# Regression

# Regression

- In classification, class values or labels are categories {Play Golf, Don't Play Golf}
- In regression, the class attribute takes real values

$$y \approx f(X)$$

Class attribute(Dependent variable)
$y \in R$

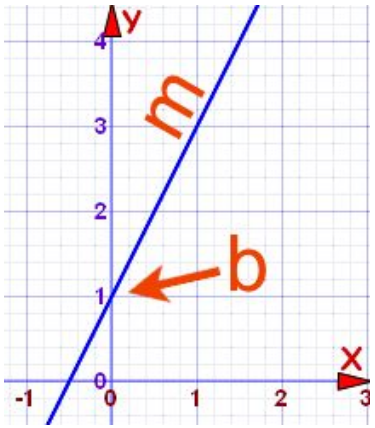
Features (Regressors)
$x_1, x_2, \ldots, x_m$

Regression finds the relation between y and the vector $(x_1, x_2, \ldots, x_m)$

# Linear Regression

In linear regression, we assume the relation between the class attribute $y$ and feature set $\boldsymbol{x}$ to be linear:

$$y = mx + b$$




$$y = \sum_{i=0}^{n} x_i w_i \qquad x_0 = 1$$

where $\boldsymbol{w}$ represents the vector of regression coefficients

- Regression can be solved by estimating *the weights* from the training data
  - Least squares is often used to solve the problem

# Solving Linear Regression Problems

- Regression can be solved by estimating *the weights* from the training data
  - "Least squares" is a popular method to solve regression problems

$$\epsilon^2 = \|\epsilon^2\| = \|Y - XW\|^2$$

# Least Squares

Find W such that it minimizes ‖Y - XW‖² for regressors X and labels Y

$$min||Y - XW||^2$$

$$\frac{\partial}{\partial W}||Y - XW||^2 = 0$$

$$||X||^2 = X^T X \Rightarrow \frac{\partial}{\partial W}(Y - XW)^T(Y - XW) = 0$$

$$\frac{\partial}{\partial W}(Y^T - W^T X^T)(Y - XW) = 0$$

$$\frac{\partial}{\partial W}(Y^T Y - Y^T XW - W^T X^T Y + W^T X^T XW) = 0$$

$$-2X^T Y + 2X^T XW = 0$$

$$2X^T Y = 2X^T XW$$

$$W = (X^T X)^{-1} X^T Y$$

# Evaluation

# Evaluating Supervised Learning

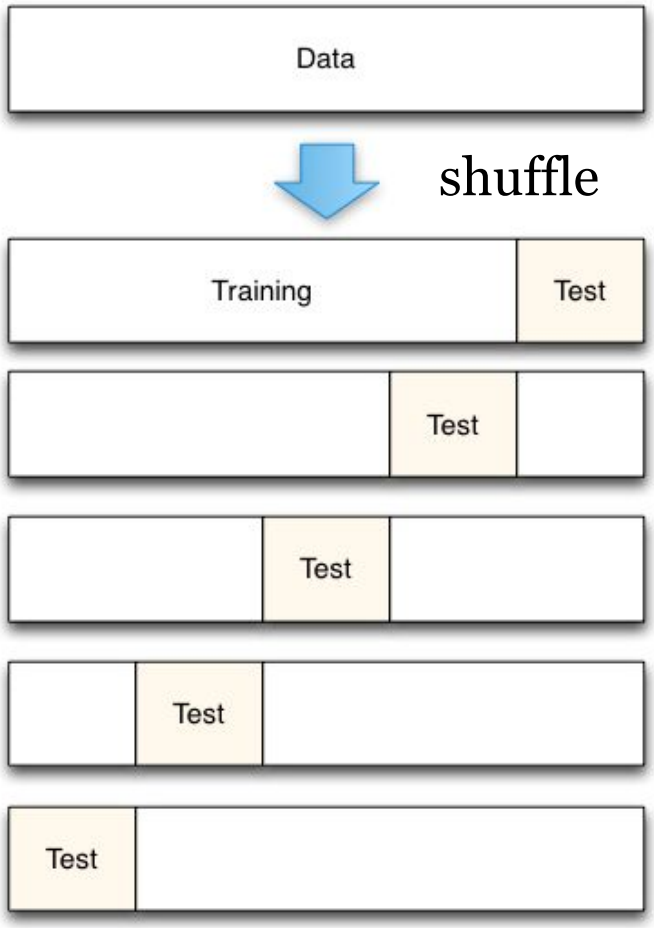
- To evaluate we use a training-testing framework
  - A training dataset (i.e., the labels are known) is used to train a model
  - the model is evaluated on a test dataset.

- Since the correct labels of the test dataset are unknown, in practice, the training set is divided into two parts:
  - one used for training and
  - the other used for testing.

- When testing, the labels from this test set are removed. After these labels are predicted using the model, the predicted labels are compared with the masked labels (ground truth).

# Evaluating Supervised Learning

- K-Fold Cross Validation
  - Shuffle the data.
  - Divide the training set into *k* equally sized sets.
  - Run the algorithm *k* times.
  - The average performance of the algorithm over *k* rounds measures the performance of the algorithm.



# Evaluating Classification

- As the class labels are discrete, we can measure the accuracy by dividing number of correctly predicted labels (C) by the total number of predictions (N)
  - Accuracy = C/N
  - Error rate = 1 – Accuracy

| | | Actual | |
|---|---|---|---|
| | | Yes | No |
| **Predicted** | Yes | True Pos. | False Pos. |
| | No | False Neg. | True Neg. |

$$P = \frac{TP}{TP + FP}$$

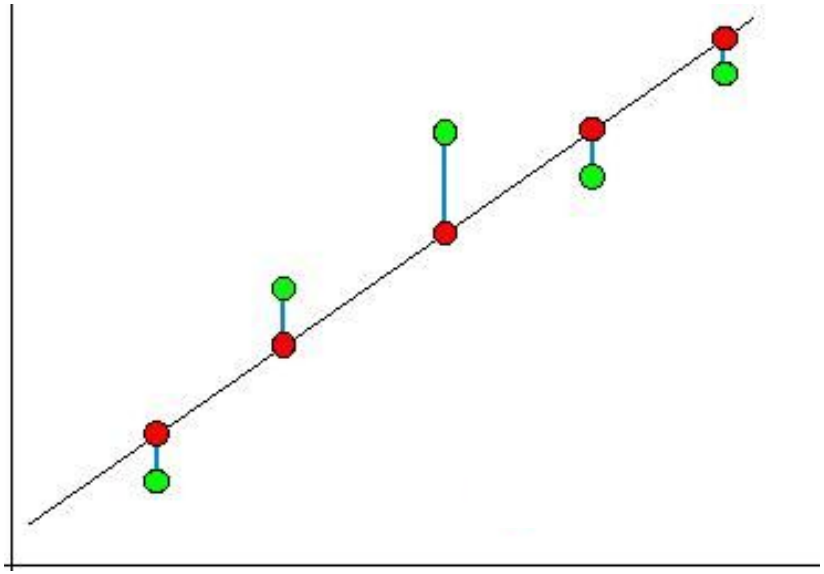
$$R = \frac{TP}{TP + FN}$$

$$F_1 = \frac{2PR}{P + R}$$

# Evaluating Regression Performance

- The labels cannot be predicted *exactly*.
- It is needed to set a margin to accept or reject the predictions
  - For example, when the observed temperature is 71 any prediction within 71±0.5 can be considered correct
- RMSE:

$$\sqrt{\frac{\sum_{t=1}^{n}(\hat{y}_t - y_t)^2}{n}}$$



# Unsupervised Learning

# Unsupervised Learning

**Unsupervised division of instances into groups of similar objects**

- Clustering is a form of **unsupervised learning**
  - The clustering algorithms do not have examples showing how the samples should be grouped together (unlabeled data)
- Clustering algorithms group together **similar items**

# Measuring Distance/Similarity in Clustering Algorithms

- **The goal of clustering:**
  - to group together similar items
- Instances are put into different clusters based on the distance to other instances
- **Any clustering algorithm requires a distance measure**

The most popular (dis)similarity measure for continuous features are ***Euclidean Distance***

# Similarity Measures: More Definitions

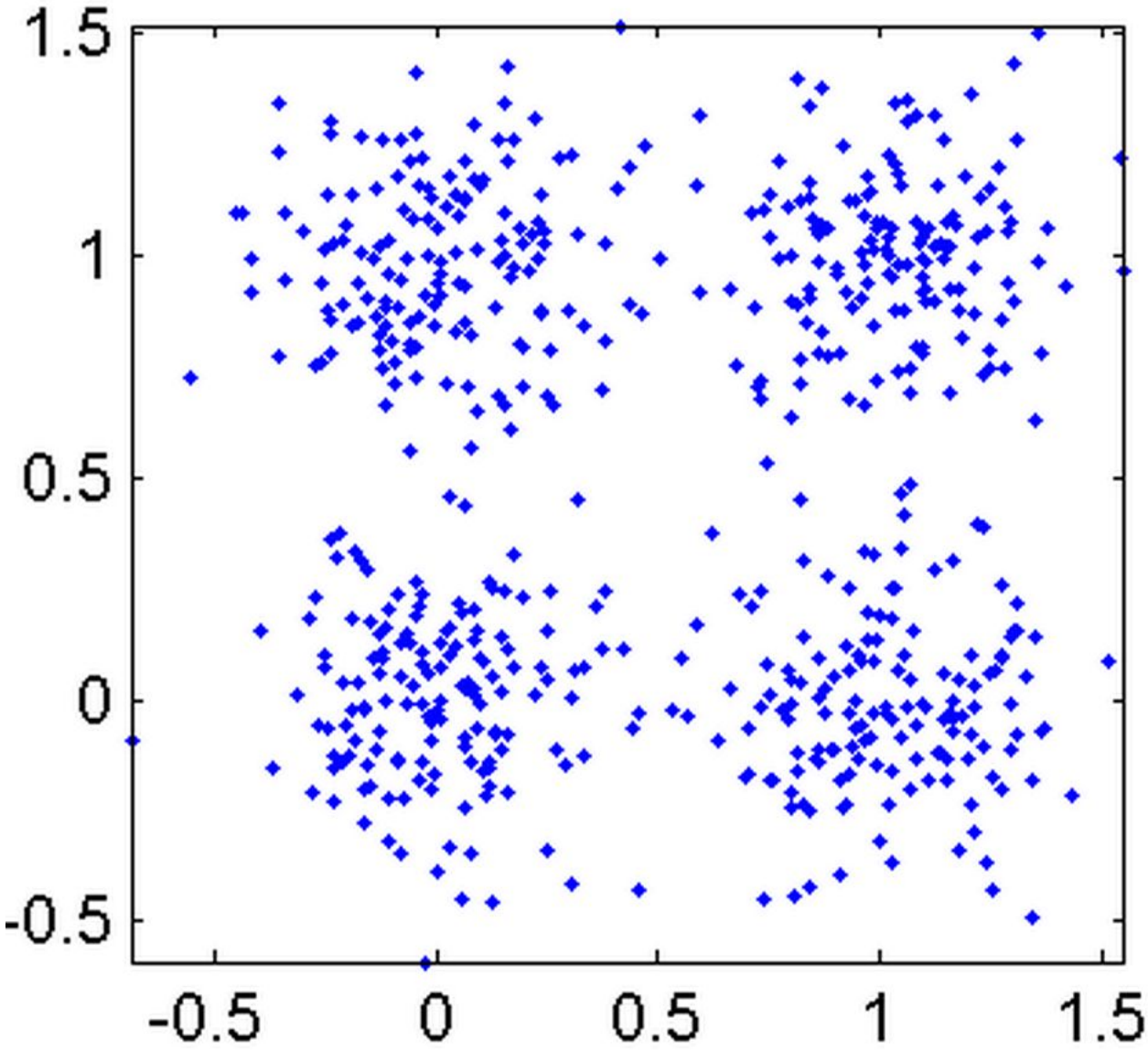
| Measure Name | Formula | Type | Description |
|---|---|---|---|
| Mahalanobis | $d(X,Y) = \sqrt{(X-Y)^T Co^{-1}(X-Y)}$ | Dissimilarity | X, Y are features vectors and *Co* is the Covariance matrix of the dataset |
| Manhattan | $d(X,Y) = \sum_i \lvert x_i - y_i \rvert$ | Dissimilarity | X, Y are features vectors |
| $L_p$-norm | $d(X,Y) = (\sum_i \lvert x_i - y_i \rvert^n)^{\frac{1}{n}}$ | Dissimilarity | X, Y are features vectors |
| Cosine | $c(X,Y) = \frac{X.Y}{\lvert X \rvert \lvert Y \rvert}$ | Similarity | X, Y are features vectors and '.' represents the inner product |

Once a distance measure is selected, instances are grouped using it.

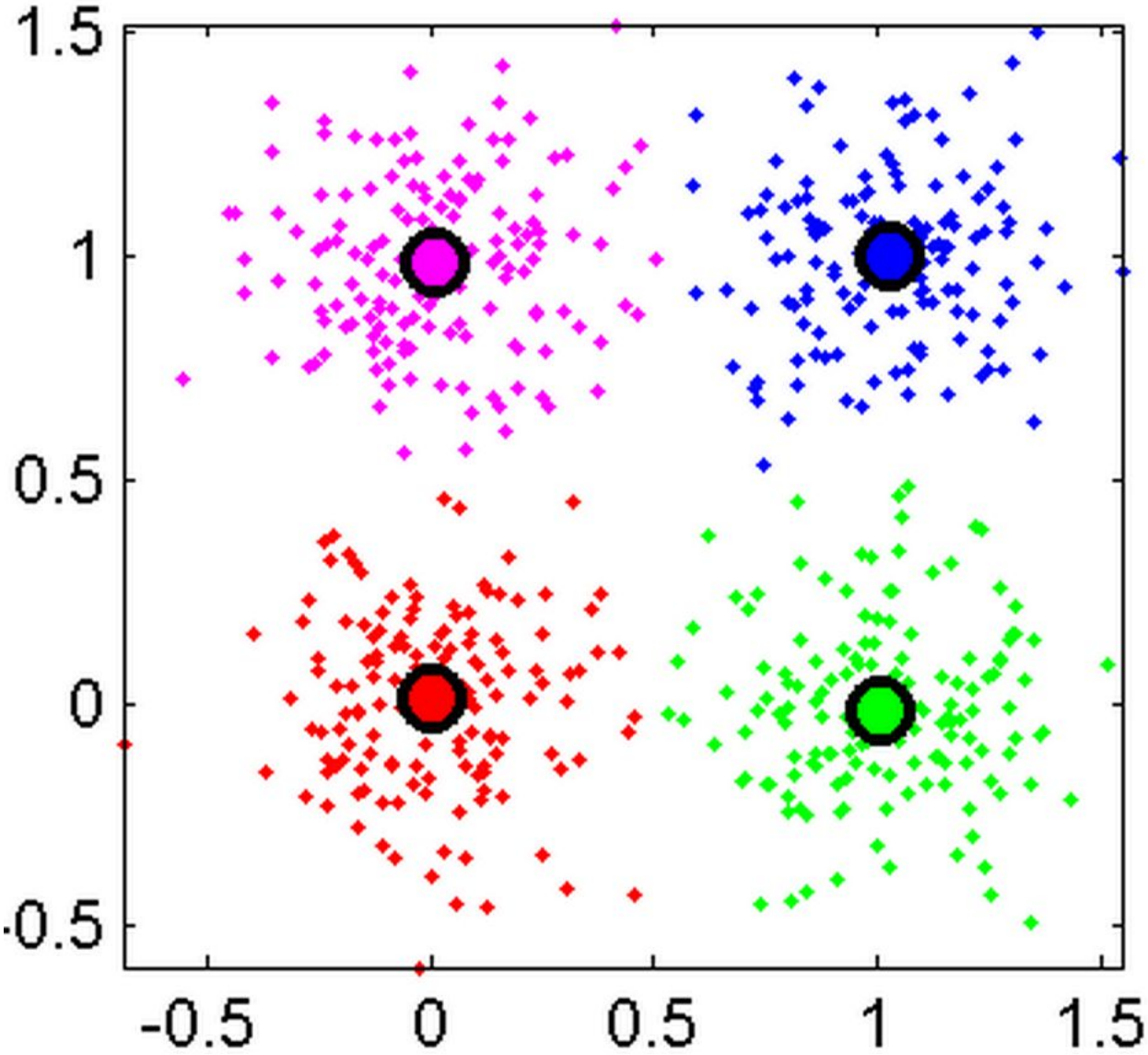
# Clustering

- Clusters are usually represented by compact and abstract notations.
- "Cluster centroids" are one common example of this abstract notation.
- Partitional Algorithms
  - Partition the dataset into a set of clusters
  - Each instance is assigned to a cluster exactly **once** and no instance remains unassigned to clusters.
  - k-Means

# k-Means Example



# k-Means Example



# k-Means

The algorithm is the most commonly used clustering algorithm.

**Algorithm 2** *K*-Means Algorithm

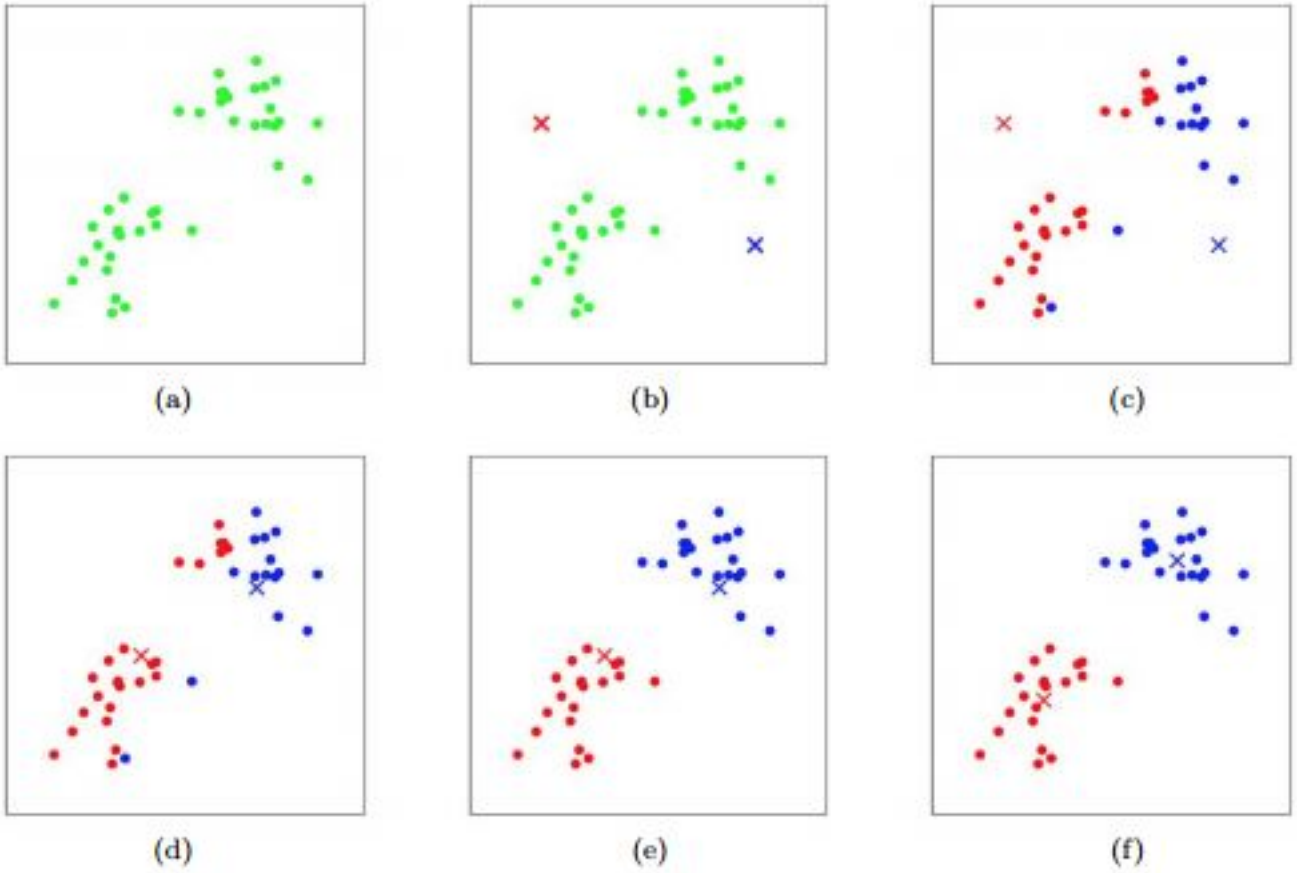
**Require:** A Dataset of Real-Value Attributes, *K* (number of Clusters)

1: **return** A Clustering of Data into *K* Clusters
2: Consider *K* random points in the data space as the initial cluster centroids.
3: **while** centroids have not converged **do**
4: Assign each data point to the cluster which has the closest cluster centroid.
5: If all data points have been assigned then recalculate the cluster centroids by averaging datapoints inside each cluster
6: **end while**

# When do we stop?

- Note that this procedure is repeated until convergence.
- The most common criterion to determine convergence is to check whether centroids are no longer changing.
- This is equivalent to clustering assignments of the data instances stabilizing.
- In practice, the algorithm execution can be stopped when the Euclidean distance between the centroids in two consecutive steps is bounded above by some small positive

# When do we stop?



# k-Means

- As an alternative, k-means implementations try to minimize an objective function. A well-known objective function in these implementations is the squared distance error:

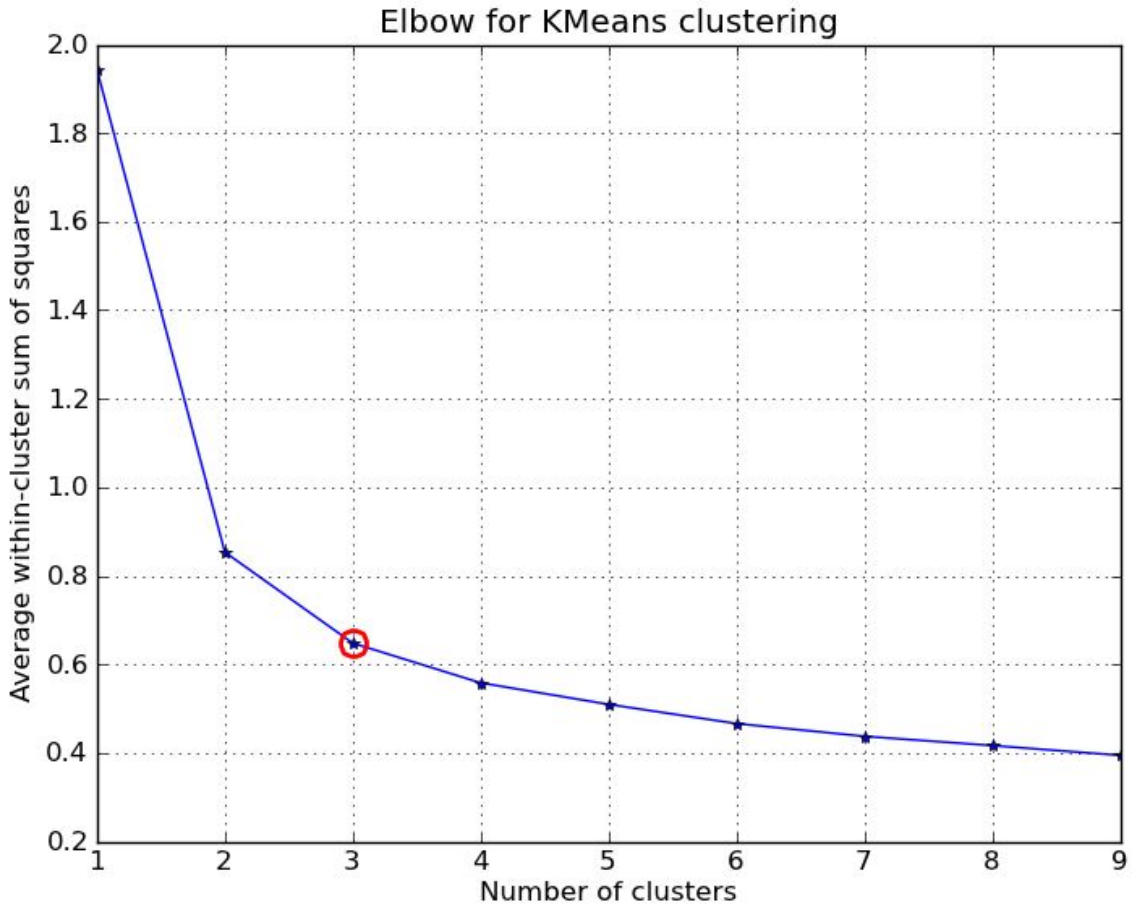
$$\sum_{i=1}^{k} \sum_{j=1}^{n(i)} \|x_j^i - c_i\|^2,$$

- where $x_j^i$ is the jth instance of cluster i, n(i) is the number of instances in cluster i, and ci is the centroid of cluster i.

- The process stops when the difference between the objective function values of two consecutive iterations of the k-means algorithm is bounded by some small value .

# Finding "k"

- Elbow Method



# k-Means

- Finding the global optimum of the *k* partitions is computationally expensive (NP-hard).
- This is equivalent to finding the optimal centroids that minimize the objective function

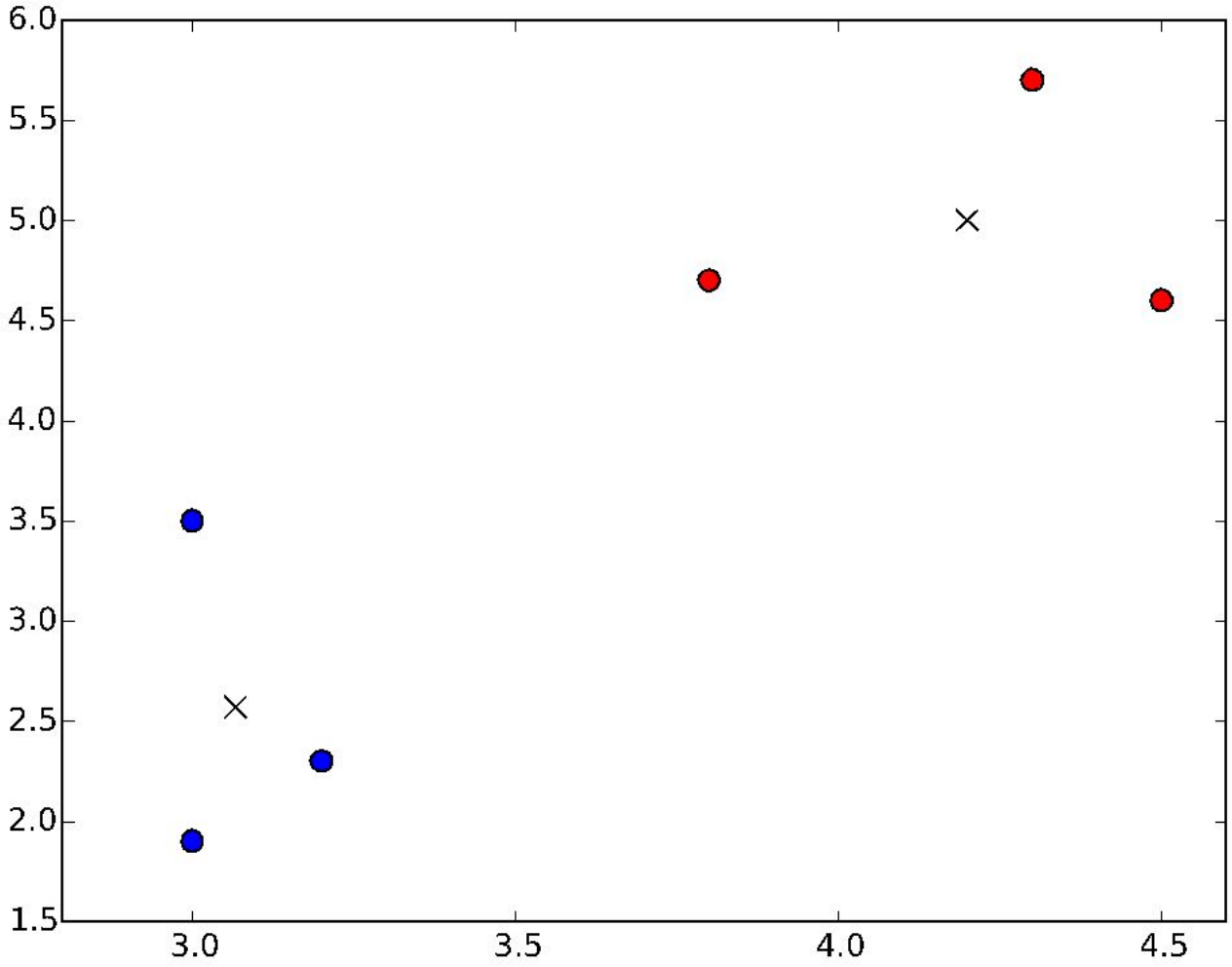
- However, there are efficient heuristic algorithms that are commonly employed and converge quickly to an optimum that might not be global.
  - running k-means multiple times and selecting the clustering assignment that is observed most often or is more desirable based on an objective function, such as the squared error.

# K-Means Example

| ID | Feature 1 | Feature 2 |
|---|---|---|
| 1 | 3.0 | 3.5 |
| 2 | 4.5 | 4.6 |
| 3 | 3.8 | 4.7 |
| 4 | 4.3 | 5.7 |
| 5 | 3.2 | 2.3 |
| 6 | 3.0 | 1.9 |

K = 2
Starting centroids = {1, 6}

# K-Means Example



# Evaluation with Ground Truth

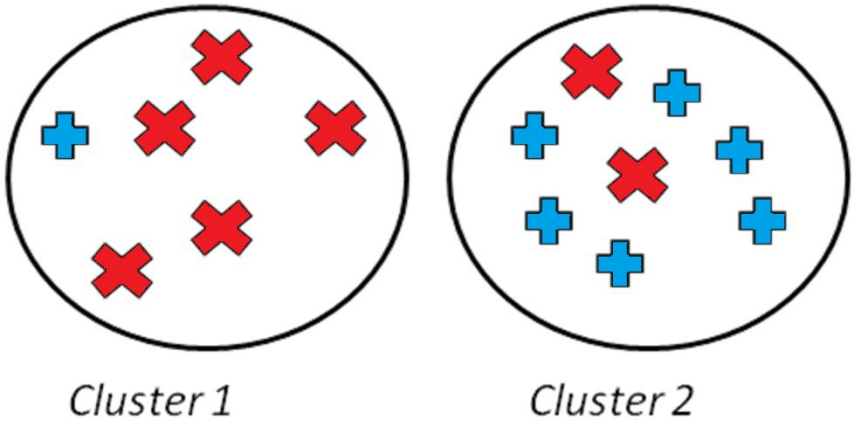
When ground truth is available, the evaluator has prior knowledge of what a clustering should be

- That is, we know the correct clustering assignments.

- We will discuss these methods in community analysis chapter

# Evaluating the Clusterings

When we are given objects of two different kinds, the perfect clustering would be that objects of the same type are clustered together.




- Evaluation with ground truth
- Evaluation without ground truth

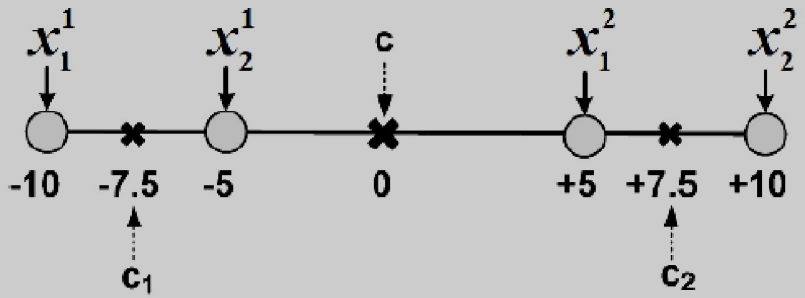
# Evaluation without Ground Truth

- Cohesiveness
  - In clustering, we are interested in clusters that exhibit cohesiveness.
  - In cohesive clusters, instances inside the clusters are close to each other.
- Separateness
  - We are also interested in clusterings of the data that generates clusters that are well separated from one another

# Cohesiveness

- Cohesiveness
  - In statistical terms, this is equivalent to having a small standard deviation, i.e., being close to the mean value.
  - In clustering, this translates to being close to the centroid of the cluster

$$cohesiveness = \sum_{i=1}^{k} \sum_{j=1}^{n(i)} dist(x_j^i, c_i)^2$$



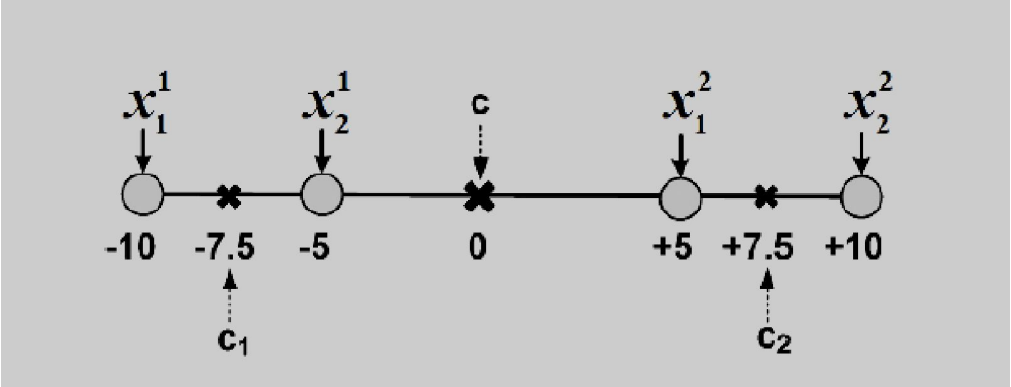

$$cohesiveness = |-10-(-7.5)|^2+|-5-(-7.5)|^2+|5-7.5)|^2+|10-7.5|^2 = 25. \quad (5.59)$$

# Separateness

- Separateness
  - We are also interested in clusterings with clusters that are well separated from one another.
  - In statistics, separateness can be measured by standard deviation.
  - Standard deviation is maximized when instances are far from the mean.
  - In clustering terms, this is equivalent to cluster centroids being far from the mean of the entire dataset

$$separateness = \sum_{i=1}^{k} dist(c, c_i)^2$$

# Separateness Example




$$separateness = |-7.5-0|^2 + |7.5-0|^2 = 112.5.$$

- In general we are interested in clusters that are both cohesive and separate -> Silhouette index

# Silhouette Index

- The silhouette index combines both cohesiveness and separateness.
- It compares the average distance value between instances in the same cluster and the average distance value between instances in different clusters.
- In a well-clustered dataset, the average distance between instances in the same cluster is small (cohesiveness) and the average distance between instances in different clusters is large (separateness).

# Silhouette Index

- For any instance x that is a member of cluster C

- Compute the within-cluster average distance

$$a(x) = \frac{1}{|C| - 1} \sum_{y \in C, y \neq x} \|x - y\|^2.$$

- Compute the average distance between x and instances in cluster G that is closest to x in terms of the average distance between x and members of G

$$b(x) = \min_{G \neq C} \frac{1}{|G|} \sum_{y \in G} \|x - y\|^2.$$

# Silhouette Index

- Clearly we are interested in clusterings where a(x)<b(x)

$$s(x) = \frac{b(x) - a(x)}{\max(b(x), a(x))},$$

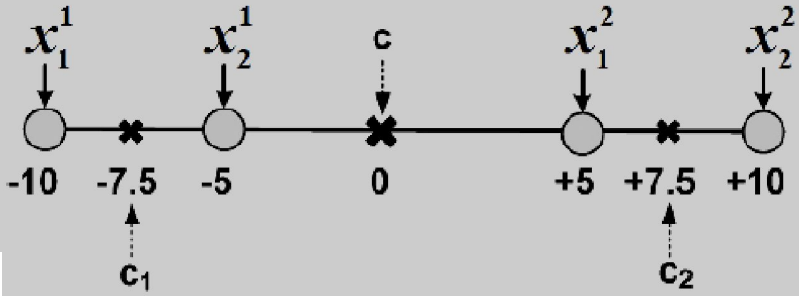
$$silhouette = \frac{1}{n} \sum_{x} s(x).$$

- Silhouette can take values between [-1,1]
- The best case happens when for all x,
  - a(x)=0, b(x)>a(x)

# Silhouette Index - Example

$$a(x) = \frac{1}{|C| - 1} \sum_{y \in C, y \neq x} \|x - y\|^2.$$

$$b(x) = \min_{G \neq C} \frac{1}{|G|} \sum_{y \in G} \|x - y\|^2.$$




$$s(x) = \frac{b(x) - a(x)}{\max(b(x), a(x))},$$

$$silhouette = \frac{1}{n} \sum_{x} s(x).$$

# Silhouette Index - Example

$$
\begin{aligned}
s(x_1^1) &= \frac{312.5 - 25}{312.5} = 0.92 \\
a(x_2^1) &= |-5-(-10)|^2 = 25 \\
b(x_2^1) &= \frac{1}{2}(|-5-5|^2 + |-5-10|^2) = 162.5 \\
s(x_2^1) &= \frac{162.5 - 25}{162.5} = 0.84 \\
a(x_1^2) &= |5-10|^2 = 25 \\
b(x_1^2) &= \frac{1}{2}(|5-(-10)|^2 + |5-(-5)|^2) = 162.5 \\
s(x_1^2) &= \frac{162.5 - 25}{162.5} = 0.84 \\
a(x_2^2) &= |10-5|^2 = 25 \\
b(x_2^2) &= \frac{1}{2}(|10-(-5)|^2 + |10-(-10)|^2) = 312.5 \\
s(x_2^2) &= \frac{312.5 - 25}{312.5} = 0.92.
\end{aligned}
$$