Birla Institute of Applied Sciences
बिरला इंस्टिट्यूट ऑफ़ अप्लाइड साइंसेस
Bhimtal, Distt: Nainital, Uttarakhand- 263136

Pattern Recognition

- S. S. Samant

Normalization and Standardization

Standardization (Z-score normalization)

- Variables that are measured at different scales do not contribute equally to the analysis and might end up creating a bias. We use formula:

$$Z = \frac{\overset{\text{Score}}{x} - \overset{\text{Mean}}{\mu}}{\underset{\text{SD}}{\sigma}}$$

Normalization and Standardization

Standardization (Z-score normalization)

- Variables that are measured at different scales do not contribute equally to the analysis and might end up creating a bias.
- Ex. a variable that ranges between 0 and 100 will outweigh a variable that ranges between 0 and 1. Without standardization, the variable with the larger range gets higher weight of 100.
- Assumes that the data has a Gaussian (bell curve) distribution.
- In sklearn, we can use:

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaled_data = scaler.fit_transform(unscaled_data)
```

Normalization and Standardization

Normalization (Min-Max Scalar)

- In this data is scaled to a fixed range, usually [0,1].
- Scaling is typically done via the following equation:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

- In sklearn, we can use:

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
scaled_data = scaler.fit_transform(unscaled_data)
```

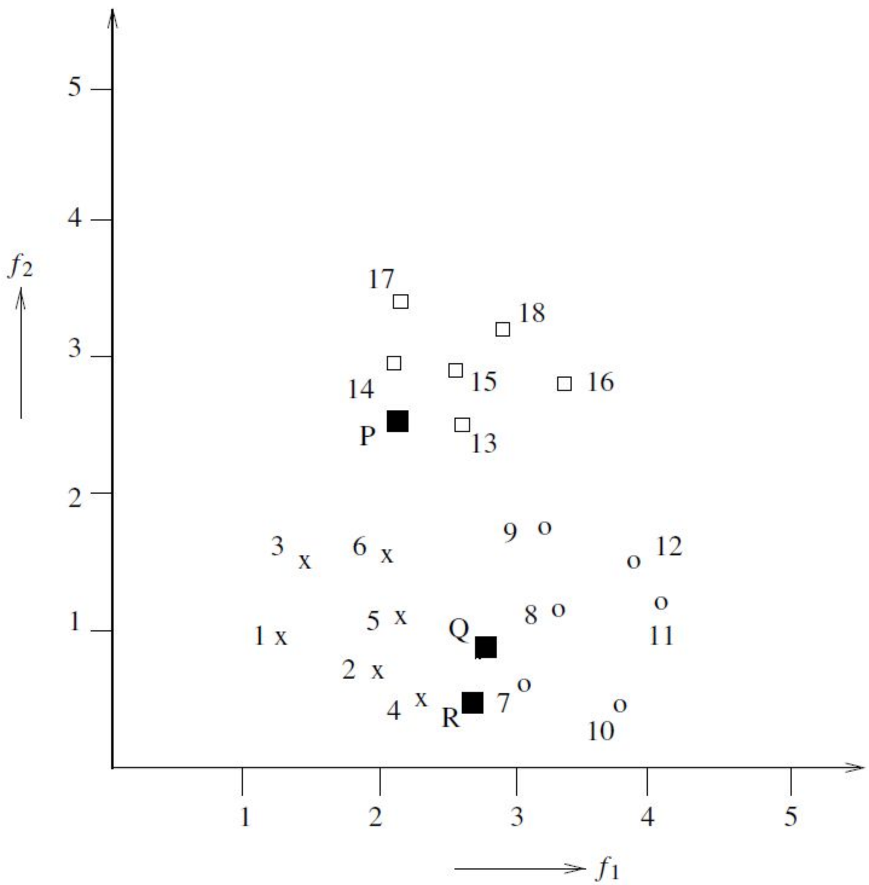
Nearest-Neighbor Variants, Prototype Selection

Nearest Neighbor - Variants

- 1NN or 1-Nearest Neighbour
  - Similar to kNN with k=1
- Modified k-Nearest Neighbors (mkNN)

Modified k-Nearest Neighbors

- A variant of kNN classifier where the class of k-Nearest neighbors is used (but not directly as in kNN)

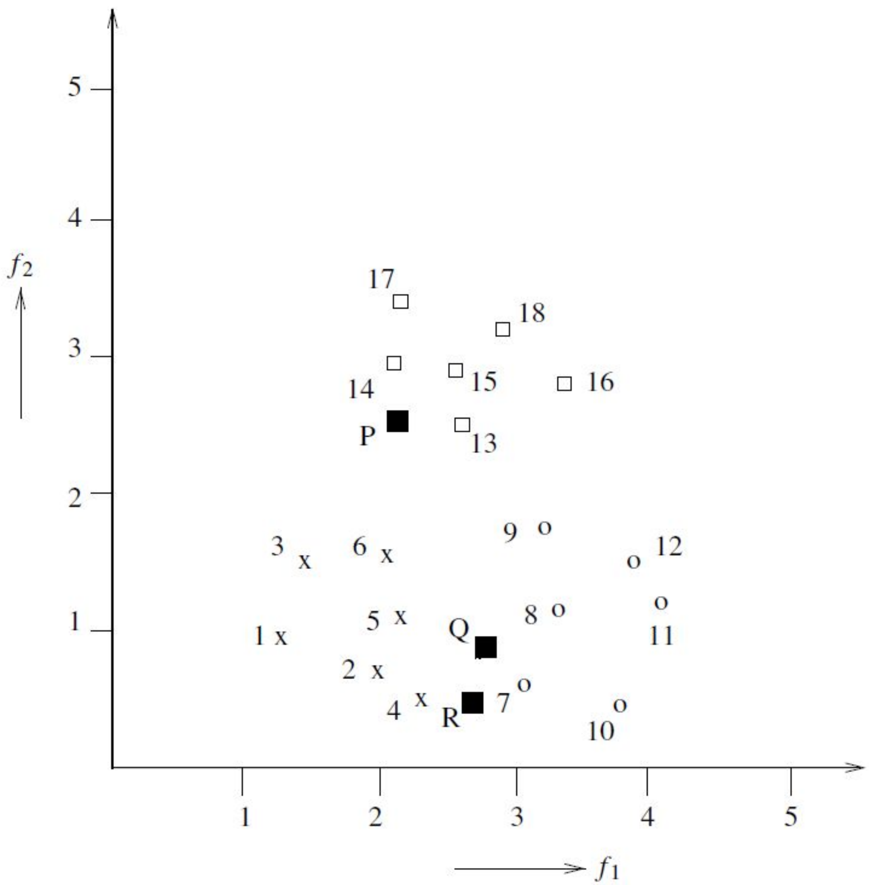


Example dataset.

- Two features f1 and f2
- Using kNN (k=5), P will be classified to class *square*, Q and R to class *cross*

Modified k-Nearest Neighbors

- A variant of kNN classifier where the class of k-Nearest neighbors is used (but not directly as in kNN)



Example dataset.

mkNN

Out of k-nearest neighbors, if d_{min} is the distance of the nearest neighbor and d_{max} is the distance of the farthest neighbor, then *weight of the class* of each of these k-neighbors is normalized using:

$$w_i = \frac{(d_{\max} - d_i)}{(d_{\max} - d_{\min})}$$

w_1 is set to 1.

- Sum weights for each class, the pattern belongs to the class with the highest score

Prototype Selection

- The nearest neighbor classifiers require time linear in the sample size for classification that goes up as the training data size goes up
- If the training dataset size can be reduced, the time required for classification can be reduced
- This reduction can be accomplished either by *reducing the number of training patterns*, *reducing the number of features*, or both
- Reducing the number of training patterns - *prototype selection*.
  - For example, Condensed Nearest Neighbor (CNN) method

Condensed Nearest Neighbor (CNN) algorithm

The CNN starts with a set of all patterns *Data* and a condensed set *Condensed* which is initially empty.

Data is scanned randomly, or better, using some method.

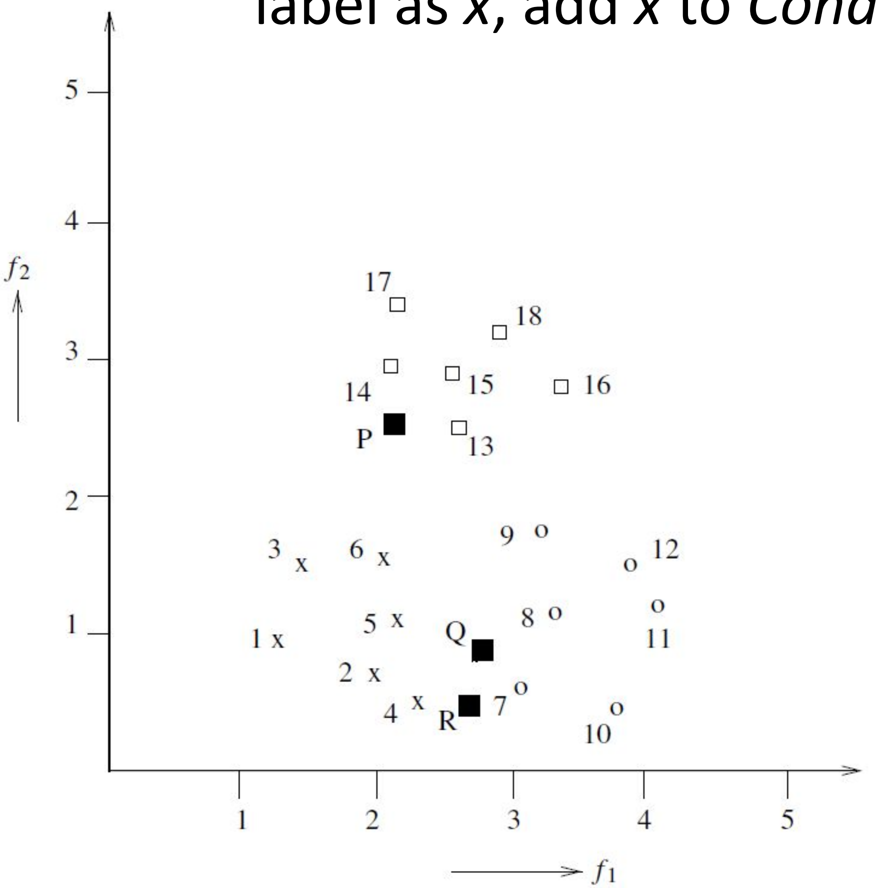
The first pattern in *Data* is put into *Condensed*. After this, the following set of statements are repeated until there is no change in *Condensed* in an iteration:

1. For every pattern *x* in *Data*, find its NN in *Condensed*
2. If the nearest neighbor does not have the same class label as *x*, remove x from Data and add it to *Condensed*

NOTE: Usually, there is an initial preprocessing step for outlier removal as we will see in an example.

Condensed Nearest Neighbor (CNN) algorithm

1. For every pattern *x* in *Data*, find its NN in *Condensed*
2. If the nearest neighbor does not have the same class label as *x*, add *x* to *Condensed*



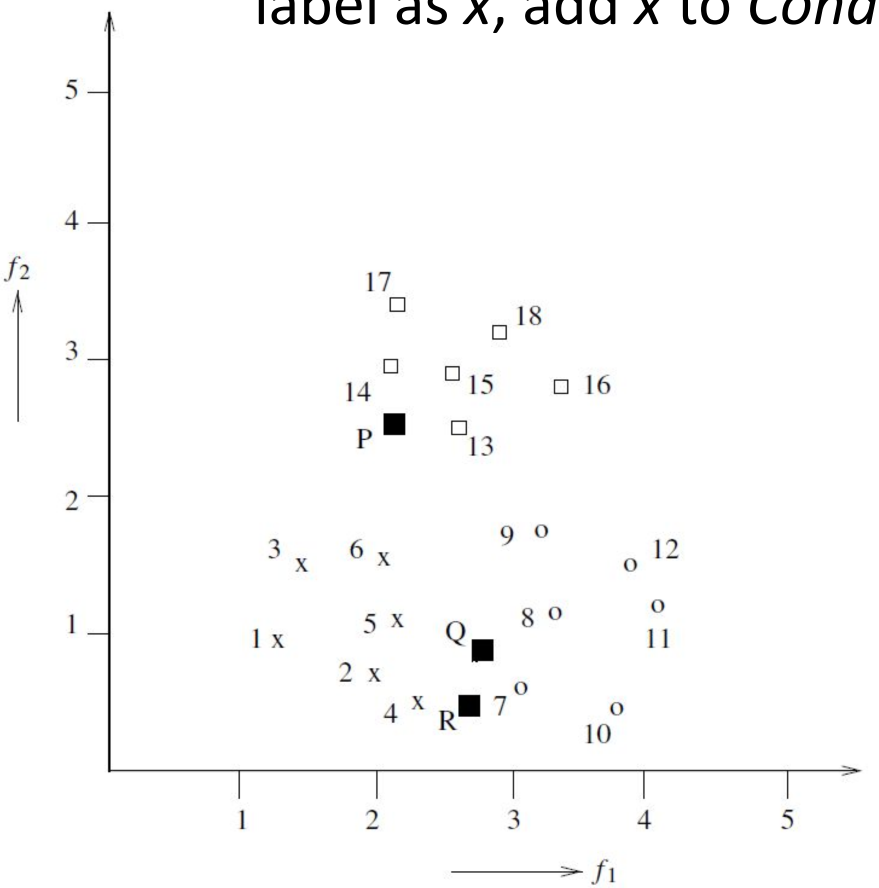
Example dataset.

- If the patterns are scanned by the algorithm in the order $x1$, $x2$, $x3$, $x4$, $x5$, $x6$, $x7$, ... , pattern $x1$ belonging to class 'cross' will be first put into *Condensed*.
- $x2$, $x3$, $x4$, $x5$, and $x6$ will be left out and $x7$ which belongs to class 'circle' will be put into *Condensed*
- So in the first iteration, the first pattern presented to the algorithm from each class will be included in *Condensed*, along with other patterns

Condensed Nearest Neighbor (CNN) algorithm

1. For every pattern *x* in *Data*, find its NN in *Condensed*
2. If the nearest neighbor does not have the same class label as *x*, add *x* to *Condensed*

- Patterns put into *Condensed* depend on the order in which the patterns are presented to the algorithm. It is therefore an **order-dependent algorithm**



Example dataset.

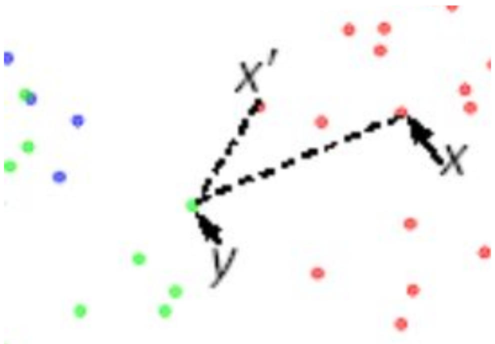
Condensed Nearest Neighbor (CNN) algorithm

1. For every pattern *x* in *Data*, find its NN in *Condensed*
2. If the nearest neighbor does not have the same class label as *x*, add *x* to *Condensed*

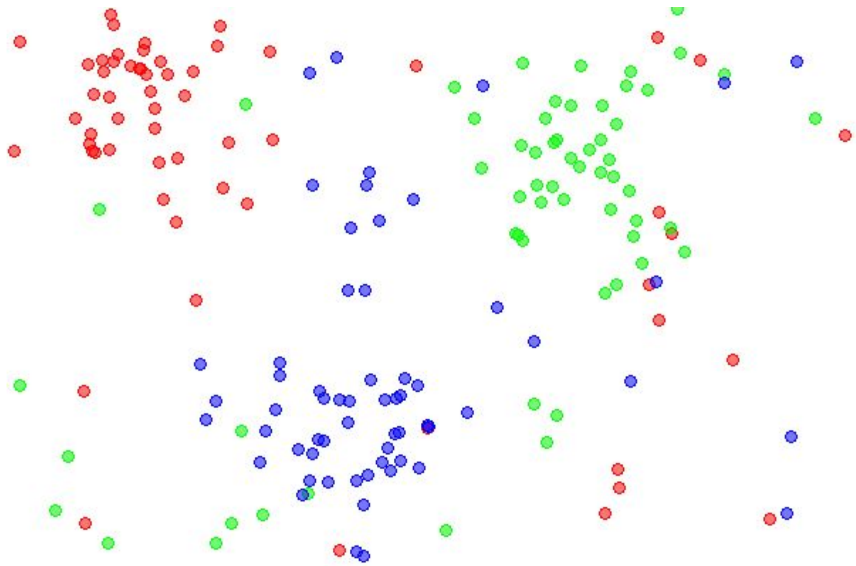
- Efficient to scan patterns in order of decreasing border ratio. Border ratio of point x is:

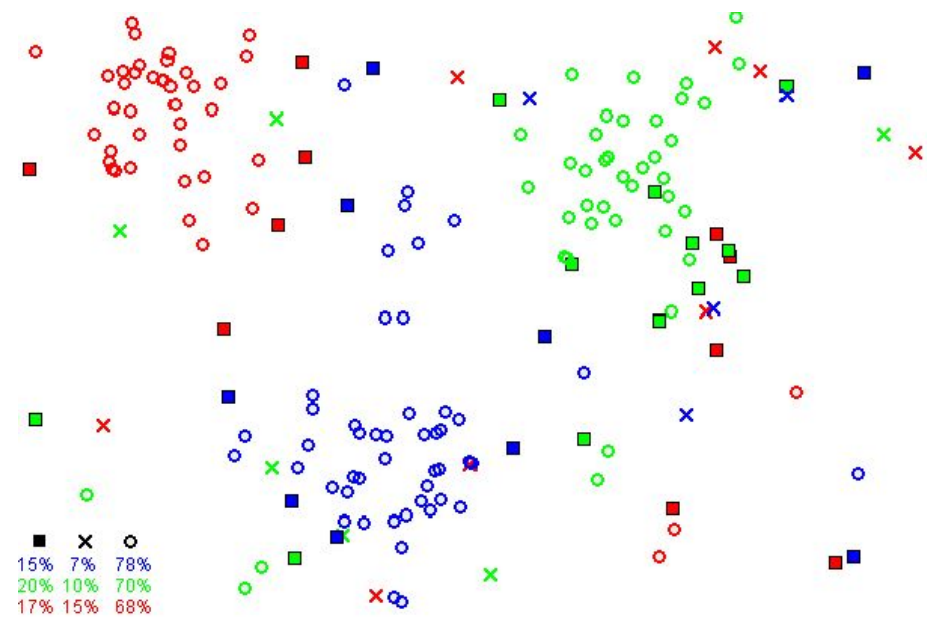
$$a(x) = \frac{||x'-y||}{||x-y||}$$

where $||x-y||$ is the distance to the closest example *y* having a different color than *x*, and $||x'-y||$ is the distance from *y* to its closest example *x'* with the same label as *x*.

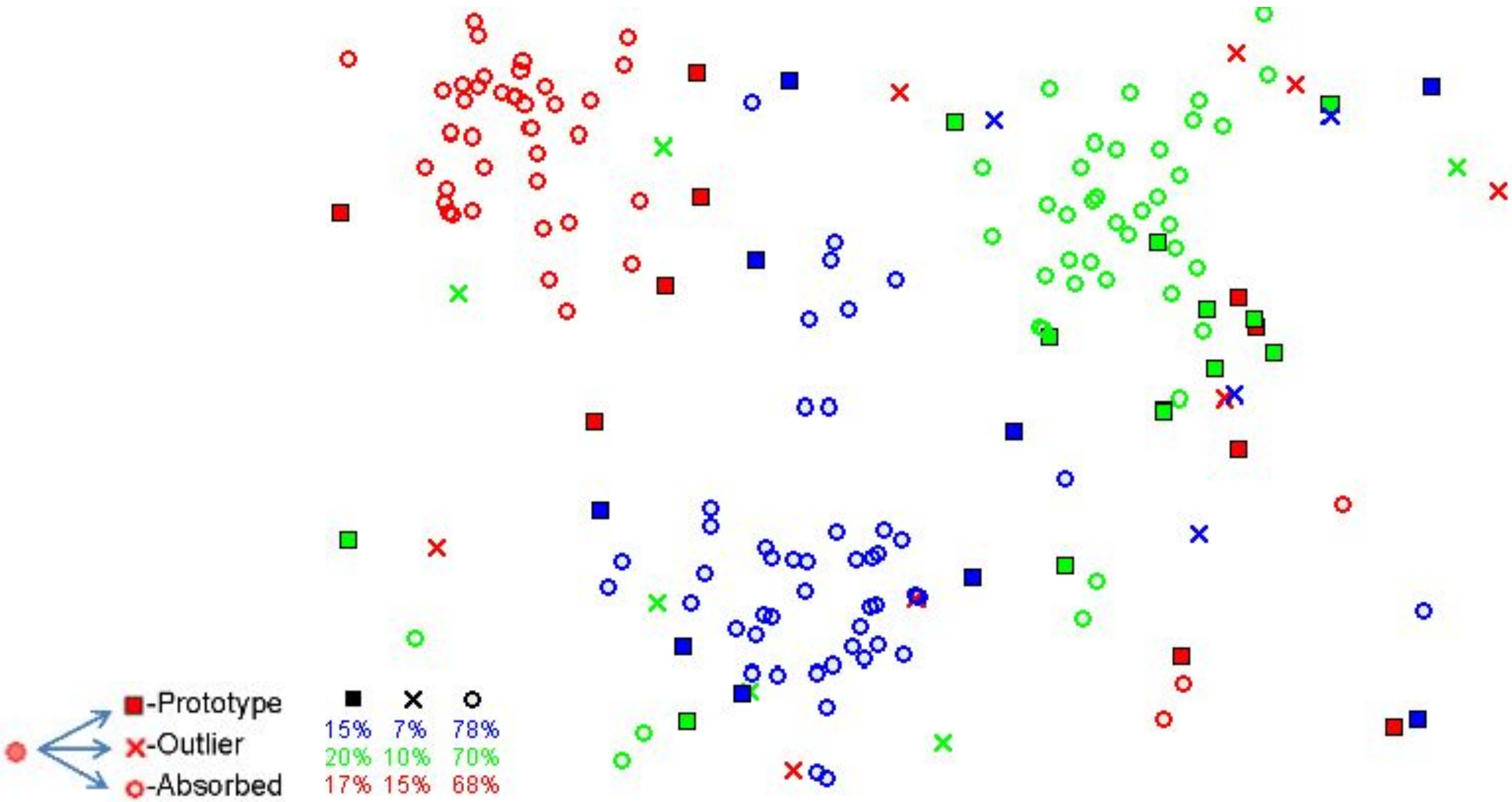


Condensed Nearest Neighbor - Example





Condensed Nearest Neighbor - Example



the class outliers selected by the (3,2)NN rule (all the three nearest neighbors of these instances belong to other classes)

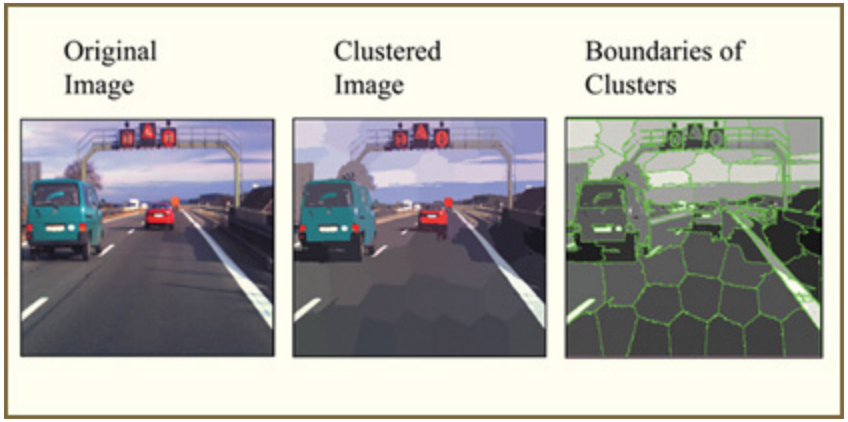
*(k,r)*NN class-outlier: *k* nearest neighbors include more than *r* examples of other classes

Clustering

- Unsupervised learning
- No training labels available during clustering
  - However, some labels *may* be used during evaluation of clustering algorithms

Clustering

- Divide data into similar groups
- Goals of cluster analysis:
  - Clustering for understanding data (Information retrieval, climate, business)
  - Clustering as a utility (summarization, compression)



Clustering

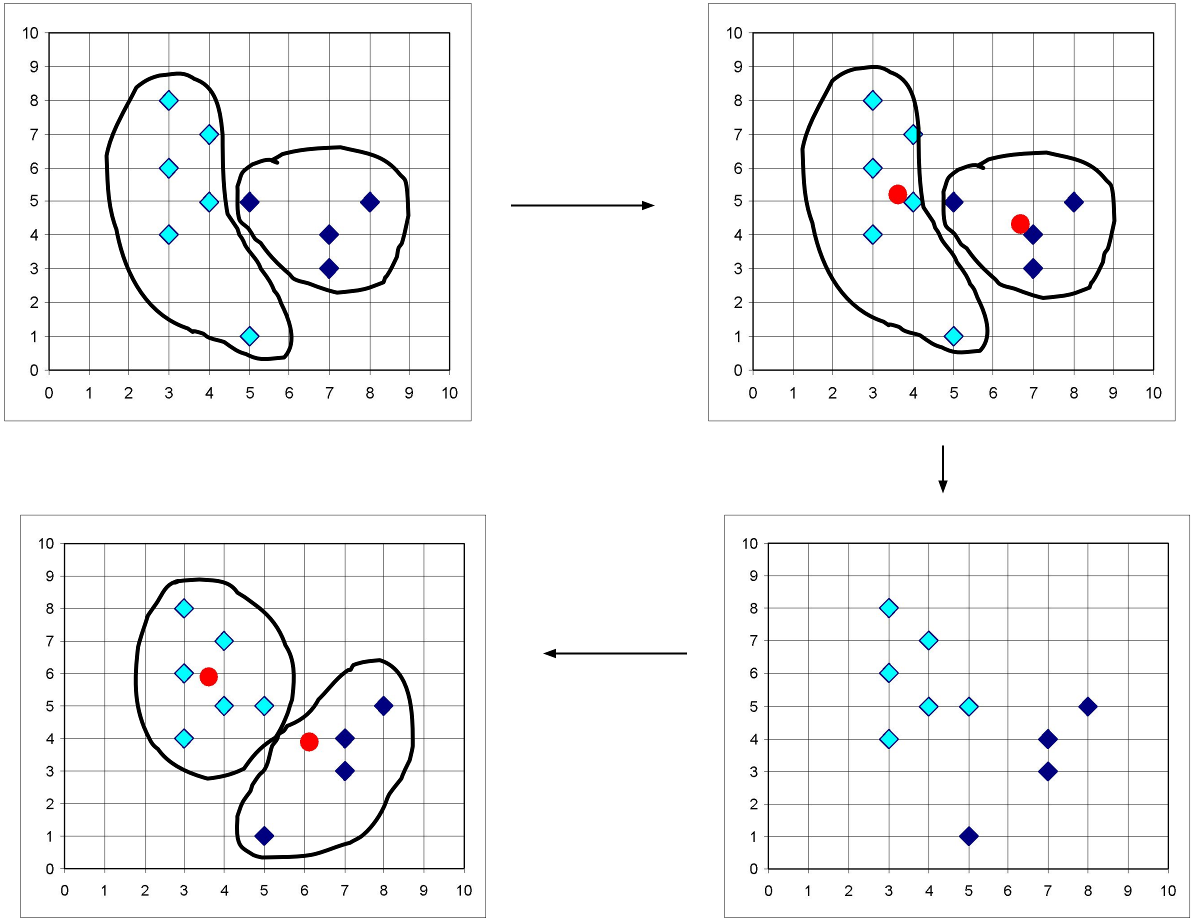
- Divide data into similar groups
- Goals of cluster analysis:
  - Clustering for understanding data (Info. retrieval, climate, business)
  - Clustering as a utility (summarization, compression)
- Examples of clustering algorithms: *k-means, hierarchical agglomerative clustering*, DBSCAN

K-means Clustering

- Given *k*, the *k-means* algorithm consists of four steps:
  - Select initial centroids at random.
  - Assign each object to the cluster with the nearest centroid.
  - Compute each centroid as the mean of the objects assigned to it.
  - Repeat previous 2 steps until there is no change.

K-means Clustering

- Example



K-means example (K=2)

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

K-means example (K=2)

Initialization: Randomly we choose following two centroids (k=2) for two clusters.
In this case the 2 centroid are: m1=(1.0,1.0) and m2=(5.0,7.0).

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

	Individual	Mean Vector
Group 1	1	(1.0, 1.0)
Group 2	4	(5.0, 7.0)

K-means example (K=2)

- Thus, we obtain two clusters containing:
  {1,2,3} and {4,5,6,7}.
- Their new centroids are:

$$m_1 = \left(\frac{1}{3}(1.0+1.5+3.0), \frac{1}{3}(1.0+2.0+4.0)\right) = (1.83, 2.33)$$

$$m_2 = \left(\frac{1}{4}(5.0+3.5+4.5+3.5), \frac{1}{4}(7.0+5.0+5.0+4.5)\right)$$

$$= (4.12, 5.38)$$

Individual	Centroid 1	Centroid 2
1	0	7.21
2 (1.5, 2.0)	1.12	6.10
3	3.61	3.61
4	7.21	0
5	4.72	2.5
6	5.31	2.06
7	4.30	2.92

$$d(m_1, 2) = \sqrt{|1.0-1.5|^2 + |1.0-2.0|^2} = 1.12$$

$$d(m_2, 2) = \sqrt{|5.0-1.5|^2 + |7.0-2.0|^2} = 6.10$$

K-means example (K=2)

- Now using these centroids we compute the Euclidean distance of each object, as shown in table.

Individual	Centroid 1	Centroid 2
1	1.57	5.38
2	0.47	4.28
3	2.04	1.78
4	5.64	1.84
5	3.15	0.73
6	3.78	0.54
7	2.74	1.08

- Therefore, the new clusters are:

  {1,2} and {**3**,4,5,6,7}

- Next centroids are: m1=(1.25,1.5) and m2 = (3.9,5.1)

K-means example (K=2)

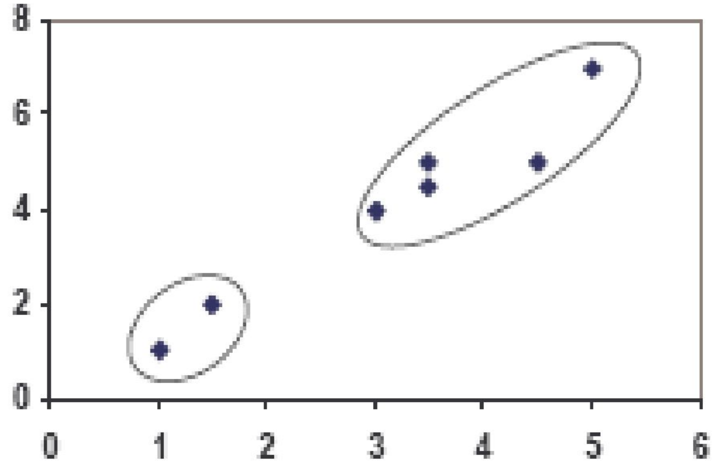
The clusters obtained are:

{1,2} and {3,4,5,6,7}

- Therefore, there is no change in the cluster.
- Thus, the algorithm comes to a halt
  - Final result consist of 2 clusters {1,2} and {3,4,5,6,7}

Individual	Centroid 1	Centroid 2
1	0.56	5.02
2	0. 56	3.92
3	3.05	1.42
4	6.66	2.20
5	4.16	0.41
6	4.78	0.61
7	3.75	0.72

K-means example (K=2)



Thank You!