Inspire…Educate…Transform.

# Clustering and IBL

**Jeevan Sreerama**
Sr. Data Scientist, INSOFE

August 8th, 2016

# DISTANCE METRICS

# Desiderata for proximity

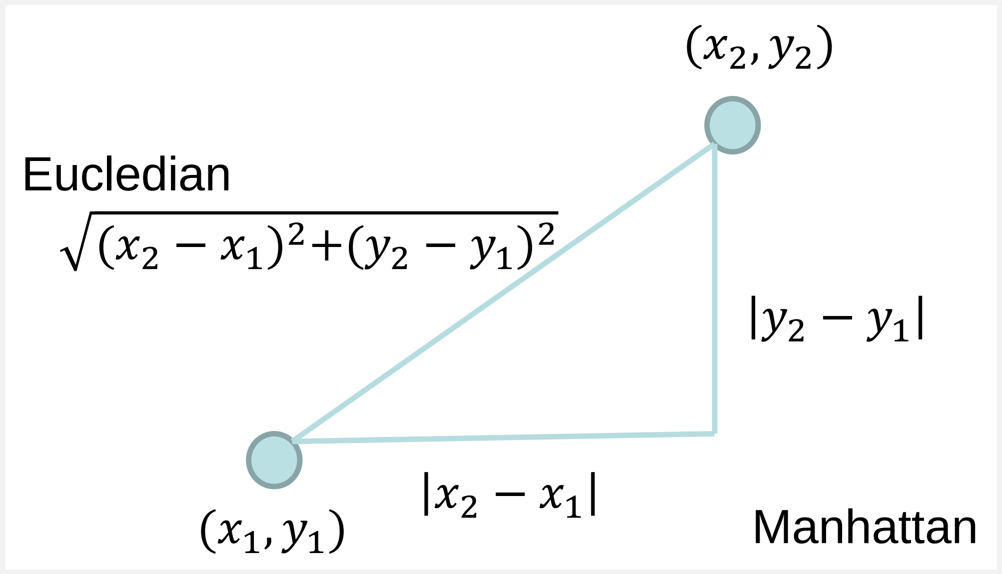
- If $d_1$ is near $d_2$, then $d_2$ is near $d_1$.
- If $d_1$ near $d_2$, and $d_2$ near $d_3$, then $d_1$ is not far from $d_3$.
- No document is closer to $d$ than $d$ itself.

# Distance for numeric attributes

- We denote distance with: $dist(\mathbf{x}_i, \mathbf{x}_j)$
  - Where $\mathbf{x}_i$ and $\mathbf{x}_j$ are data points (vectors)
- Minkowski distance

$$dist(\mathbf{x}_i, \mathbf{x}_j) = ((x_{i1} - x_{j1})^h + (x_{i2} - x_{j2})^h + ... + (x_{ir} - x_{jr})^h)^{\frac{1}{h}}$$

  - Where h is positive integer.
  - h = **2** is **Euclidean** distance
  - h = **1** is **Manhattan** distance



# When to choose what?

- When all attributes have similar scale: (1,2), (2,1)
  - Manhattan = Abs(1-2)+Abs(2-1) = 2
  - Euclidean = $\sqrt{2}$

# Choosing the distance metric

- When attributes have different ranges (10, 100), (50, 500)
  - Manhattan = 440
  - Euclidean= 401.99
- Manhattan is more stable than Euclidean
  - Scaling is better

# Squared Euclidean and Chebyshev distance

- Squared Euclidean distance: Place greater weight on data points that are further apart

$$dist(\mathbf{x}_i, \mathbf{x}_j) = (x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ir} - x_{jr})^2$$

- Chebyshev distance: Two data points are "different" if they are different on any one of the attributes.

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \max(|\, x_{i1} - x_{j1} \,|, |\, x_{i2} - x_{j2} \,|, \ldots, |\, x_{ir} - x_{jr} \,|)$$

  - **Security alerts**

# More metrics

- Weighted squares
  - ➢A particular attribute may be a lot more important than other attributes
- Text: Cosine similarity

$$\cos(\vec{q},\vec{d}) = \frac{\overbrace{\vec{q}\bullet\vec{d}}^{\text{Dot product}}}{|\vec{q}||\vec{d}|} = \overbrace{\frac{\vec{q}}{|\vec{q}|}\bullet\frac{\vec{d}}{|\vec{d}|}}^{\text{Unit vectors}} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}\sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

| Doc | Team | Coach | Hockey | Baseball | Soccer | Penalty | Score | Win | loss |
|---|---|---|---|---|---|---|---|---|---|
| Doc1 | 5 | 0 | 3 | 0 | 2 | 0 | 0 | 2 | 0 |
| Doc2 | 3 | 0 | 2 | 0 | 1 | 1 | 0 | 1 | 0 |
| Doc3 | 0 | 7 | 0 | 2 | 1 | 0 | 0 | 3 | 0 |
| Doc4 | 0 | 1 | 0 | 0 | 1 | 2 | 2 | 0 | 3 |

# Distance functions for Binary & Nominal attributes

- We use a confusion matrix to introduce the distance functions/measures.

# Confusion matrix

| | | Data point $j$ | | |
|---|---|---|---|---|
| | | 1 | 0 | |
| Data point $i$ | 1 | $a$ | $b$ | $a+b$ |
| | 0 | $c$ | $d$ | $c+d$ |
| | | $a+c$ | $b+d$ | $a+b+c+d$ |

$$Hamming\ distance = \frac{\#of\ dissimilar\ attributes}{\#of\ dissimilar + \#of\ similar} = \frac{b+c}{b+c+a+d}$$

# Confusion matrix

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
|---|---|---|---|---|---|
| $R_1$ | 1 | 0 | 0 | 1 | 1 |
| $R_2$ | 0 | 0 | 0 | 1 | 0 |

What is the Manhattan Distance for $R_1$-$R_2$?

2

What is the distance normalized for # of attributes?

2/5

| | | $R_2$ | |
|---|---|---|---|
| | | 1 | 0 |
| $R_1$ | 1 | 1 (a) | 2 (b) |
| | 0 | 0 (c) | 2 (d) |

$$Distance = \frac{b + c}{a + b + c + d} = \frac{2}{5}$$

# Symmetric binary attributes

- A binary attribute is **symmetric** if both of its states (0 and 1) have equal importance, and carry the same weights, e.g., male and female of the attribute Gender

# Asymmetric binary attributes

- Asymmetric: if one of the states is more important or more valuable than the other.
  - By convention, state 1 represents the more important state, which is typically the rare or infrequent state.
  - Jaccard coefficient is a popular measure

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{b+c}{a+b+c}$$

| | | Data point $j$ | | |
|---|---|---|---|---|
| | | 1 | 0 | |
| Data point $i$ | 1 | $a$ | $b$ | $a+b$ |
| | 0 | $c$ | $d$ | $c+d$ |
| | | $a+c$ | $b+d$ | $a+b+c+d$ |

  - We can have some variations, adding weights

# Dissimilarity between Binary Variables

- Example

| Name | Gender | Fever | Cough | Test-1 | Test-2 | Test-3 | Test-4 |
|---|---|---|---|---|---|---|---|
| Jack | M | Y | N | P | N | N | N |
| Mary | F | Y | N | P | N | P | N |
| Jim | M | Y | P | N | N | N | N |

  - Gender is a symmetric attribute
  - The remaining attributes are asymmetric binary
  - Let the values Y (Yes) and P (Positive) be set to 1, and the value N (Negative) be set to 0

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

# Categorical variables with multiple levels

- A generalization of the binary variable in that it can take more than 2 states, e.g., red, yellow, blue, green
- Method 1: Simple matching
  - *m*: # of matches, *p*: total # of variables

$$d(i,j) = \frac{p-m}{p}$$

- Method 2: use a large number of binary variables
  - creating a new binary variable for each of the *M* nominal states

# Another distance metric used in supervised learning

Value difference measure (VDM):$d_{ij}$

$$\sum_{h=1}^{All\ classes} |P(h|val_i) - P(h|val_j)|$$

| ID | Age | Income | Family | CCAvg | Personal Loan |
|---|---|---|---|---|---|
| 1 | Young | Low | 4 | Low | 0 |
| 2 | Old | Low | 3 | Low | 0 |
| 3 | Middle | Low | 1 | Low | 0 |
| 4 | Middle | Medium | 1 | Low | 0 |
| 5 | Middle | Low | 4 | Low | 0 |
| 6 | Middle | Low | 4 | Low | 0 |
| 10 | Middle | High | 1 | High | 1 |
| 17 | Middle | Medium | 4 | Medium | 1 |
| 19 | Old | High | 2 | High | 1 |
| 30 | Middle | Medium | 1 | Medium | 1 |
| 39 | Old | Medium | 3 | Medium | 1 |
| 43 | Young | Medium | 4 | Low | 1 |
| 48 | Middle | High | 4 | Low | 1 |

Distance between F1 and F2

$$= |P(0|F1) - P(0|F2)| + |P(1|F1) - P(1|F2)|$$

$$= |0.5 - 0| + |0.5 – 1|$$

$$= 1$$

# Ordinal variables

- Same as numeric
- Look up is better than computation

# Look up matrix for ordinal with 3 states

$$
\begin{array}{cc}
 & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 0 & 1 & 4 \\ 1 & 0 & 1 \\ 4 & 1 & 0 \end{bmatrix}
\end{array}
$$

# Clustering

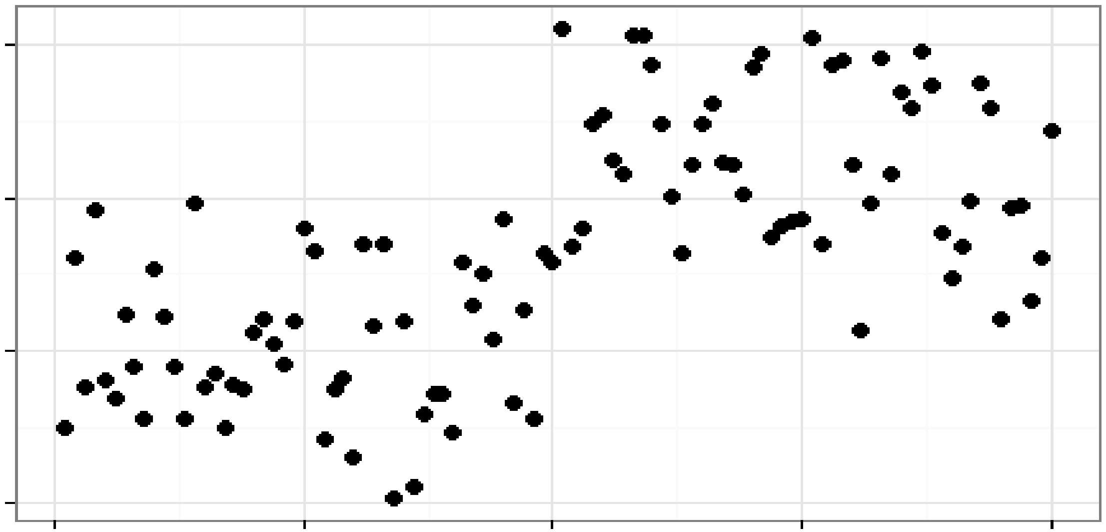
# Unsupervised learning

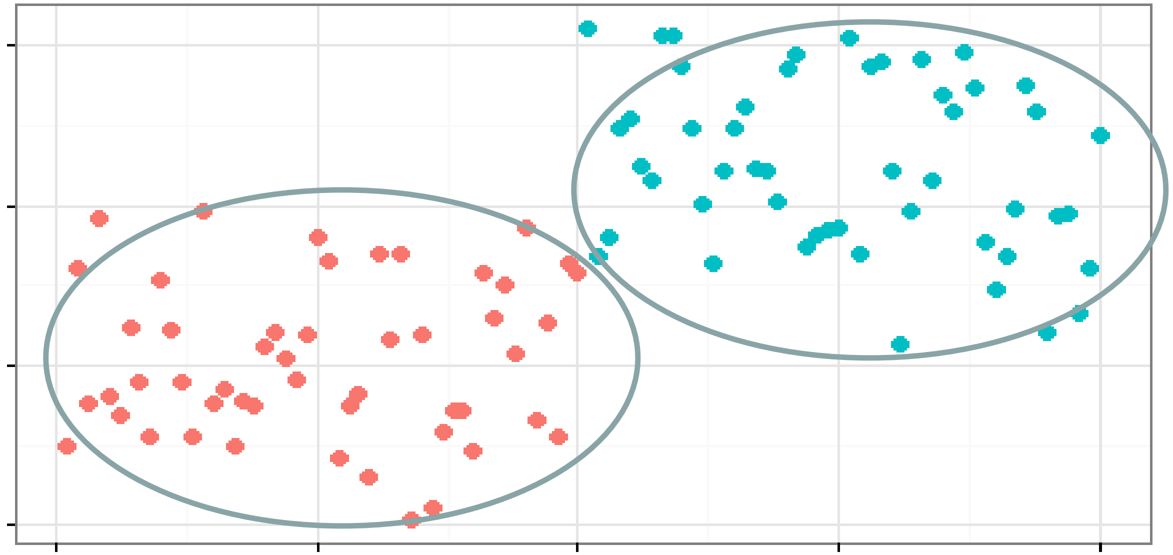
- Supervised:  Data and target
- Unsupervised:  Just data

# Clustering

- One of the unsupervised learning techniques

- Finding similarity groups in data, called **clusters**, i.e.,
  - Data instances that are similar to (near) each other are in the same cluster
  - Data instances that are very different (far away) from each other fall in different clusters.





# A few clustering applications

- In marketing, segment customers according to their similarities
  - To do targeted marketing
  - It is not uncommon to have over 100,000 segments in insurance clustering

# Google search

- Given a collection of text documents, organize them according to their content similarities
  - e.g., Google news

# Algorithms

- **Hierarchical** approach: Create a hierarchical decomposition of the set of data (or objects) using some criterion (Wald)

- **Partitioning** approach: Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors (K-means, Spectral clustering)

- **Model-based** methods: A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other (EM)

# HIERARCHICAL (AGGLOMERATIVE) CLUSTERING

# Agglomerative clustering (Hierarchical)

- Assign each item to its own cluster, so that if you have N items, you now have N clusters, each containing just one item.
- Merge most similar clusters into a single cluster, so that now you have one less cluster.
- Compute distances (similarities) between the new cluster and each of the old clusters.
- Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.

# Example of agglomerative clustering

| | BOS | NY | DC | MIA | CHI | SEA | SF | LA | DEN |
|---|---|---|---|---|---|---|---|---|---|
| BOS | 0 | 206 | 429 | 1504 | 963 | 2976 | 3095 | 2979 | 1949 |
| NY | 206 | 0 | 233 | 1308 | 802 | 2815 | 2934 | 2786 | 1771 |
| DC | 429 | 233 | 0 | 1075 | 671 | 2684 | 2799 | 2631 | 1616 |
| MIA | 1504 | 1308 | 1075 | 0 | 1329 | 3273 | 3053 | 2687 | 2037 |
| CHI | 963 | 802 | 671 | 1329 | 0 | 2013 | 2142 | 2054 | 996 |
| SEA | 2976 | 2815 | 2684 | 3273 | 2013 | 0 | 808 | 1131 | 1307 |
| SF | 3095 | 2934 | 2799 | 3053 | 2142 | 808 | 0 | 379 | 1235 |
| LA | 2979 | 2786 | 2631 | 2687 | 2054 | 1131 | 379 | 0 | 1059 |
| DEN | 1949 | 1771 | 1616 | 2037 | 996 | 1307 | 1235 | 1059 | 0 |

- No assignment of centroid upfront.
- Each point is considered a cluster.
- Find the closest clusters and merge them.

| | BOS/NY | DC | MIA | CHI | SEA | SF | LA | DEN |
|---|---|---|---|---|---|---|---|---|
| BOS/NY | 0 | 223 | 1308 | 802 | 2815 | 2934 | 2786 | 1771 |
| DC | 223 | 0 | 1075 | 671 | 2684 | 2799 | 2631 | 1616 |
| MIA | 1308 | 1075 | 0 | 1329 | 3273 | 3053 | 2687 | 2037 |
| CHI | 802 | 671 | 1329 | 0 | 2013 | 2142 | 2054 | 996 |
| SEA | 2815 | 2684 | 3273 | 2013 | 0 | 808 | 1131 | 1307 |
| SF | 2934 | 2799 | 3053 | 2142 | 808 | 0 | 379 | 1235 |
| LA | 2786 | 2631 | 2687 | 2054 | 1131 | 379 | 0 | 1059 |
| DEN | 1771 | 1616 | 2037 | 996 | 1307 | 1235 | 1059 | 0 |

| | BOS/NY/DC | MIA | CHI | SEA | SF | LA | DEN |
|---|---|---|---|---|---|---|---|
| BOS/NY/DC | 0 | 1075 | 671 | 2684 | 2799 | 2631 | 1616 |
| MIA | 1075 | 0 | 1329 | 3273 | 3053 | 2687 | 2037 |
| CHI | 671 | 1329 | 0 | 2013 | 2142 | 2054 | 996 |
| SEA | 2684 | 3273 | 2013 | 0 | 808 | 1131 | 1307 |
| SF | 2799 | 3053 | 2142 | 808 | 0 | 379 | 1235 |
| LA | 2631 | 2687 | 2054 | 1131 | 379 | 0 | 1059 |
| DEN | 1616 | 2037 | 996 | 1307 | 1235 | 1059 | 0 |

| | BOS/ NY/DC | MIA | CHI | SEA | SF/LA | DEN |
|---|---|---|---|---|---|---|
| BOS/NY/DC | 0 | 1075 | 671 | 2684 | 2631 | 1616 |
| MIA | 1075 | 0 | 1329 | 3273 | 2687 | 2037 |
| CHI | 671 | 1329 | 0 | 2013 | 2054 | 996 |
| SEA | 2684 | 3273 | 2013 | 0 | 808 | 1307 |
| SF/LA | 2631 | 2687 | 2054 | 808 | 0 | 1059 |
| DEN | 1616 | 2037 | 996 | 1307 | 1059 | 0 |

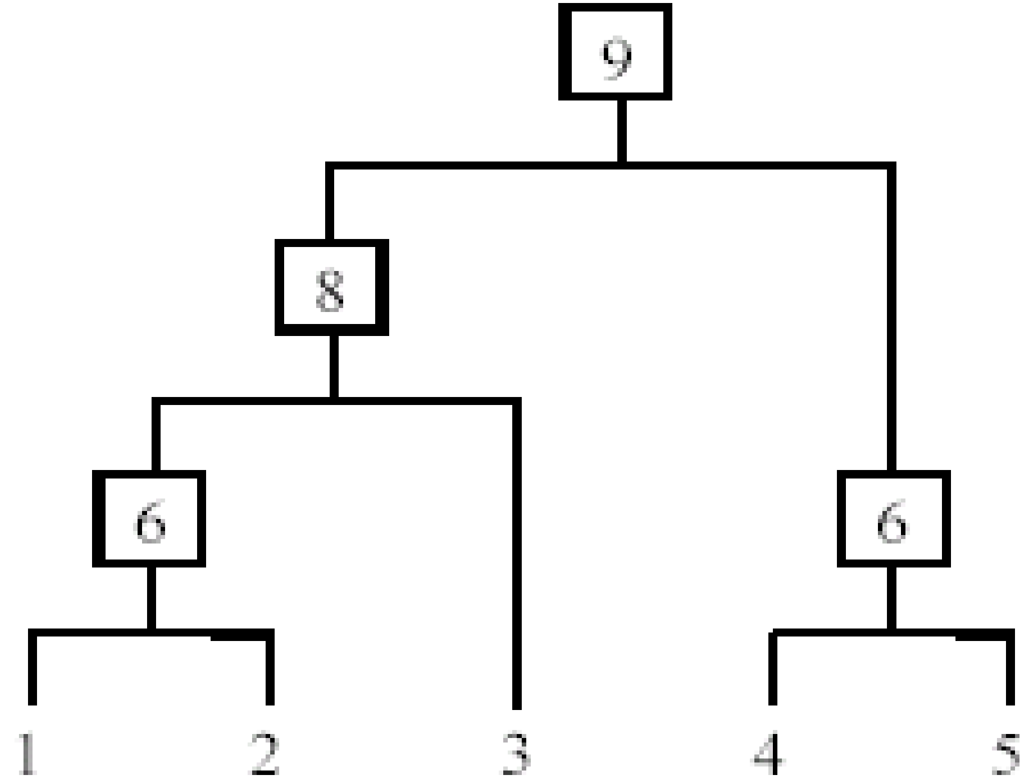
| | BOS/NY/DC/<br>CHI | MIA | SEA | SF/LA | DEN |
|---|---|---|---|---|---|
| BOS/NY/DC/CHI | 0 | 1075 | 2013 | 2054 | 996 |
| MIA | 1075 | 0 | 3273 | 2687 | 2037 |
| SEA | 2013 | 3273 | 0 | 808 | 1307 |
| SF/LA | 2054 | 2687 | 808 | 0 | 1059 |
| DEN | 996 | 2037 | 1307 | 1059 | 0 |

| | BOS/NY/DC/CHI | MIA | SF/LA/SEA | DEN |
|---|---|---|---|---|
| BOS/NY/DC/CHI | 0 | 1075 | 2013 | 996 |
| MIA | 1075 | 0 | 2687 | 2037 |
| SF/LA/SEA | 2054 | 2687 | 0 | 1059 |
| DEN | 996 | 2037 | 1059 | 0 |

| | BOS/NY /DC/CHI/DEN | MIA | SF/LA/SEA |
|---|---|---|---|
| BOS/NY/DC/CHI/DEN | 0 | 1075 | 1059 |
| MIA | 1075 | 0 | 2687 |
| SF/LA/SEA | 1059 | 2687 | 0 |

| | BOS/NY /DC/CHI /DEN/SF /LA/SEA | MIA |
|---|---|---|
| BOS/NY/DC/CHI/DEN/SF/LA/SEA | 0 | 1075 |
| MIA | 1075 | 0 |

# Hierarchical Clustering




**Decomposes data objects into a several levels of nested partitioning (<u>tree</u> of clusters).**

**A <u>clustering</u> of the data objects is obtained by <u>cutting</u> the dendrogram at the desired level, then each <u>connected component</u> forms a cluster.**

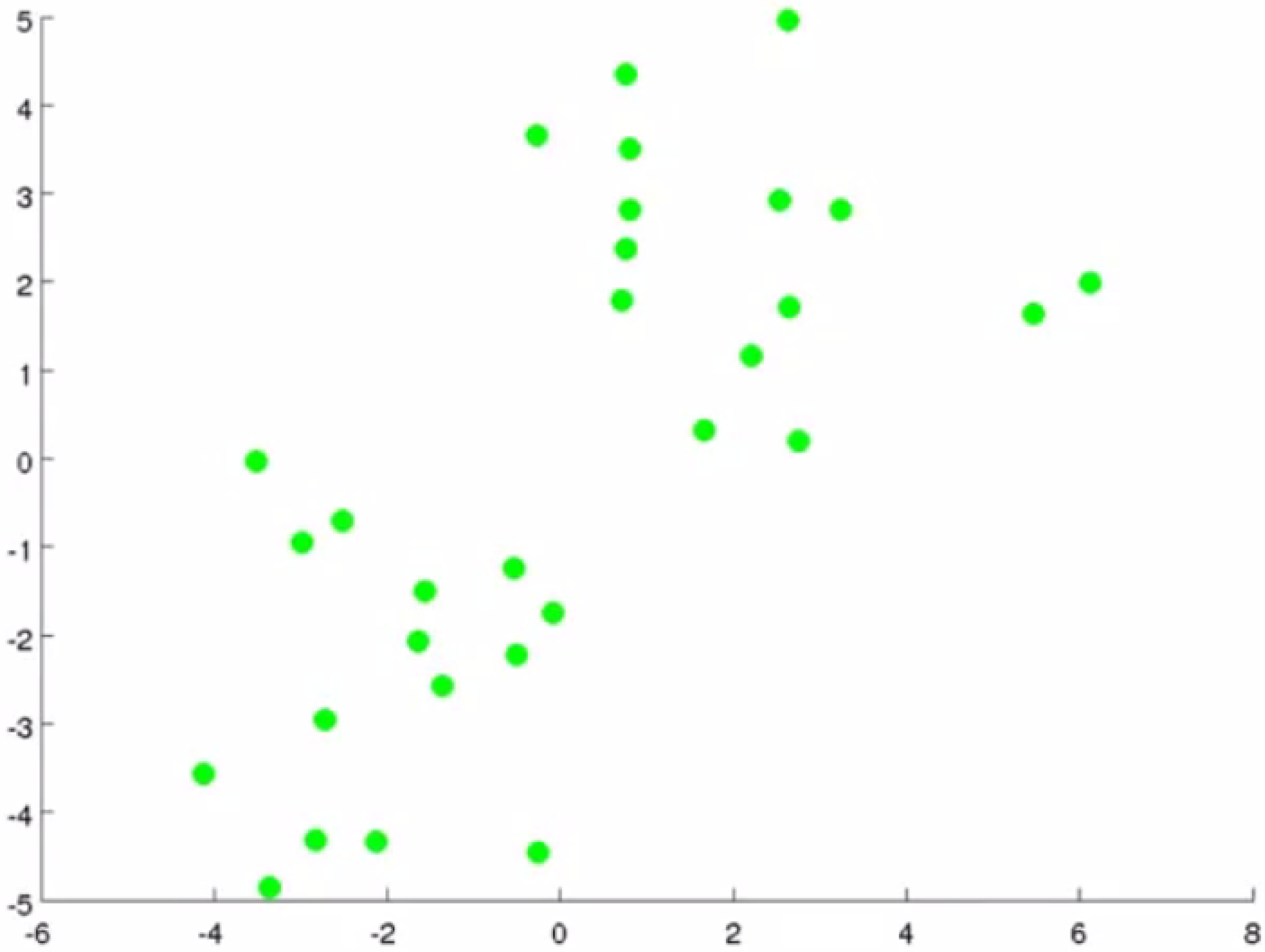
Partitioning algorithms

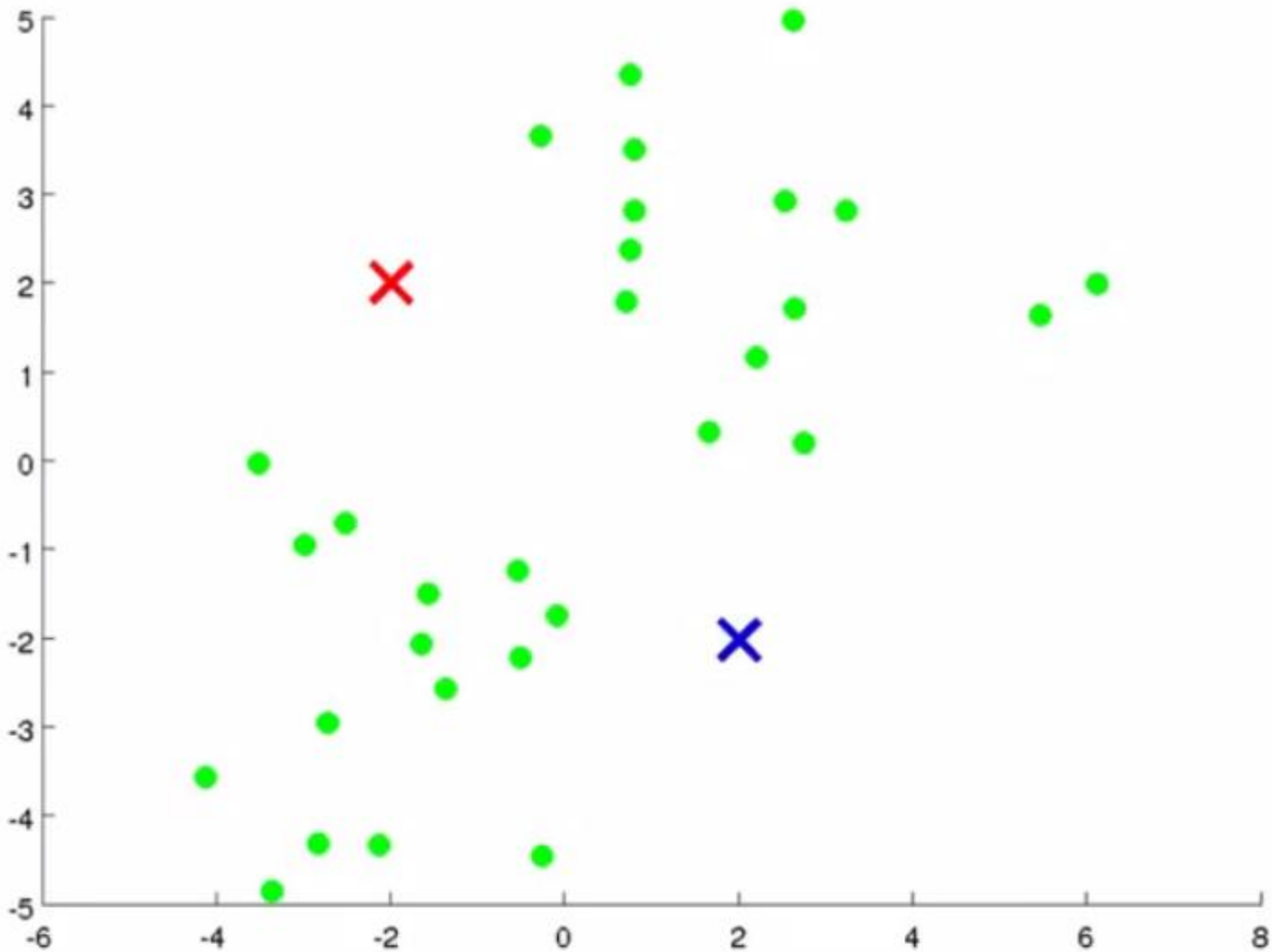
# K-MEANS AND K-MEDOIDS

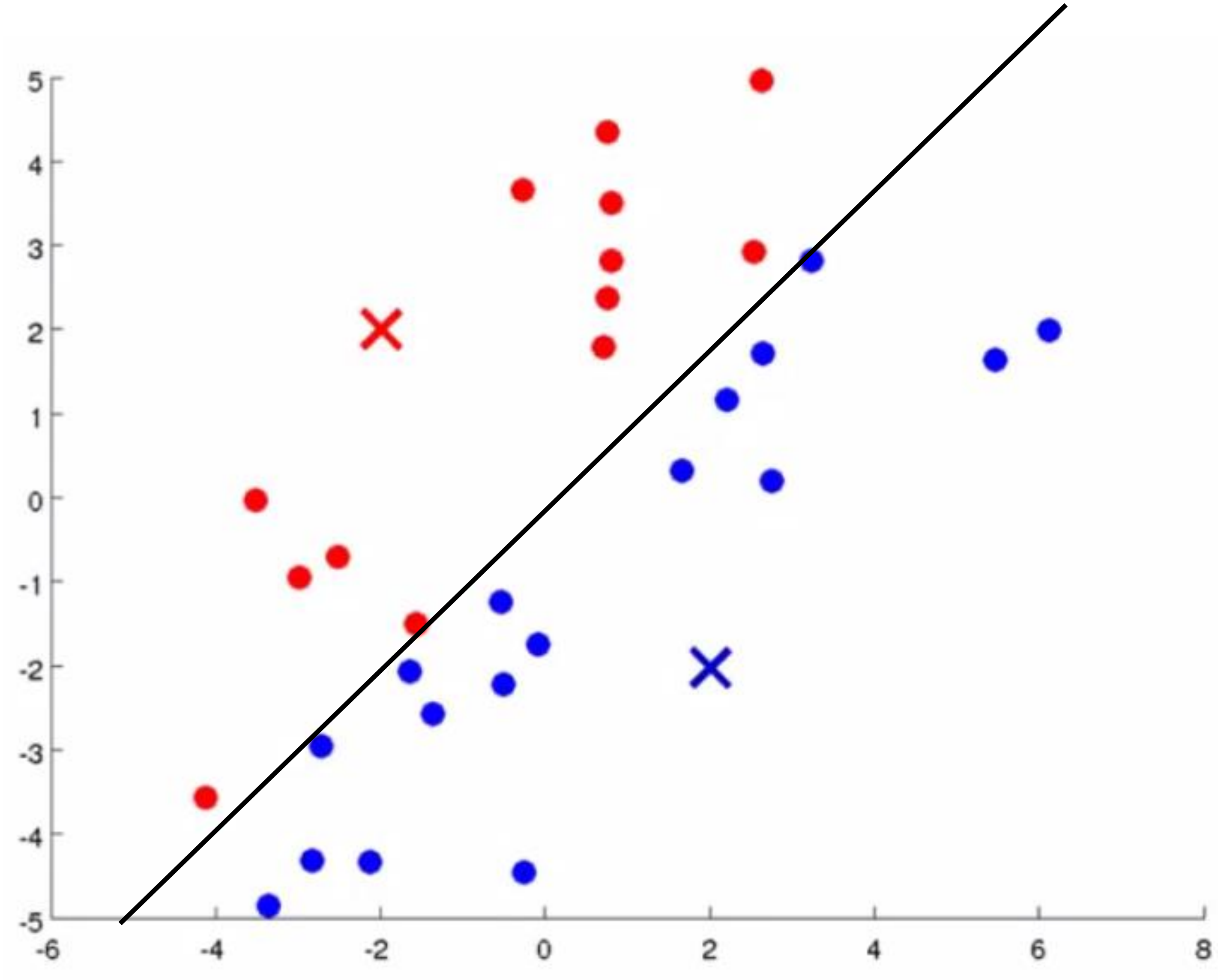
# K-means clustering

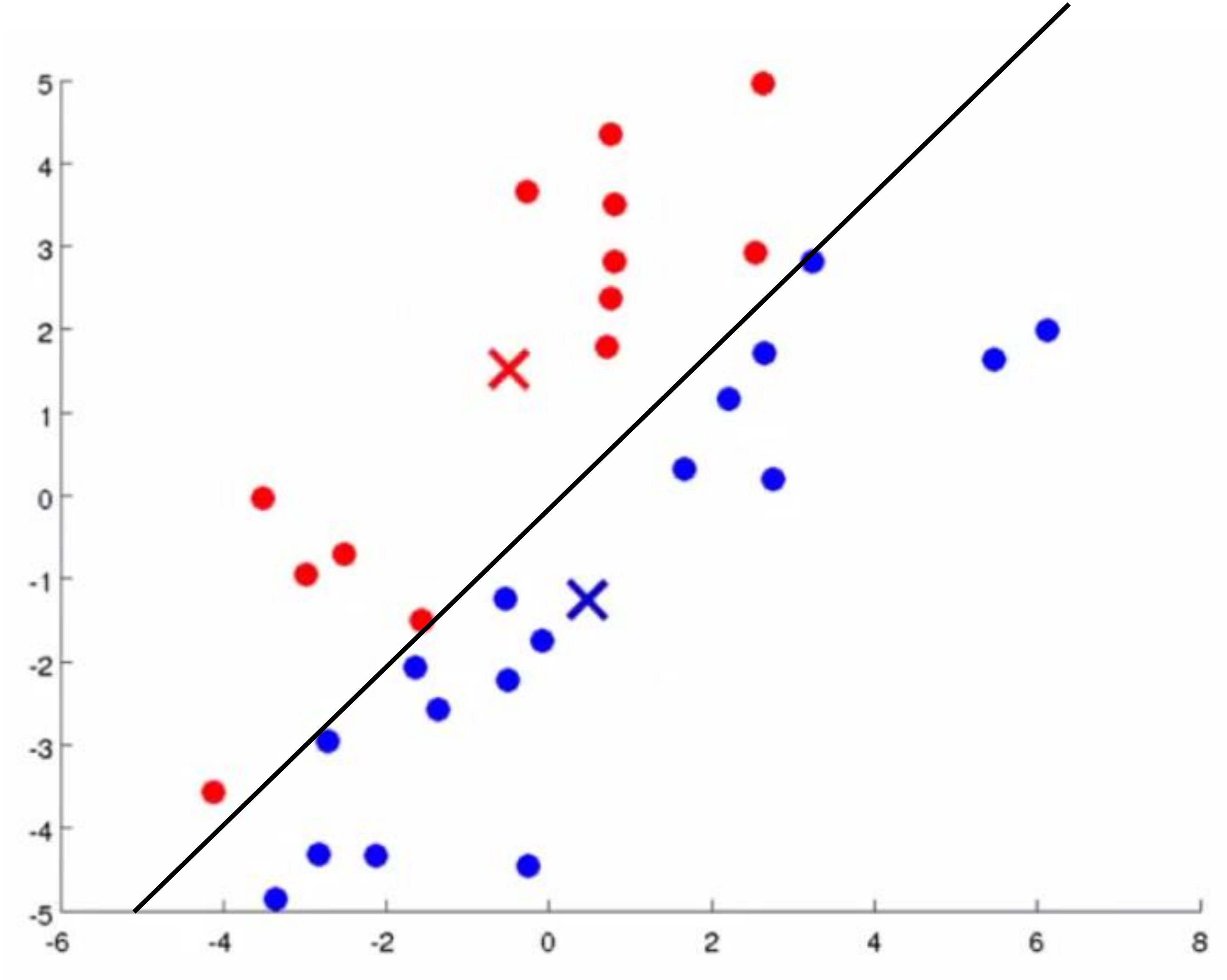
- K-means is a partitional clustering algorithm as it partitions the given data into *k* clusters.
  – Each cluster has a cluster **center**, called **centroid**.
  – *k* is specified by the user

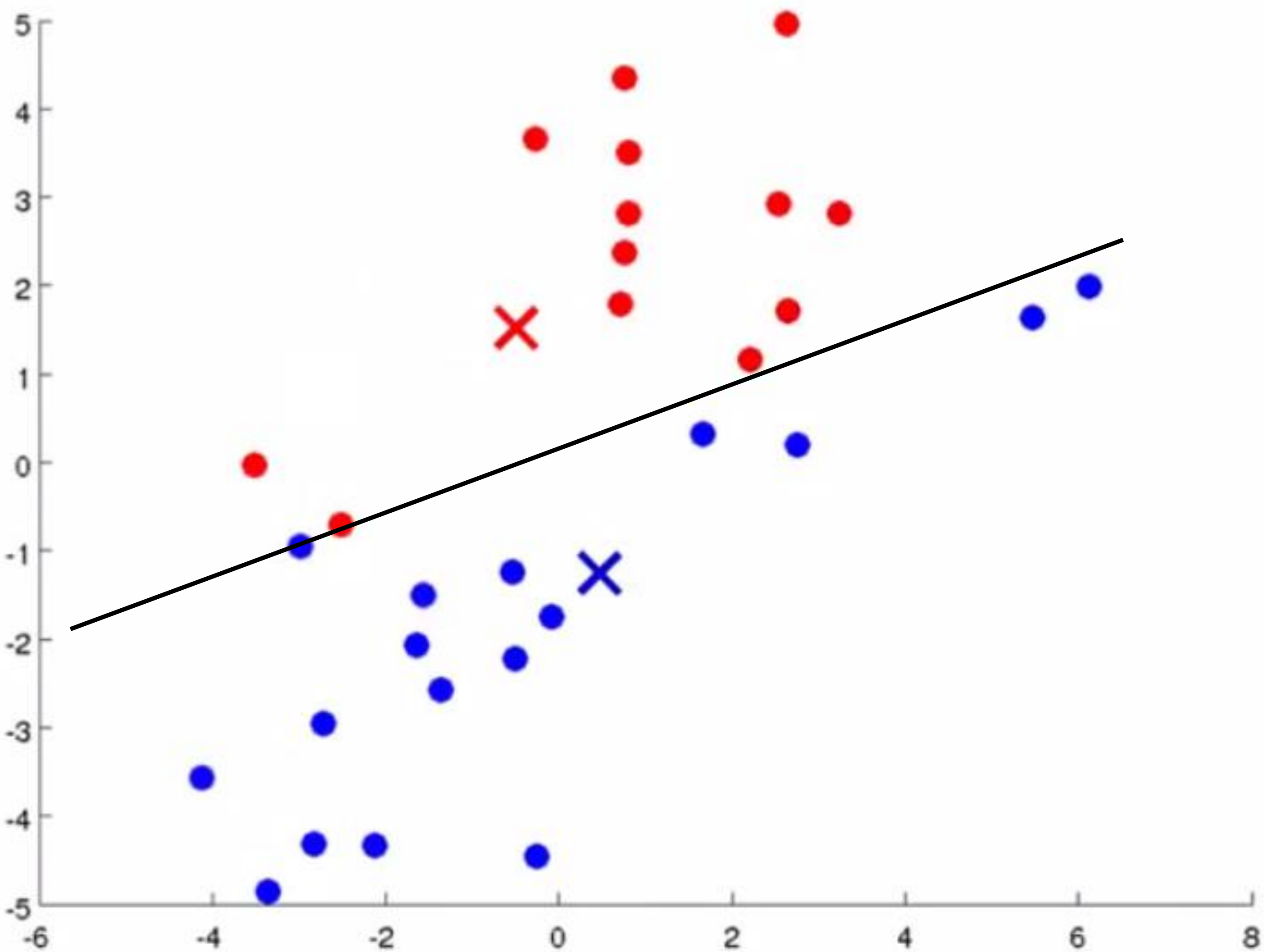
Pictures courtesy Andrew Ngs course on Coursera

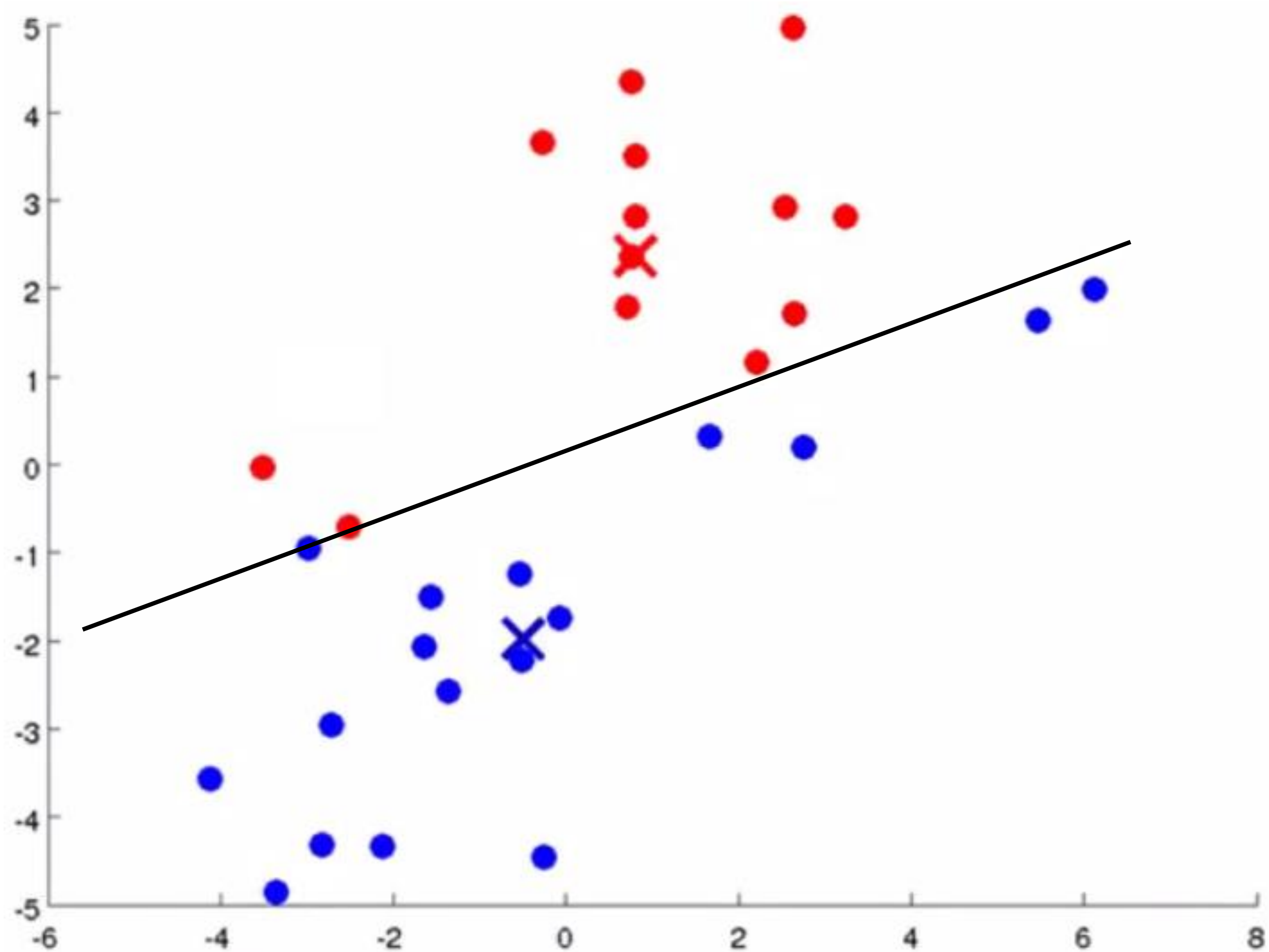


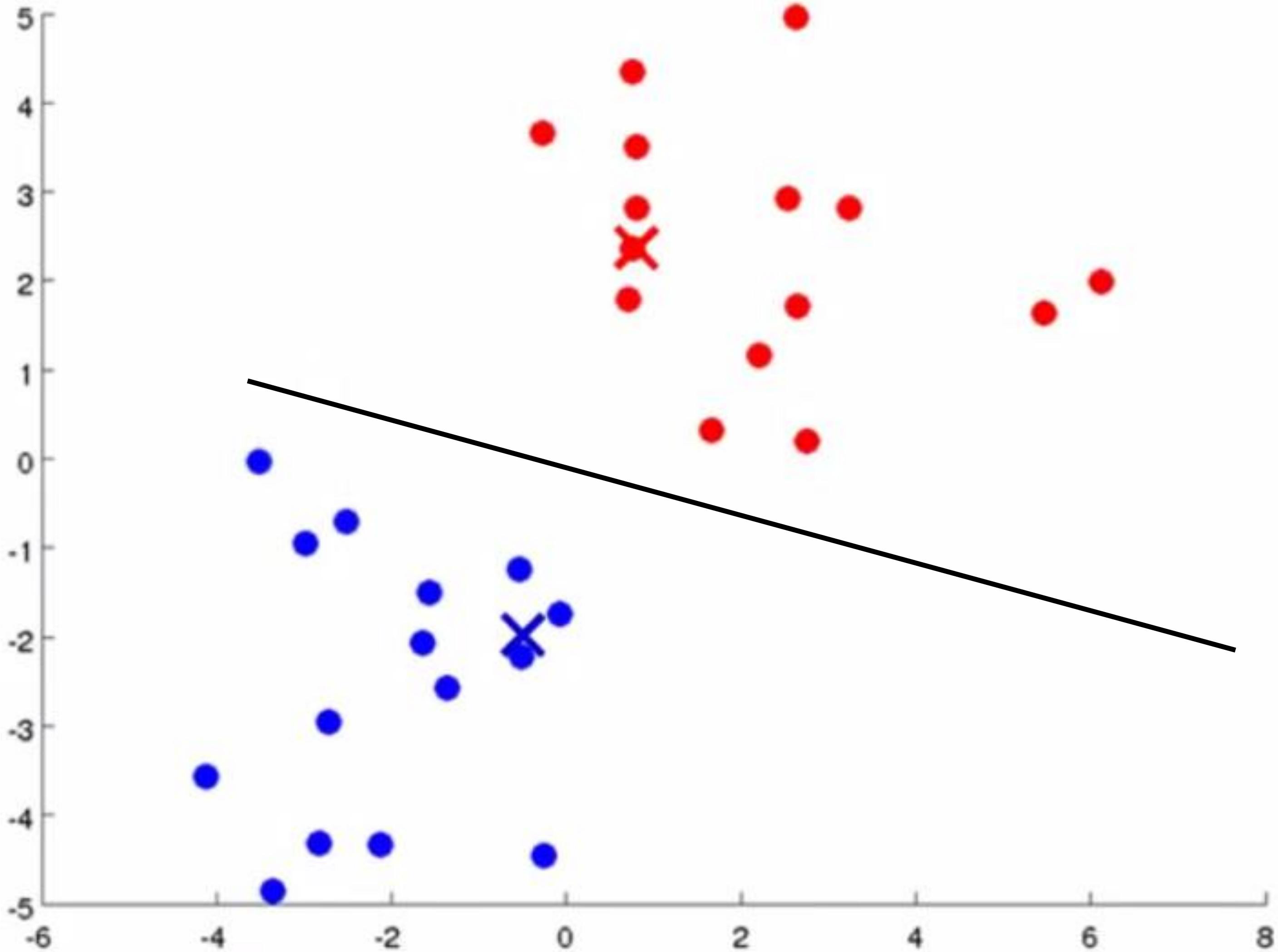

5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8

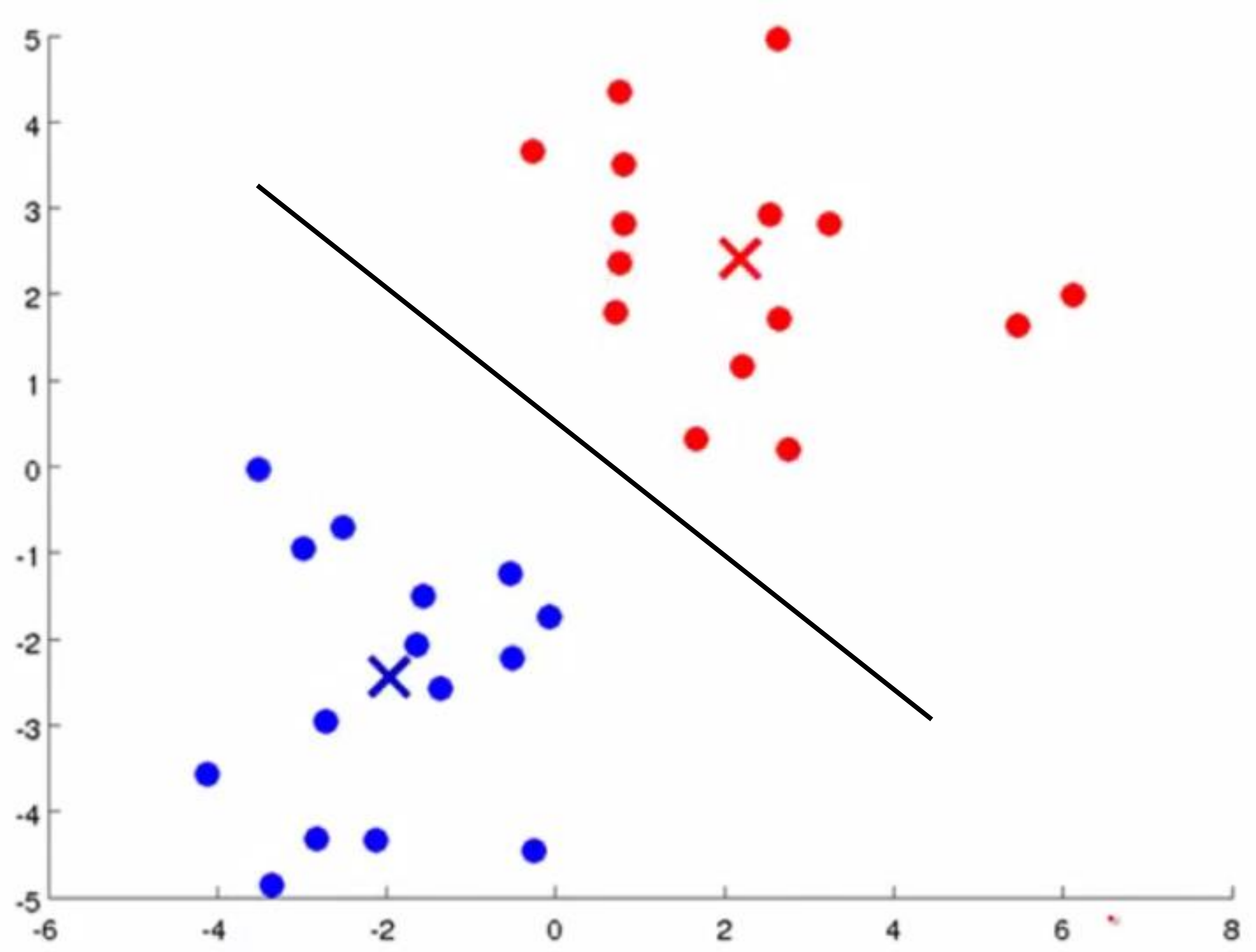

5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8


5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8


5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8


5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8


5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8


5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8

# 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, or **if some clusters don't get any points** go to 2.

# Optimizing

$$\frac{1}{m}\sum_{i=1}^{m}||x^{(i)} - \mu_{c^{(i)}}||^2$$

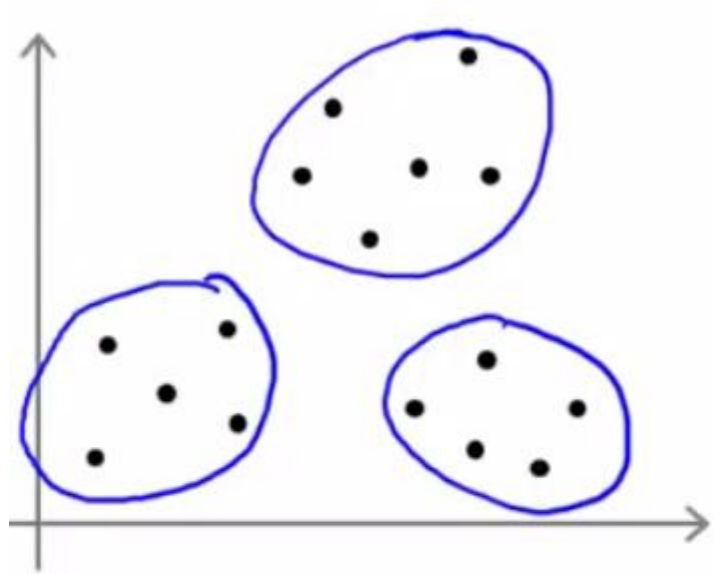
# Stopping/convergence criterion

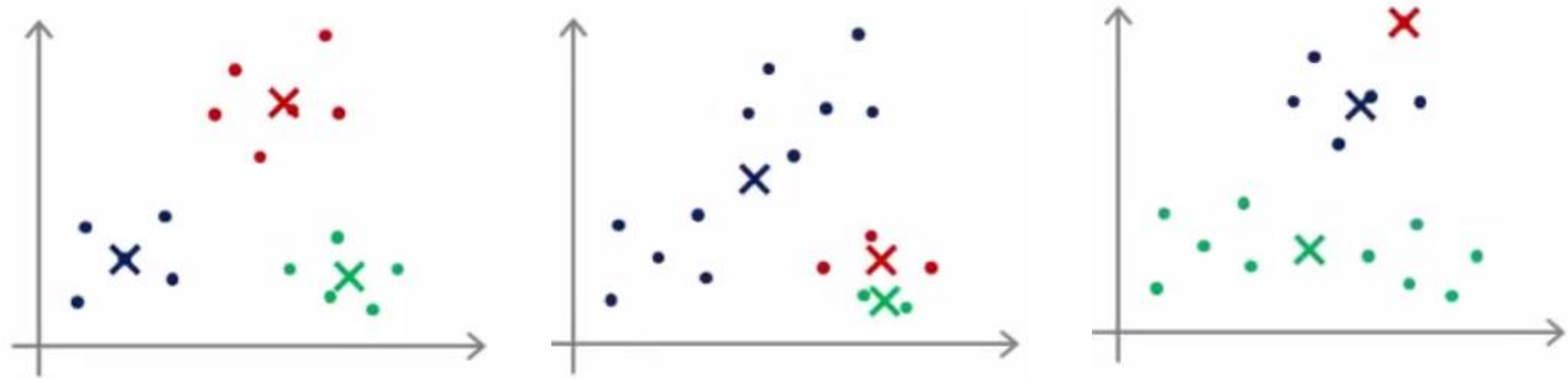
1. No (or minimum) re-assignments of data points to different clusters,
2. No (or minimum) change of centroids, or
3. Minimum decrease in the **sum of squared error** (SSE),

$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2 \qquad (1)$$

- $C_i$ is the $j$th cluster, $\mathbf{m}_j$ is the centroid of cluster $C_j$ (the mean vector of all the data points in $C_j$

# Local optima





# What Is the Problem with K-Means?

- The k-means algorithm is sensitive to outliers !

- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster.

# What Is the Problem with Medoids?

- More robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean
- Works efficiently for small data sets but does not **scale well** for large data sets.
  - $O(k(n-k)^2)$ for each iteration

    where n is # of data,k is # of clusters

# HOW DO WE EMPLOY DISTANCE IN A CLUSTER?

# R CODE DEMO

# K-means versus Hierarchical

- Flat clustering produces a single partitioning
- Flat clustering needs the number of clusters to be specified
- Flat clustering is usually more efficient run-time wise

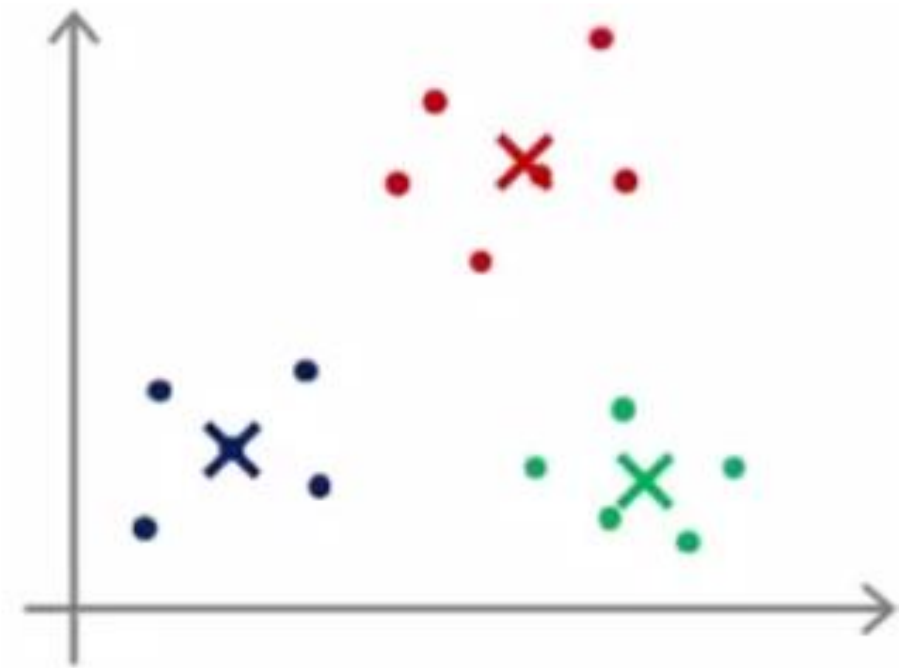
- Hierarchical Clustering can give different partitionings depending on the level-of-resolution we are looking at
- Hierarchical clustering doesn't need the number of clusters to be specified
- Hierarchical clustering can be slow (has to make several merge/Split decisions)

# ENGINEERING

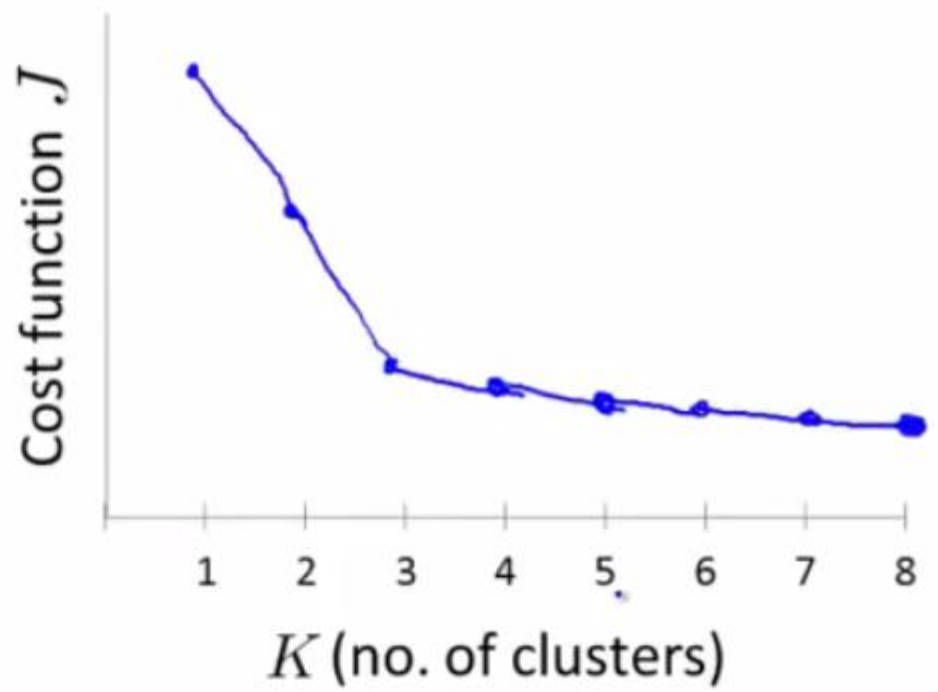
# Stability Check of the Clusters

- To check the stability of the clusters take a random sample of 95% of records. Compute the clusters. If the clusters formed are very similar to the original, then the clusters are fine.

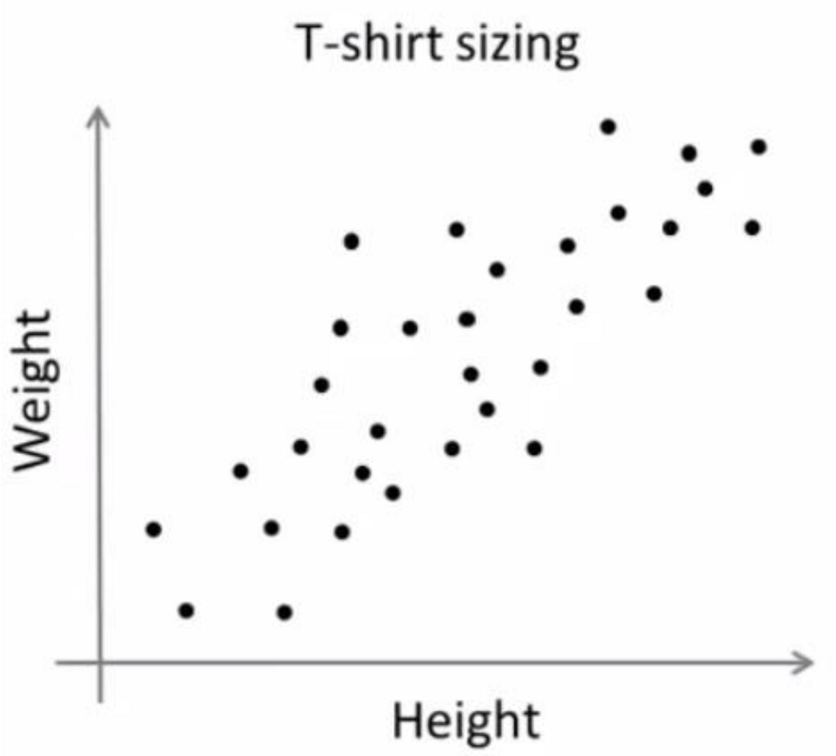
# Linearly clustered data

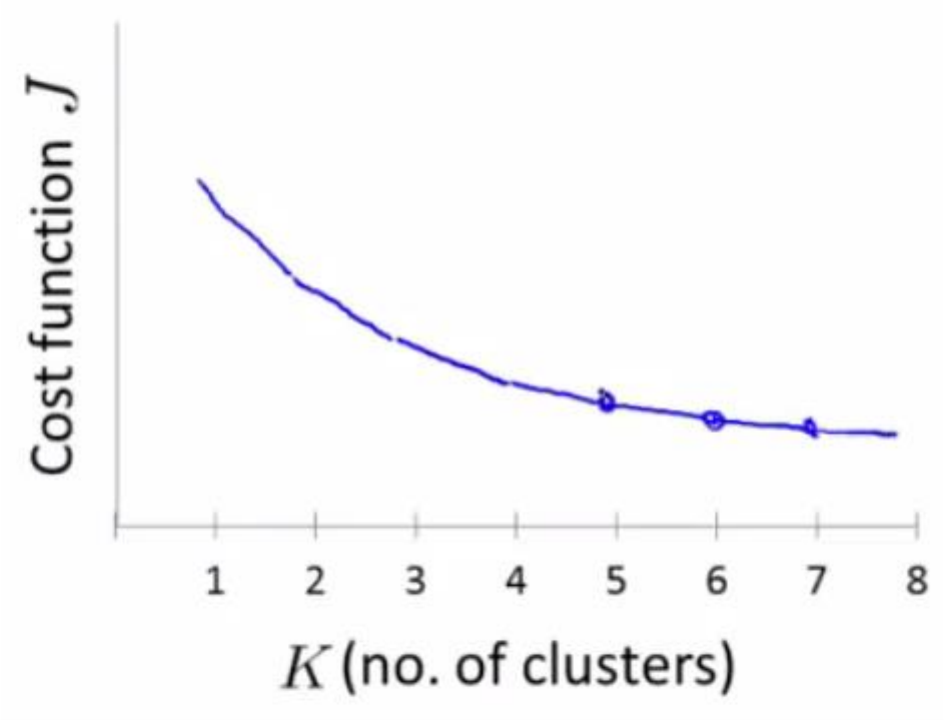


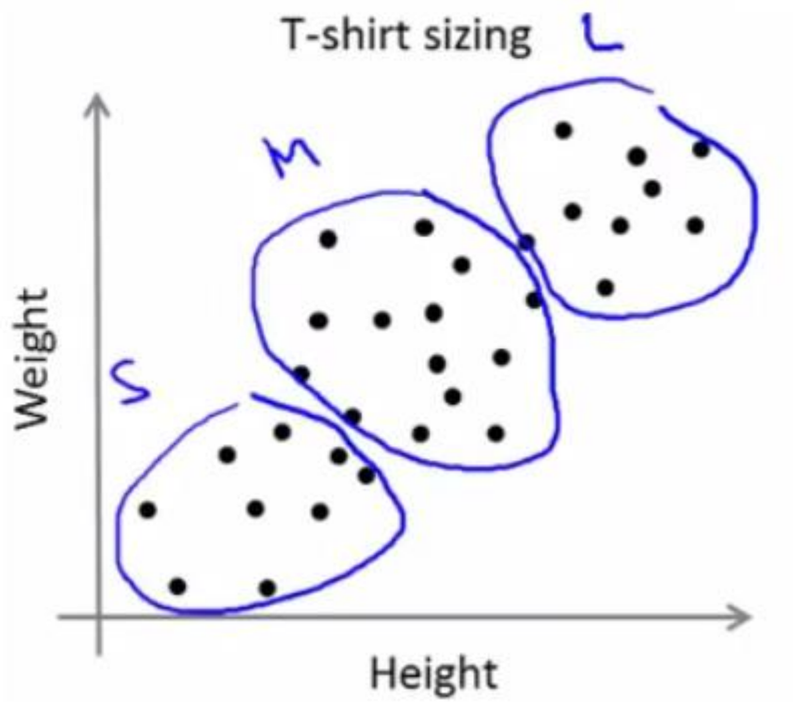
Nice



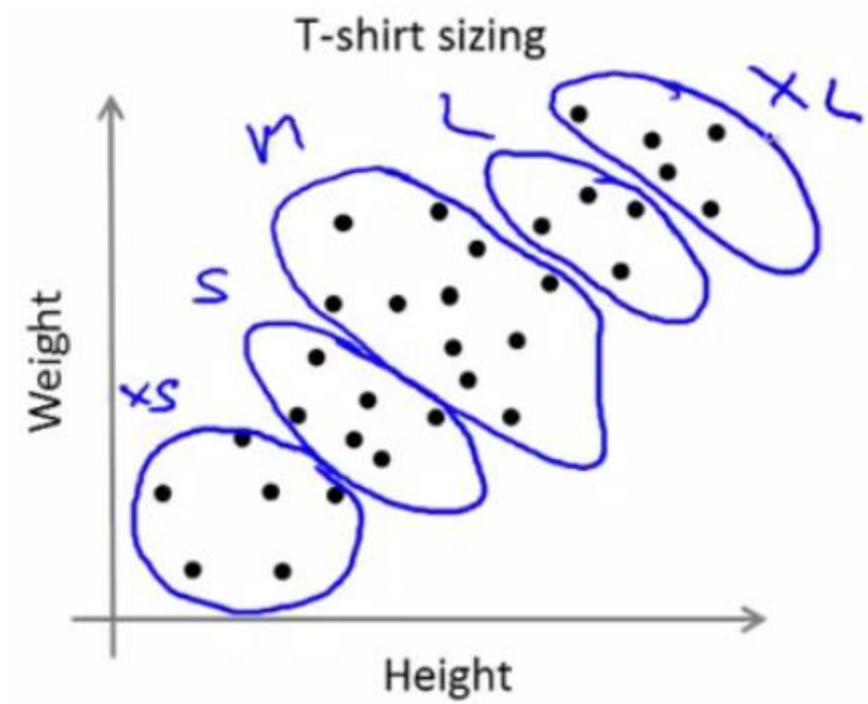
# Linearly separable but merged





T-shirt sizing
L
M
S
Weight
Height

T-shirt sizing
XL
L
M
S
XS
Weight
Height

# Linearly separable

- Run 50-500 simulations for small k (2-10). For large k (100 or so), we can do 1-5 simulations
- Pick the one that gives the best S

# Clustering Process Summary

- Choose an appropriate distance metric and calculate
- Decide *k* either based on the elbow or business user's intuition when no elbow found
- Kernel (higher dimensions), if required
- Cluster (k-means, etc.)
- Check stability of clusters using 90% or 95% data
- Define a cluster with properties (mean, median, etc.)

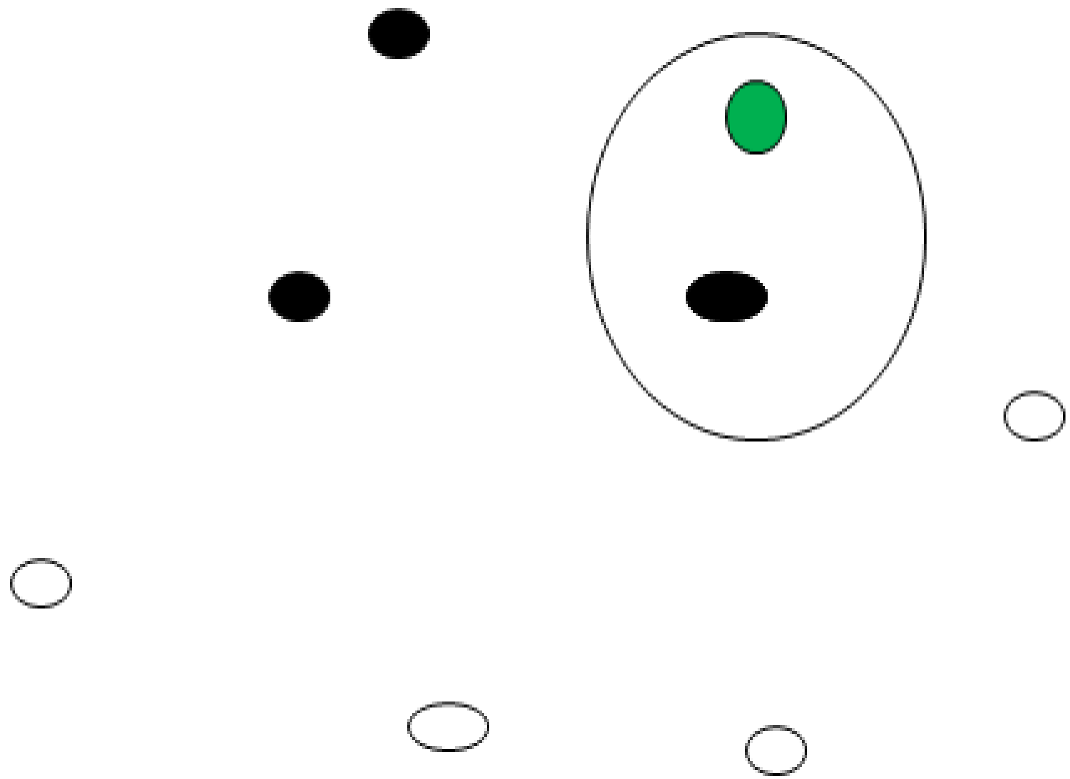
# Instance Based Learning

# Lazy Learning

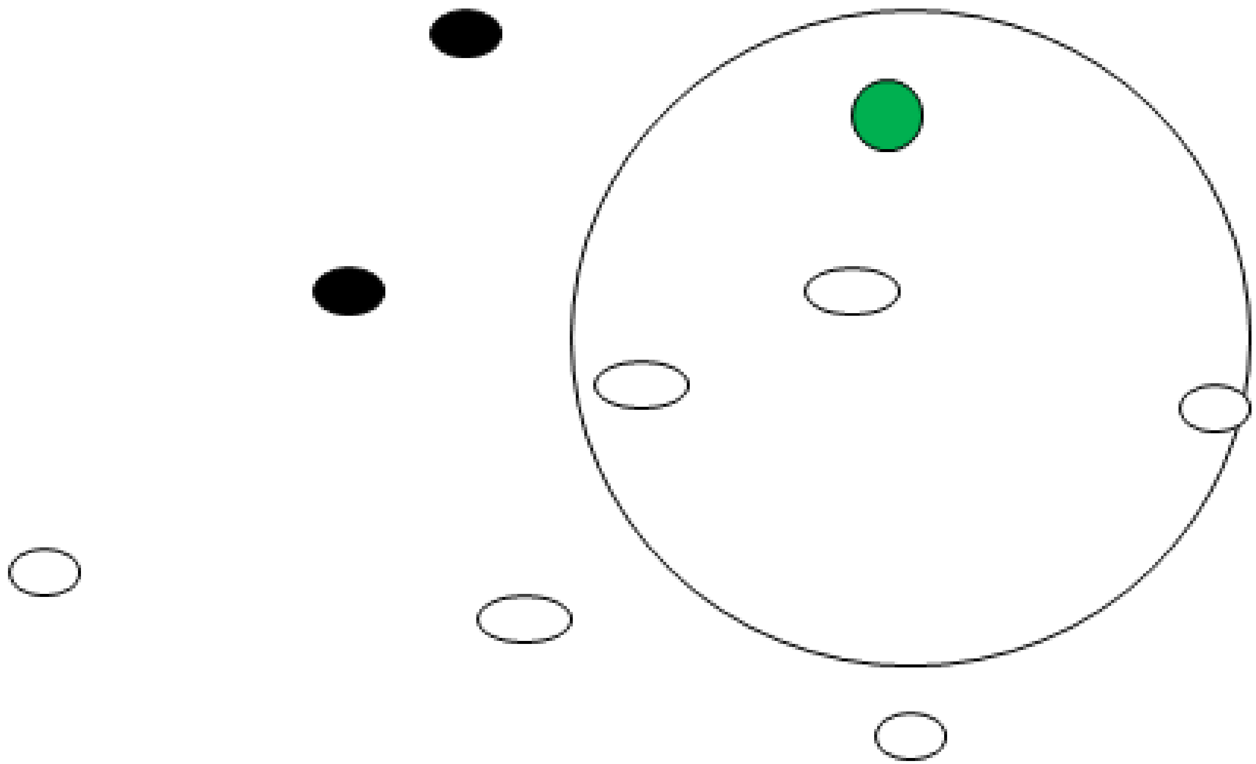
- Eager Learning
  - Explicit description of target function on the whole training set
- Instance-based / Lazy Learning
  - Learning = Storing all training instances
  - Classification = Assigning target function to a new instance

# KNN

# K=1



# K=3



# Process is simple

- Pick a number of neighbors you want to use for classification or regression (K)
- Choose a method to measure distances (same consideration as clustering)
- Keep a data set with records

# Process is simple

- For every new point, identify the number of nearest neighbors you picked using the method you chose

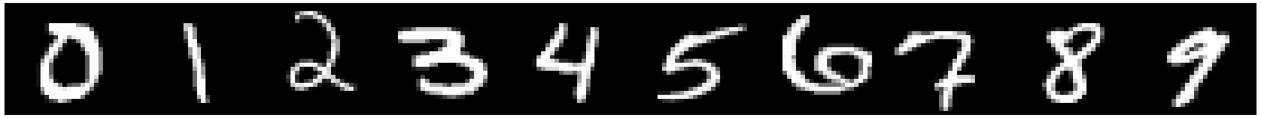
- Let them vote if it is a classification or take a mean/median for regression!

# K-NN is

- Supervised
- Non parametric
- Lazy
- Local heuristic

# kNN Example: Digit Recognition

UNITED STATES POSTAL SERVICE ®



- Digit Recognition
  - Handwritten digits
  - 28x28 pixel images: $d$ = 784
  - 60,000 training samples
  - 10,000 test samples

- Nearest neighbour is competitive

| | Test Error Rate (%) |
|---|---|
| Linear classifier (1-layer NN) | 12.0 |
| K-nearest-neighbors, Euclidean | 5.0 |
| K-nearest-neighbors, Euclidean, deskewed | 2.4 |
| K-NN, Tangent Distance, 16x16 | 1.1 |
| K-NN, shape context matching | 0.67 |
| 1000 RBF + linear classifier | 3.6 |
| SVM deg 4 polynomial | 1.1 |
| 2-layer NN, 300 hidden units | 4.7 |
| 2-layer NN, 300 HU, [deskewing] | 1.6 |
| LeNet-5, [distortions] | 0.8 |
| Boosted LeNet-4, [distortions] | 0.7 |

# K-NN

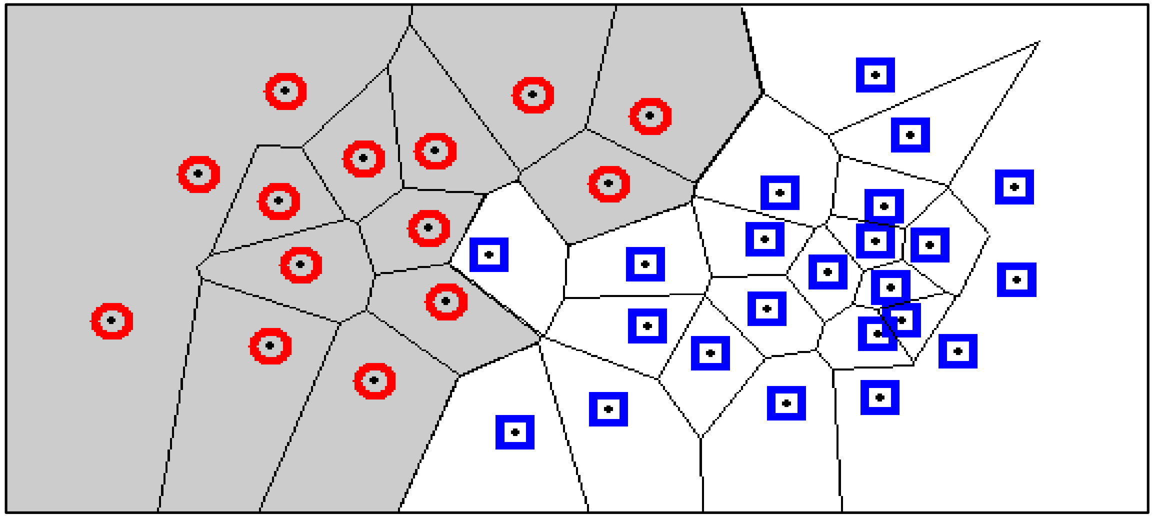
- Comes with a theoretical guarantee

- It is a Gibbs classifier.  The accuracy will be bounded by 2* Bayes optimal classifier

# Advantages

- If lazy
  - Simple
  - You can draw a very complex decision surface
    - Voronoi diagrams

# Decision Regions



- **A Voronoi diagram**
- Each cell contains one sample, and every location within the cell is closer to that sample than to any other sample.

- Every query point will be assigned the classification of the sample within that cell. The *decision boundary* separates the class regions based on the 1-NN decision rule.
- Knowledge of this boundary is sufficient to classify new points.

# Issues with KNN and instance based techniques

- Curse of dimensionality

- Requires more memory and more time

Attributes

Records

Search process

# ENGINEERING K-NN

# Attributes

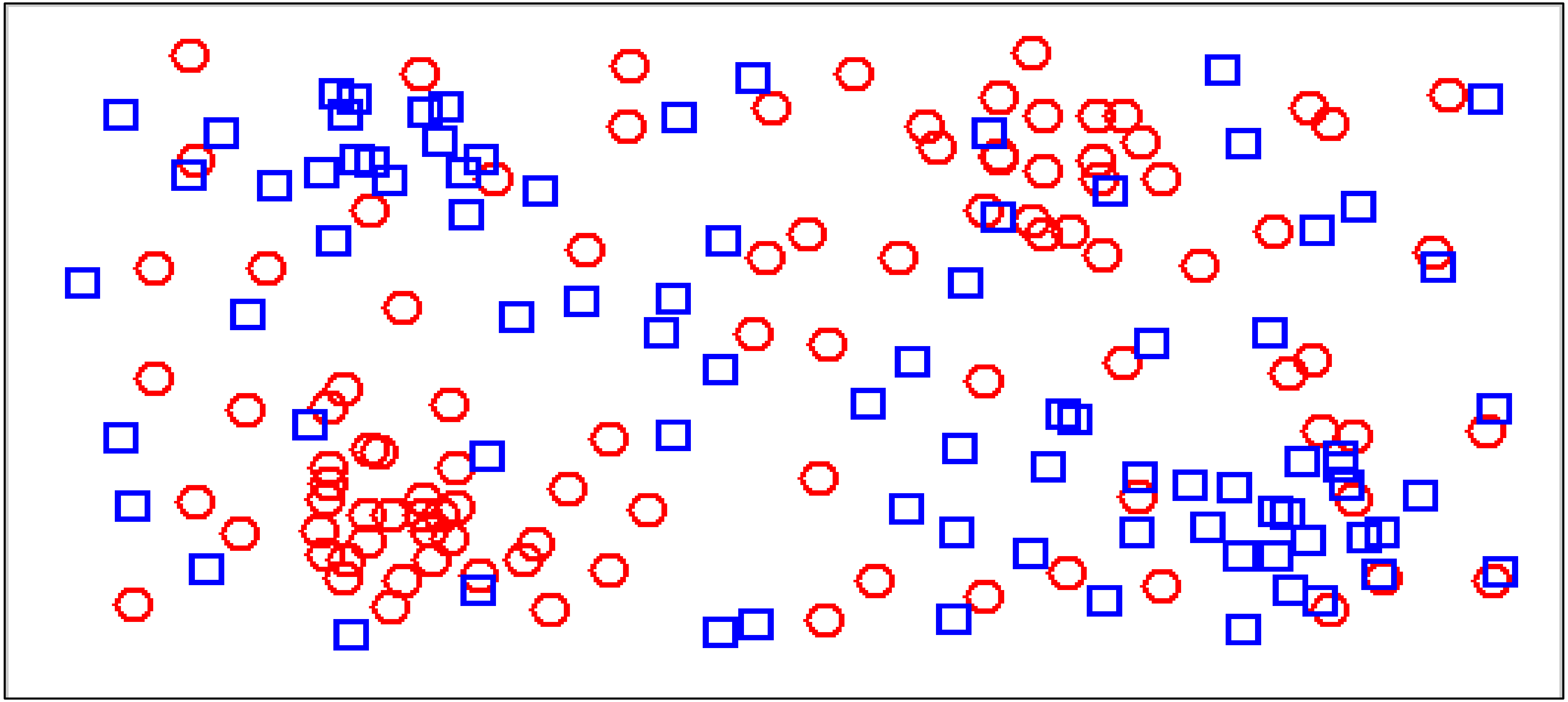
- Scaling the attributes is important
  - Attributes with larger range can dominate
  - Categorical variables and Ordinal variables need to be converted to numeric

# Curse of dimensionality

- K-NN is heavily impacted as all points are at the surface and hence similar
- Reduce the dimensions
  - Correlation
  - Info gain (filter approach: We lose some that are important)
  - Wrapper methods
    - Forward selection, Backward elimination
  - Weighting attributes

# Records: Outliers and overfitting

- Remove outliers

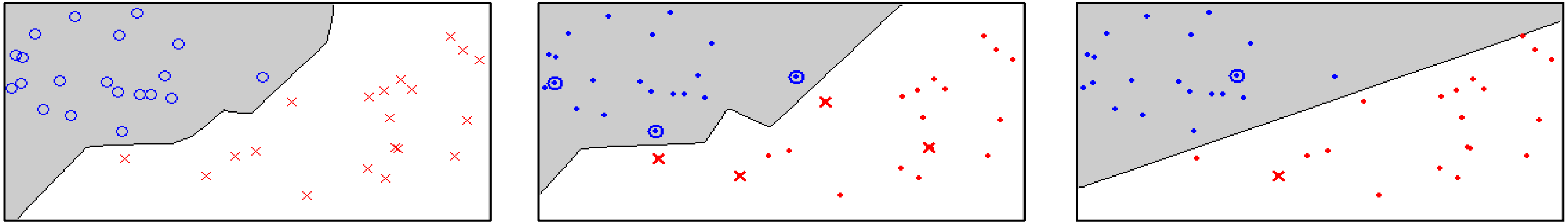


# Records: Handling missing values

- K-NN is impacted heavily by missing values
- Imputation is one option but might be self defeating

# Speeding up search

Delaunay triangulation

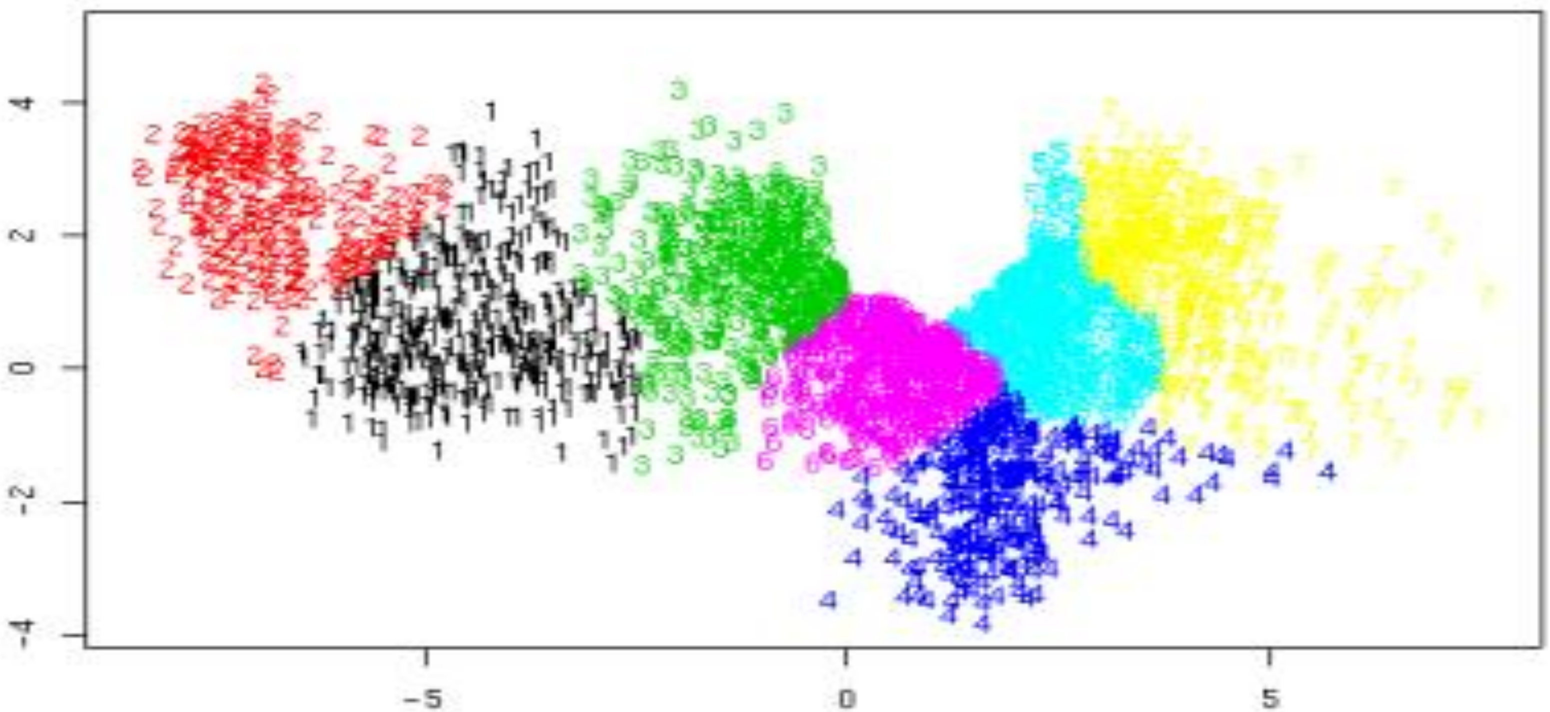


**Original data** **Condensed data** **Minimum Consistent Set**

Cran library: Class

# Speeding up

- Clustering



# COLLABORATIVE FILTERING

# Collaborative filtering

- How do I recommend?
  - Association rules
  - Similarity based (collaborative filtering)
  - Model based

# Collaborative filtering: primitive

Primitive version:

$$\hat{R}_{ik} = \alpha \sum_{X_j \in \mathbf{N}_i} W_{ij} R_{jk}$$

$$\alpha = \left(\sum |W_{ij}|\right)^{-1}$$

Similarity (Pearson coefficient):

$$W_{ij} = \frac{\sum_k (R_{ik} - \overline{R}_i)(R_{jk} - \overline{R}_j)}{\sqrt{\sum_k (R_{ik} - \overline{R}_i)^2 (R_{jk} - \overline{R}_j)^2}}$$

# Collaborative filtering: More refined

$$\hat{R}_{ik} = \overline{R}_i + \alpha \sum_{X_j \in \mathbf{N}_i} W_{ij}(R_{jk} - \overline{R}_j)$$

# Collaborative filtering

| | Matrix | Star Wars | Dark knight | Rocky | Sita Aur Gita | Star Trek | Cliffhanger | A.I. | MI | X-Men |
|---|---|---|---|---|---|---|---|---|---|---|
| Jim | 1 | 3 | 1 | 5 | 2 | 1 | | | 1 | |
| Sean | 2 | | 3 | 2 | | 4 | | 5 | | 3 |
| John | | 3 | | 4 | | 5 | | | 3 | 4 |
| Sidd | 4 | | | | 3 | | 4 | | 2 | |
| Penny | 5 | | 2 | | 2 | | 5 | | 1 | |
| Pete | | 5 | | | ? | | 4 | | | 4 |

# Collaborative filtering

| | Matrix | Star Wars | Dark knight | Rocky | Sita Aur Gita | Star Trek | Cliffhanger | A.I. | MI | X-Men |
|---|---|---|---|---|---|---|---|---|---|---|
| Jim | -0.65 | 0.65 | -0.65 | 1.96 | 0 | -0.65 | | | -0.7 | |
| Sean | -1 | | -0.14 | -1 | | 0.71 | | 1.57 | | -0.14 |
| John | | -1 | | 0.24 | | 1.434 | | | -1 | 0.24 |
| Sidd | 0.783 | | | | -0.26 | | 0.78 | | -1.3 | |
| Penny | 1.069 | | -0.53 | | -0.53 | | 1.07 | | -1.1 | |
| Pete | | 1.15 | | | ? | | -0.6 | | | -0.58 |

# Project

- Study the papers
  - http://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf
  - http://blog.yhathq.com/posts/recommender-system-in-r.html
  - http://www2.research.att.com/~volinsky/papers/ieeecomputer.pdf

## International School of Engineering


Plot 63/A, Floors 1&2, Road # 13, Film Nagar, Jubilee Hills, Hyderabad - 500 033

For Individuals: +91-9502334561/63 or 040-65743991

For Corporates: +91-9618483483

Web: http://www.insofe.edu.in

Facebook: https://www.facebook.com/insofe

Twitter: https://twitter.com/Insofeedu

YouTube: http://www.youtube.com/InsofeVideos

SlideShare: http://www.slideshare.net/INSOFE

LinkedIn: http://www.linkedin.com/company/international-school-of-engineering