Representation Learning

Representation Learning

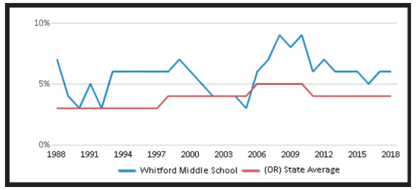
- Explicit Features
  - Text (explicit text properties)
    - Unigram, bi-gram, tri-gram, n-gram
  - Image (explicit image properties)
    - Intensity, pixel position, edges, up edges, down edges, etc.
  - Graph/Network (explicit graph properties)
    - degree, cluster coefficient, etc.

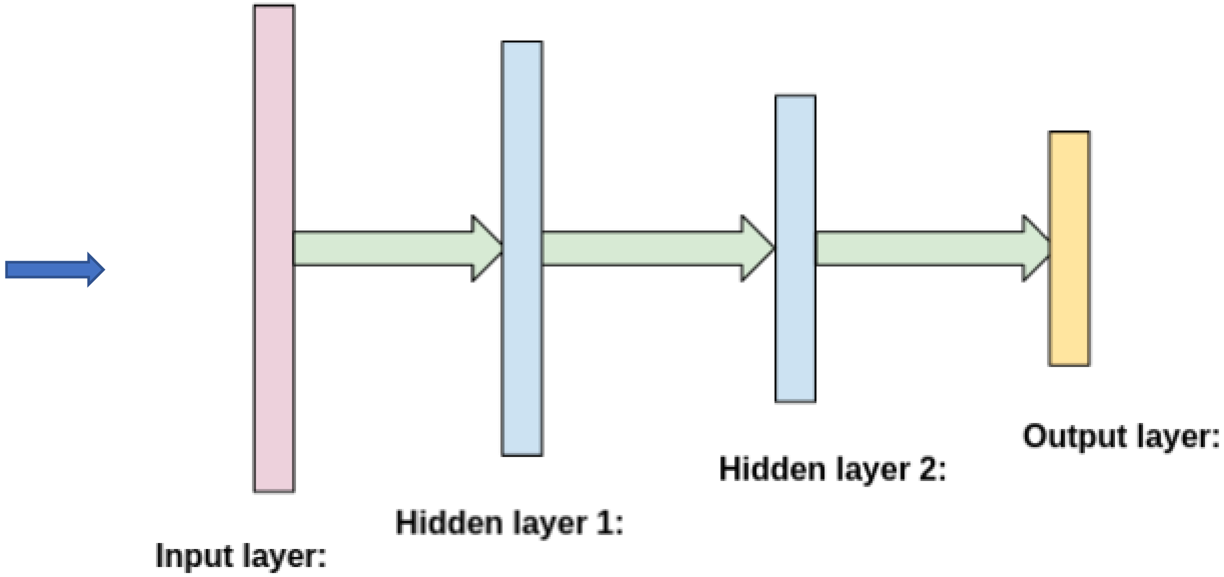
Representation Learning

- Explicit Features
  - Text (explicit text properties)
    - Unigram, bi-gram, tri-gram, n-gram
  - Image (explicit image properties)
    - Intensity, pixel position, edges, up edges, down edges, etc.
  - Graph/Network (explicit graph properties)
    - degree, cluster coefficient, etc.

- In deep learning, get rid of the explicit features
  - Learn the representation using neural models

For example


10%
5%
0%
1988 1991 1994 1997 2000 2003 2006 2009 2012 2015 2018
Whitford Middle School
(OR) State Average


Input layer:
Hidden layer 1:
Hidden layer 2:
Output layer:

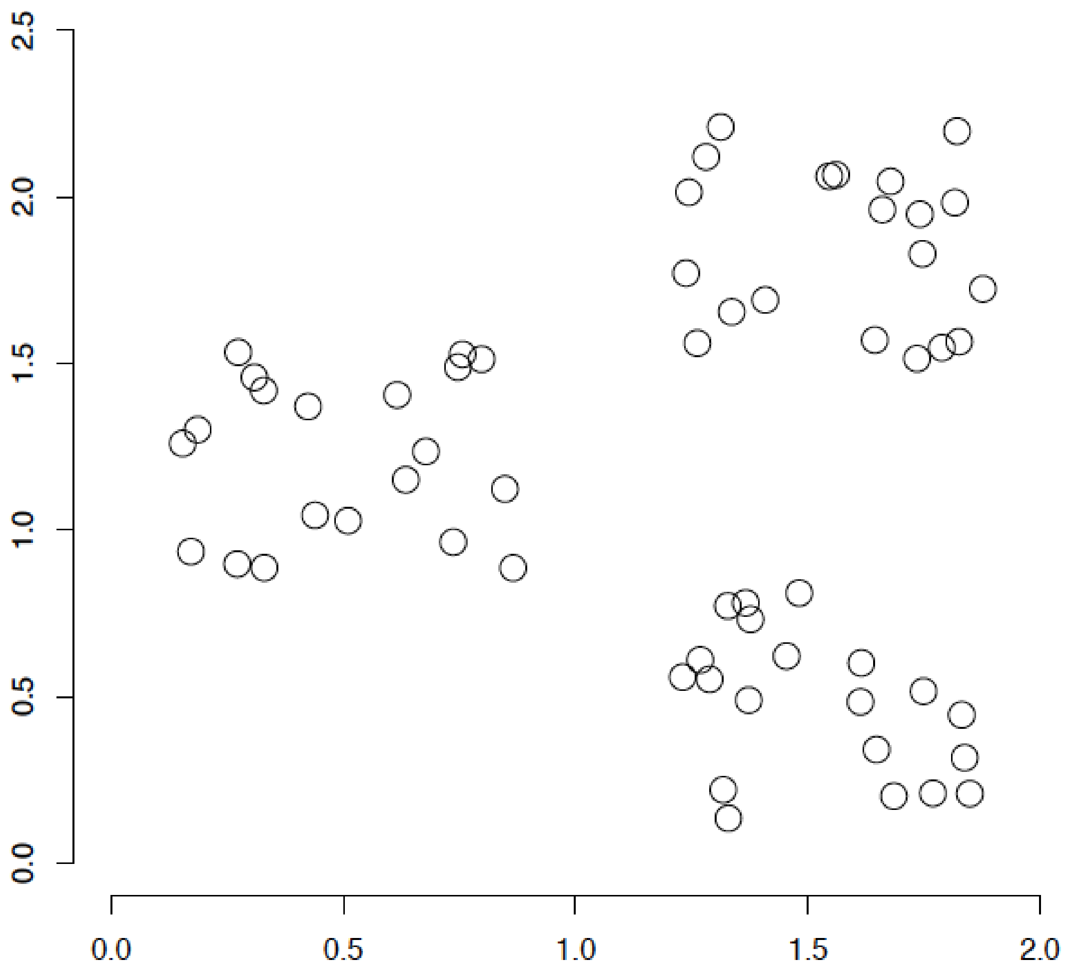
Clustering

Machine Learning Paradigms

- **Supervised Learning**
  - Learning from experience and producing output map
    - **Classification: categorical outputs**
    - **Regression: continuous output**
- **Unsupervised Learning**
  - Discovering patterns in data
    - **Clustering: grouping cohesive data points**
    - **Association: cooccurrence frequency**
- **Reinforcement Learning**
  - Learning control

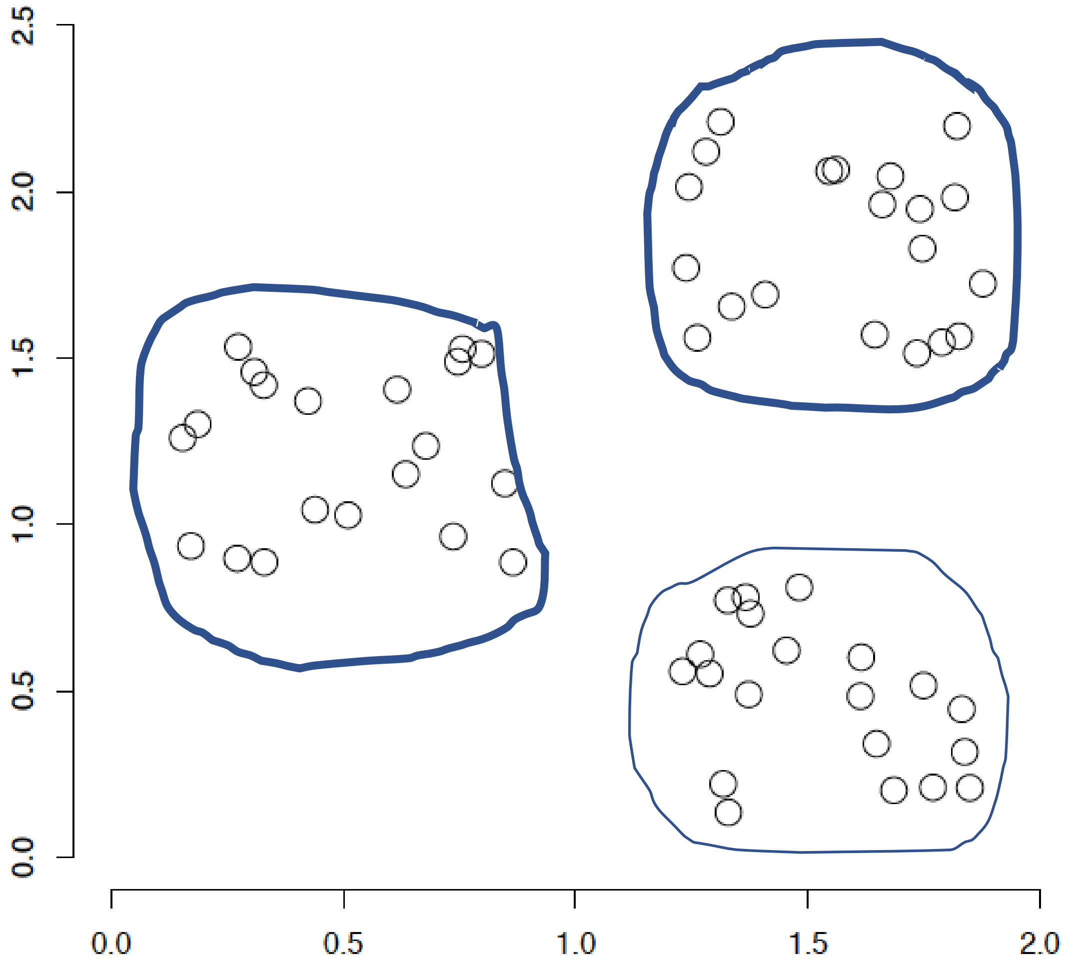
Clustering

- A way of grouping together data samples that are ***similar*** in some way - according to some criteria
- A form of ***unsupervised learning***
- It is a method of ***data exploration*** – a way of looking for patterns or structure in the data that are of interest



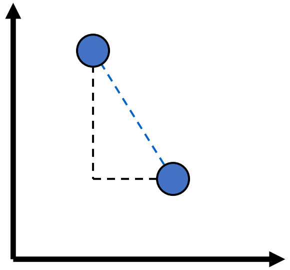
Clustering

- A way of grouping together data samples that are ***similar*** in some way - according to some criteria
- A form of ***unsupervised learning***
- It is a method of ***data exploration*** – a way of looking for patterns or structure in the data that are of interest



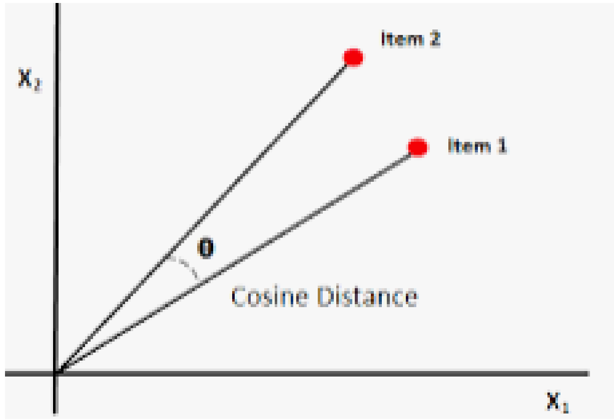
Similarity/Dissimilarity?

- Euclidean distance



$$d_{euc}(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

- Cosine Similarity



$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}},$$

Similarity/Dissimilarity?

- Pearson linear correlation

$$\rho(\mathbf{x},\mathbf{y}) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

$$d_p = \frac{1 - \rho(\mathbf{x},\mathbf{y})}{2}$$

$$\bar{x} = \frac{1}{n}\sum_{i}^{n} x_i$$

$$\bar{y} = \frac{1}{n}\sum_{i}^{n} y_i$$

Various Clustering Algorithms

- Partitioning (k-mean)
- Hierarchical (HAC)
- Self Organizing Map (SOM)
- Density Based (DBScan)

Clustering Algorithms

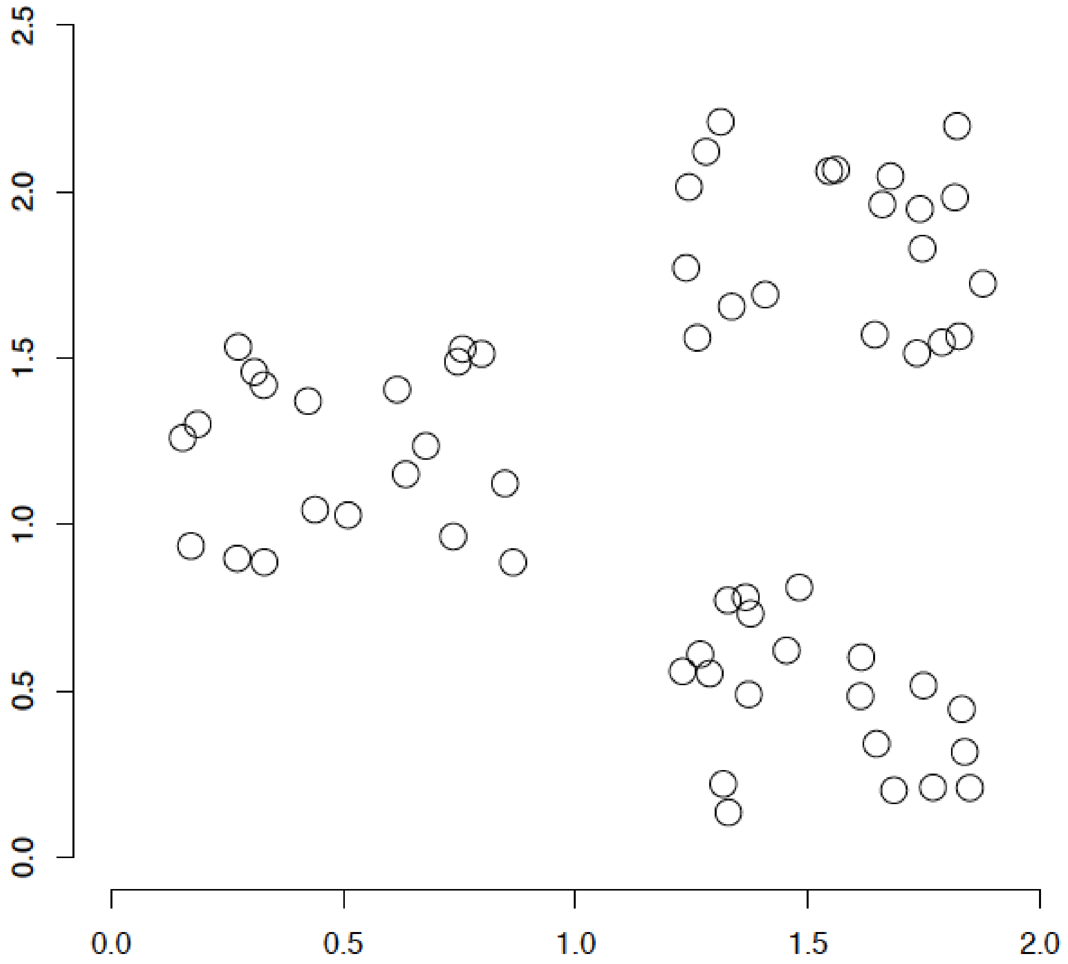
- Partitioning Based Algorithms (k-means)
- Hierarchical Algorithms
- Self Organizing Map (SOM)
- Density Based Algorithms (DBScan)

Hard and soft clustering

- Hard: No overlapping clusters
- Soft: Clusters may overlap

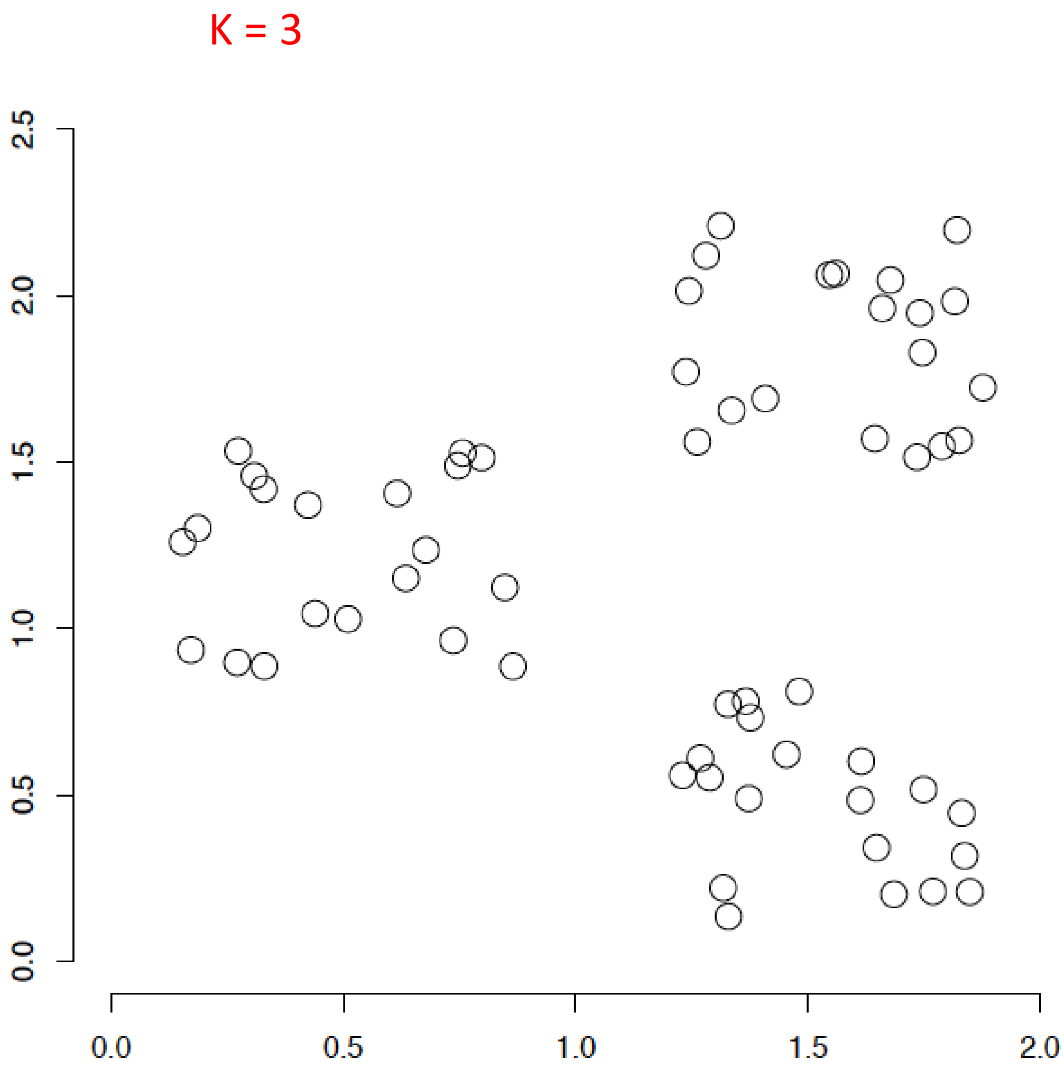
K-Means Clustering

- Choose k - the number of clusters
- Initialize cluster centers $\mu_1, \ldots \mu_k$
  - Could pick *k* data points and set cluster centers to these points
- For each data point,
  - compute distance from each k cluster centers and assign the data point to the closest cluster
- Re-compute cluster centers (mean of data points in clusters)
- Stop when there are no new re-assignments



K-Means Clustering

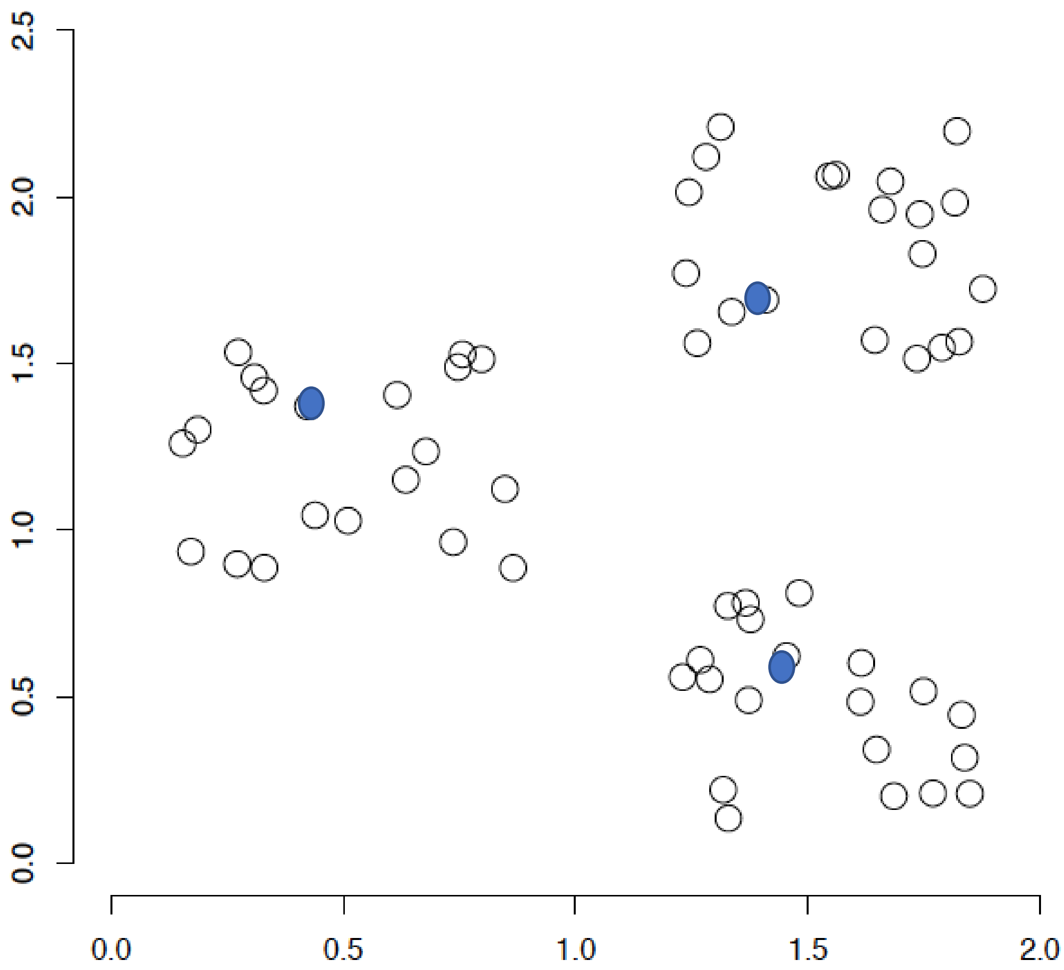
- Choose k - the number of clusters
- Initialize cluster centers $\mu_1, \dots \mu_k$
  - Could pick *k* data points and set cluster centers to these points
- For each data point,
  - compute distance from each k cluster centers and assign the data point to the closest cluster
- Re-compute cluster centers (mean of data points in clusters)
- Stop when there are no new re-assignments



K-Means Clustering

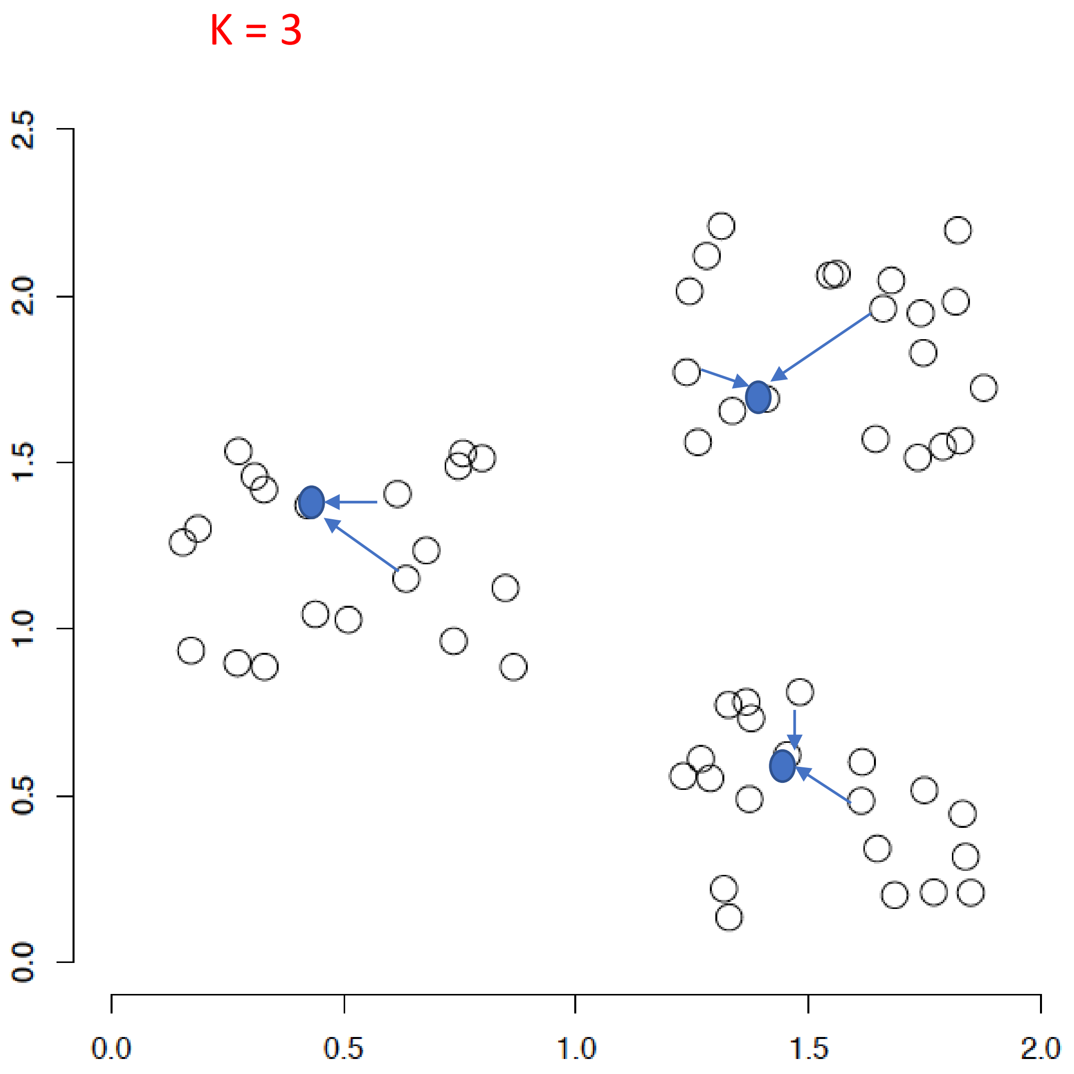
- Choose k - the number of clusters
- Initialize cluster centers $\mu_1, \ldots \mu_k$
  - Could pick *k* data points and set cluster centers to these points
- For each data point,
  - compute distance from each k cluster centers and assign the data point to the closest cluster
- Re-compute cluster centers (mean of data points in clusters)
- Stop when there are no new re-assignments

K = 3



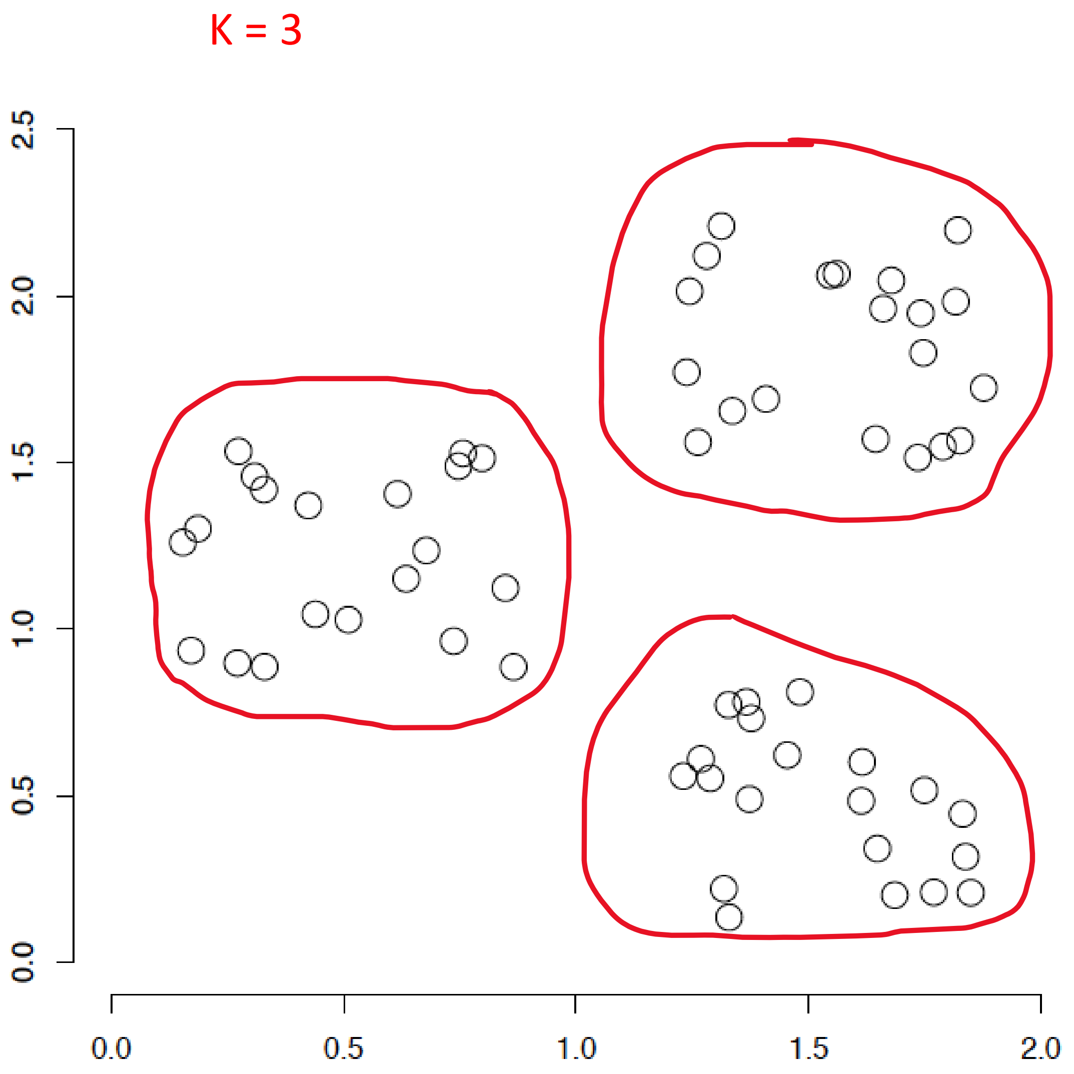
K-Means Clustering

- Choose k - the number of clusters
- Initialize cluster centers $\mu_1, \ldots \mu_k$
  - Could pick *k* data points and set cluster centers to these points
- For each data point,
  - compute distance from each k cluster centers and assign the data point to the closest cluster
- Re-compute cluster centers (mean of data points in clusters)
- Stop when there are no new re-assignments



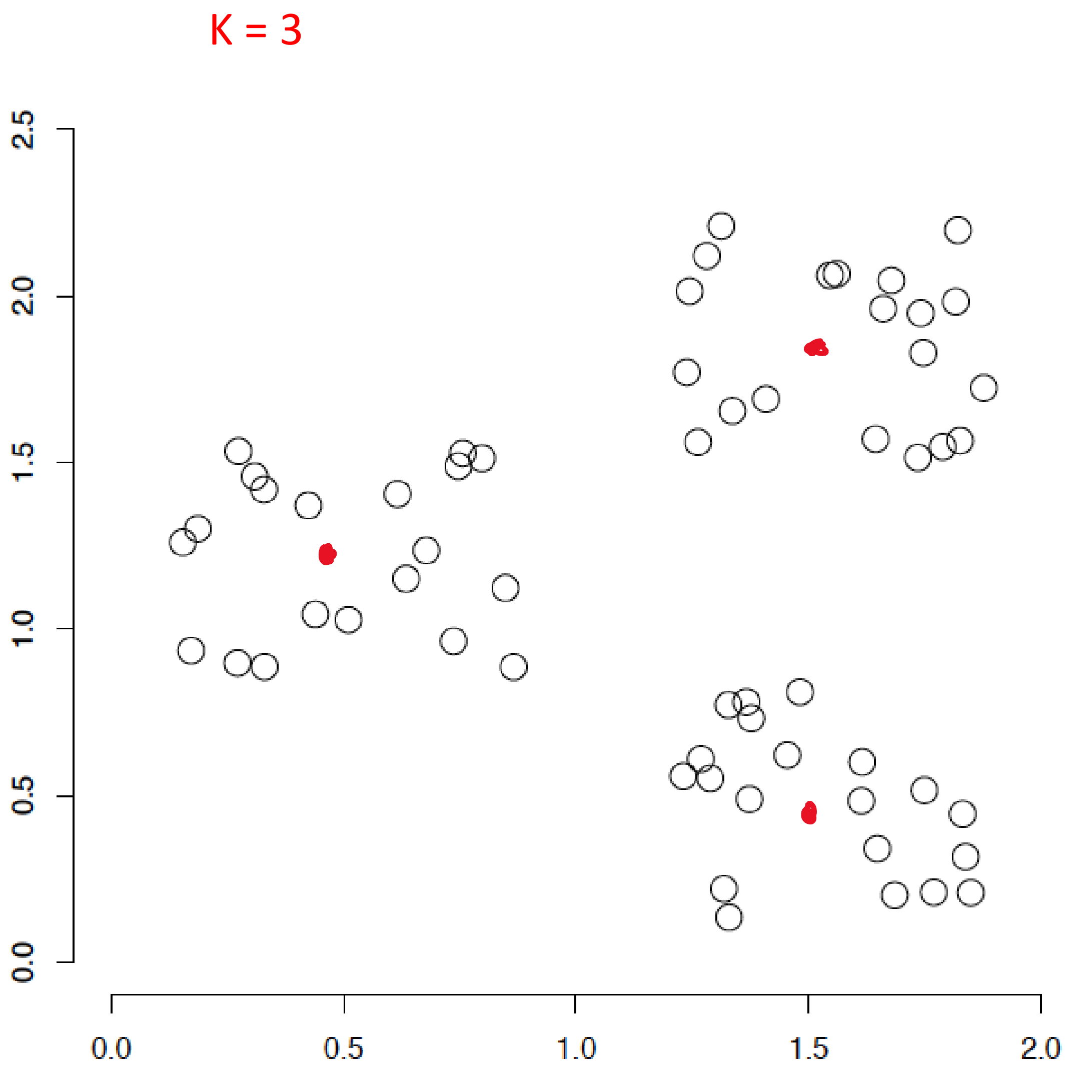
K-Means Clustering

- Choose k - the number of clusters
- Initialize cluster centers $\mu_1, \ldots \mu_k$
  - Could pick *k* data points and set cluster centers to these points
- For each data point,
  - compute distance from each k cluster centers and assign the data point to the closest cluster
- Re-compute cluster centers (mean of data points in clusters)
- Stop when there are no new re-assignments



K-Means Clustering

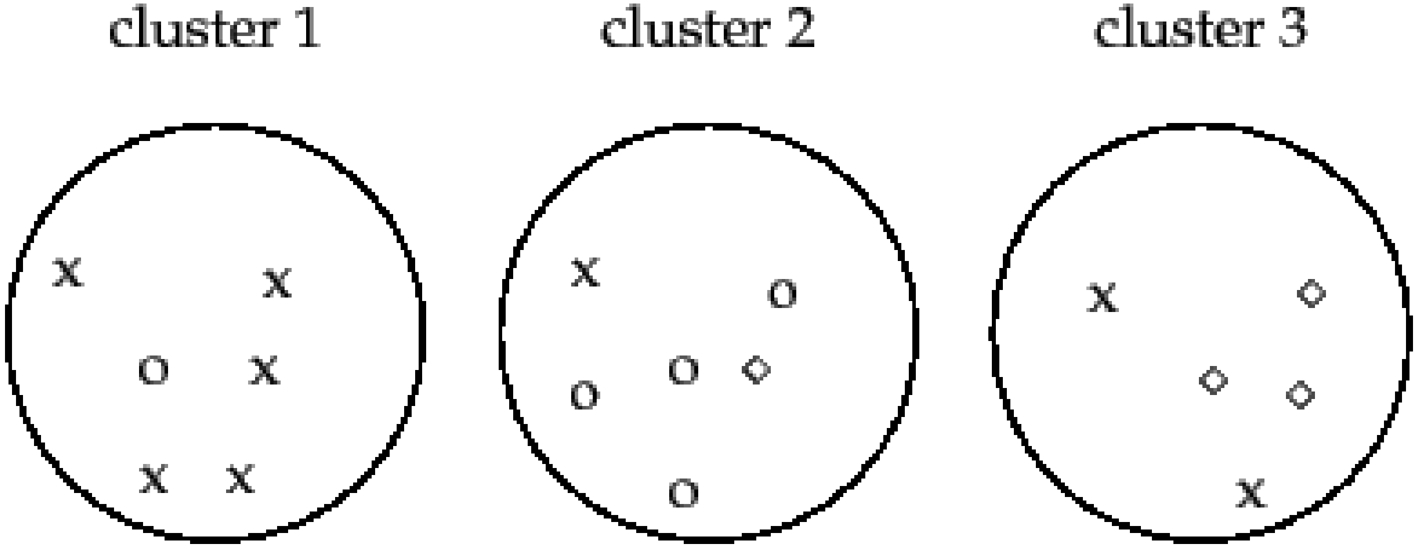
- Choose k - the number of clusters
- Initialize cluster centers $\mu_1, \ldots \mu_k$
  - Could pick *k* data points and set cluster centers to these points
- For each data point,
  - compute distance from each k cluster centers and assign the data point to the closest cluster
- Re-compute cluster centers (mean of data points in clusters)
- Repeat untill there are no new re-assignments



Evaluation Methods

- Purity
- Rand Index (RI)
- Normalised Mutual Information (NMI)

Evaluation Methods - Purity



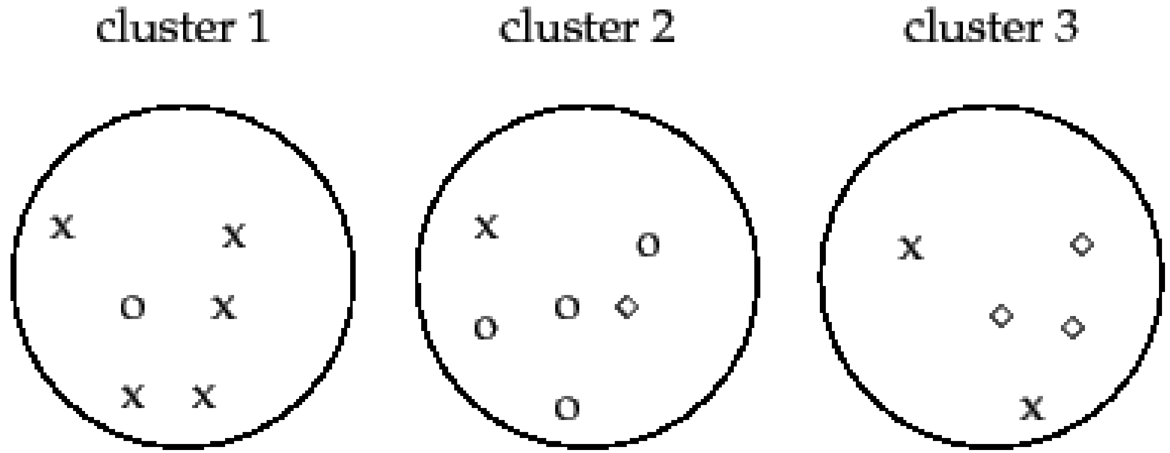
$$\text{purity}(\Omega, \mathbb{C}) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j|$$

$\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ is the set of clusters

$\mathbb{C} = \{c_1, c_2, \ldots, c_J\}$ is the set of classes

N = Number of samples

Evaluation Methods



N = Number of samples

$\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ is the set of clusters

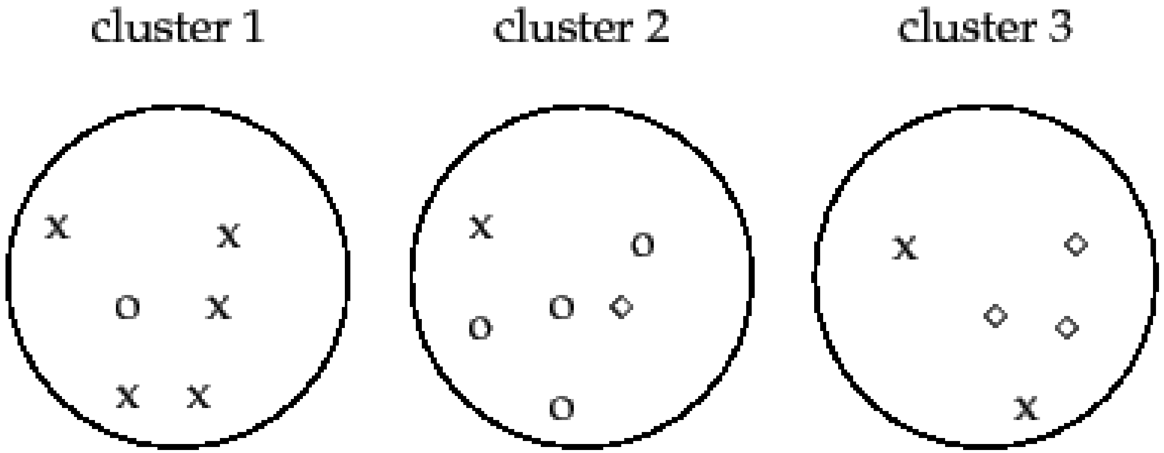
$\mathbb{C} = \{c_1, c_2, \ldots, c_J\}$ is the set of classes

$$\mathrm{NMI}(\Omega, \mathbb{C}) = \frac{I(\Omega; \mathbb{C})}{[H(\Omega) + H(\mathbb{C})]/2}$$

$$I(\Omega; \mathbb{C}) = \sum_k \sum_j P(\omega_k \cap c_j) \log \frac{P(\omega_k \cap c_j)}{P(\omega_k) P(c_j)} = \sum_k \sum_j \frac{|\omega_k \cap c_j|}{N} \log \frac{N |\omega_k \cap c_j|}{|\omega_k||c_j|}$$

$$H(\Omega) = -\sum_k P(\omega_k) \log P(\omega_k) = -\sum_k \frac{|\omega_k|}{N} \log \frac{|\omega_k|}{N}$$

Evaluation Methods



$$\mathrm{RI} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}}$$

TP: two samples belonging to same class, predicted as same cluster
TN: two samples belonging to different classes, predicted as different cluster
FP: two samples belonging to different classes, predicted as same cluster
FN: two samples belonging to same class, predicted as different cluster