about the phenomenon described by the data
  - Make an assumption about the number of clusters (K) based on the domain knowledge
  - Test the model with K-1, K, K+1 clusters and compare the error*

- If we lack domain knowledge about the studied phenomenon
  - Start with a small number of clusters and in multiple iterations test the model by incrementally increasing the number of clusters
  - In each iteration, compare the error* of the current and the previous model, and when the error reduction becomes insignificant, terminate the process

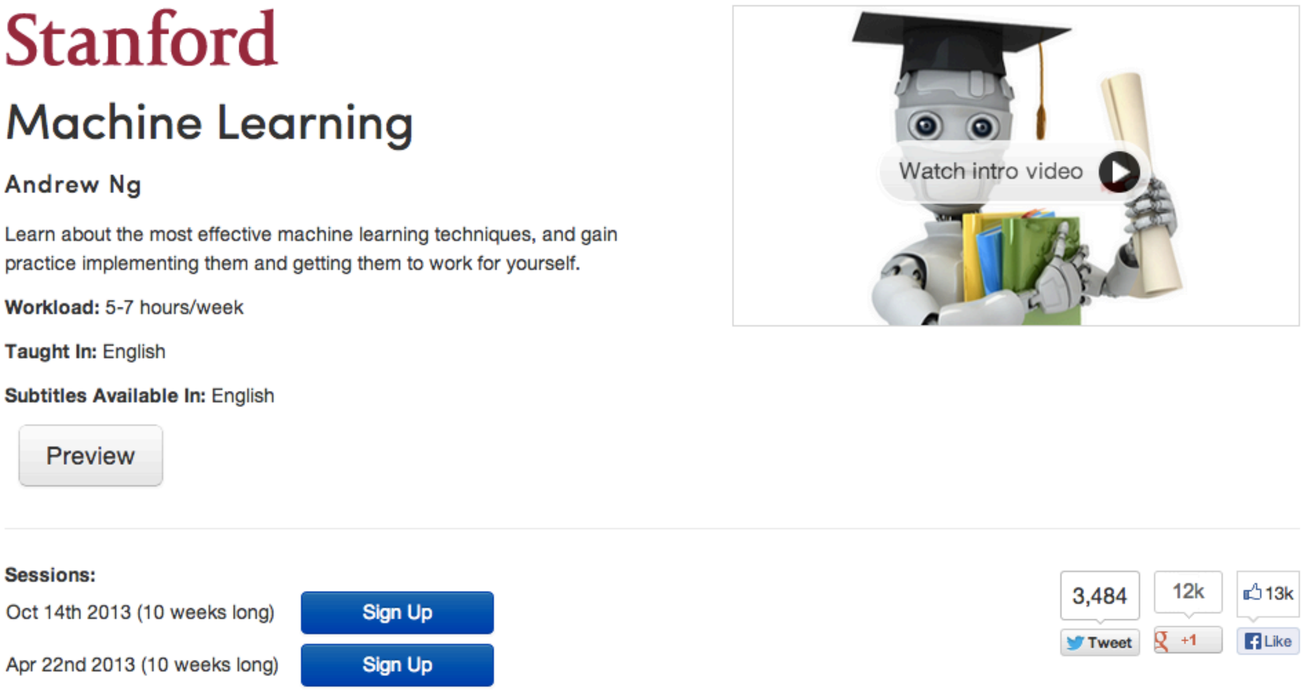
*E.g., within cluster sum of squared errors can be used for the comparison

K-MEANS: AN EXAMPLE IN WEKA

The example we will see is taken from an article, published at the *IBM Developer Works* Web site:

http://www.ibm.com/developerworks/library/os-weka2/

ACKNOWLEDGEMENT AND RECOMMENDATION



Coursera:
https://www.coursera.org/course/ml

Stanford YouTube channel:
http://www.youtube.com/view_play_list?p=A89DCFA6ADACE599

(Anonymous) questionnaire for your critiques, comments, suggestions:

http://goo.gl/cqdp3I