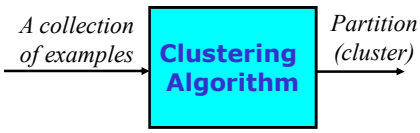
Clustering

Clustering

- Process of grouping a set of examples
- Clustering generates a partition consisting of cohesive groups or clusters from given collection of examples



- The examples to be clustered are either labelled or unlabelled
  - Algorithms which cluster labelled examples:
    - Supervised clustering
    - Classification
  - Algorithms which cluster unlabelled examples:
    - Unsupervised clustering
    - Do not rely on predefined classes
    - Learning by observation, rather than learning by examples.

2

Clustering

- Clustering is a two step process
  - Step1: Partition the collection of examples (clustering)
    - Learning by observation (training phase)
    - Group the collection of examples into finite number of clusters such that the examples that are similar to one another within the same cluster and are dissimilar to examples in other clusters
    - Obtaining cluster labels
    - Unsupervised learning: Do not rely on predefined classes and class-labelled training examples
  - Step2: Assign cluster labels to examples
    - Testing phase

3

Categorization of Clustering Methods

- Partitioning methods:
  - These methods construct K partitions of the data, where each partition represents a cluster
  - Idea: Cluster the collection of examples based on the distance between examples
  - Results in spherical shaped cluster
  1. K-means algorithm
  2. K-medoids algorithm
  3. Gaussian mixture model
- Hierarchical methods:
  - These methods create a hierarchical decomposition of the collection of examples
  - Results in spherical shaped cluster
  1. Agglomerative approach (bottom-up approach)
  2. Divisive approach (top-down approach)

4

Categorization of Clustering Methods

- Density-based methods:
  - These methods cluster collection of examples based on the notion of density
  - General idea: To continue growing the given cluster as long as density (number of examples) in the neighbourhood exceeds some threshold
  - Example:
    - DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

5

Classical Portioning Methods

- Centroid-based technique:
  - Partition the collection of examples into K clusters based on the distance between examples
  - Cluster similarity is measured in regard to the sample mean of the examples within a cluster
  - Cluster centroid or center of gravity: Sample mean value of the examples within a cluster
  - Cluster center is used to represent the cluster
  - Example: K-means algorithm
- Representative object-based technique:
  - Actual example is considered to represent the cluster
  - One representative example per cluster
  - Example: K-medoids algorithm

6

K-Means Clustering Algorithm

- Dividing the data into K groups or partitions
- Given: Training data, $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^{N}, \mathbf{x}_n \in \mathbb{R}^d$ and K
- Target: Partition the set $\mathcal{D}$ into K clusters (disjoint subsets), $\{\mathcal{D}_k\}_{k=1}^{K}$
  - Each of the clusters is associated with centers, $\boldsymbol{\mu}_k$, k=1, 2, …, K
  - Come up with the centers of clusters
  - Cluster center acts as a cluster representative
- Euclidean distance with center of a cluster can be used as a measure of dissimilarity

7

K-Means Clustering Algorithm

1. Initialize the cluster center, $\boldsymbol{\mu}_k$, k=1, 2, …, K using randomly selected K data points in $\mathcal{D}$
2. Assign each data point $\mathbf{x}_n$ to cluster center k*

$$k^* = \arg\min_k \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

3. Update $\boldsymbol{\mu}_k$, k=1, 2, …, K: Re-compute $\boldsymbol{\mu}_k$ after assigning all the data points.

$$\widehat{\boldsymbol{\mu}}_k = \frac{\sum_{\mathcal{D}_k} \mathbf{x}_n}{N_k}$$

N_k: Number of examples in cluster k

4. Repeat the steps 2 and 3 until the convergence

8

K-Means Clustering Algorithm

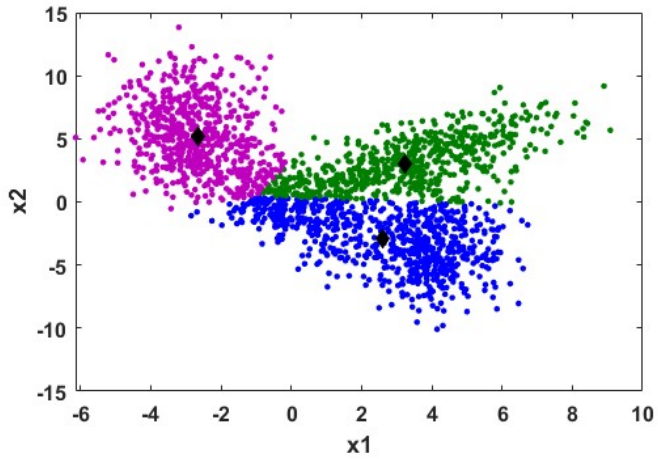
- Convergence criteria:
  - No change in the cluster assignment **OR**
  - The difference between the distortion measure (J) in the successive iteration falls below the threshold
    - Distortion measure (J) : Sum of the squares of the distance of each example to its assigned cluster center

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \left\| \mathbf{x}_n - \boldsymbol{\mu}_k \right\|^2$$

z_{nk} is 1 if $\mathbf{x}_n$ belongs to cluster k, otherwise 0

9

Illustration of K-Means Clustering



- Boundary between the cluster is linear
- Hard clustering: Each example must belong to exactly one group

10

K-Means Clustering Algorithm

- Dividing the data into K groups or partitions
- Given: Training data, $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^{N}, \mathbf{x}_n \in \mathbb{R}^d$ and K
- Target: Partition the set $\mathcal{D}$ into K clusters (disjoint subsets), $\{\mathcal{D}_k\}_{k=1}^{K}$
  - Each of the clusters is associated with centers, $\boldsymbol{\mu}_k$, k=1, 2, …, K
  - ***Better representative for a cluster***
  - Come up with the centers of clusters and covariance matrix
  - Cluster center and covariance matrix act as cluster representatives
- Mahalanobis distance with cluster representatives can be used as a measure of dissimilarity

11

K-Means Clustering Algorithm

1. Initialize the cluster center, $\boldsymbol{\mu}_k$, k=1, 2, …, K using randomly selected K data points in $\mathcal{D}$
2. Initialize the covariance matrix, $\boldsymbol{\Sigma}_k$, k=1, 2, …, K using unit matrix
3. Assign each data point $\mathbf{x}_n$ to cluster center k*

$$k^* = \arg\min_k \left(\mathbf{x}_n - \boldsymbol{\mu}_k\right)^{\mathsf{T}} \boldsymbol{\Sigma}_k^{-1} \left(\mathbf{x}_n - \boldsymbol{\mu}_k\right)$$

4. Update $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, k=1, 2, …, K: Re-compute $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ after assigning all the data points.

$$\hat{\boldsymbol{\mu}}_k = \frac{\sum_{\mathcal{D}_k} \mathbf{x}_n}{N_k} \qquad \hat{\boldsymbol{\Sigma}}_k = \frac{\sum_{\mathcal{D}_k} \left(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_k\right)\left(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_k\right)^{\mathsf{T}}}{N_k}$$

N_k: Number of examples in cluster k

5. Repeat the steps 3 and 4 until the convergence

12

K-Means Clustering Algorithm

- Convergence criteria:
  - No change in the cluster assignment **OR**
  - The difference between the distortion measure (J) in the successive iteration falls below the threshold

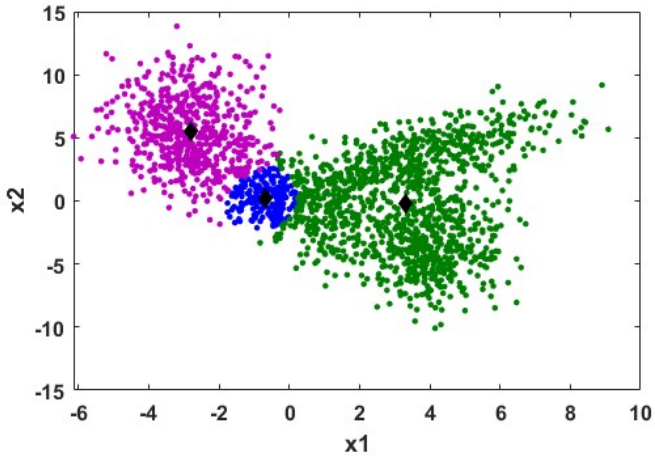
$$J = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\left[(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathsf{T}}\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k)\right]$$

z_{nk} is 1 if $\mathbf{x}_n$ belongs to cluster k, otherwise 0

- Hard clustering: Each example must belong to exactly one group

13

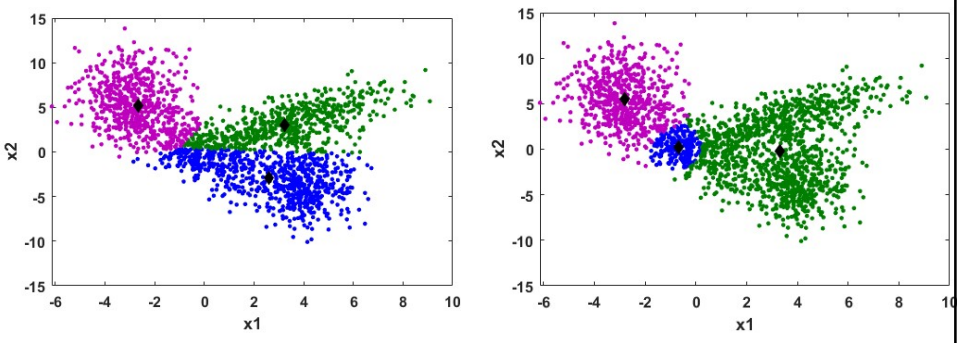
Illustration of K-Means Clustering



- Boundary between the cluster is quadratic
- Hard clustering: Each example must belong to exactly one group

14

Illustration of K-Means Clustering

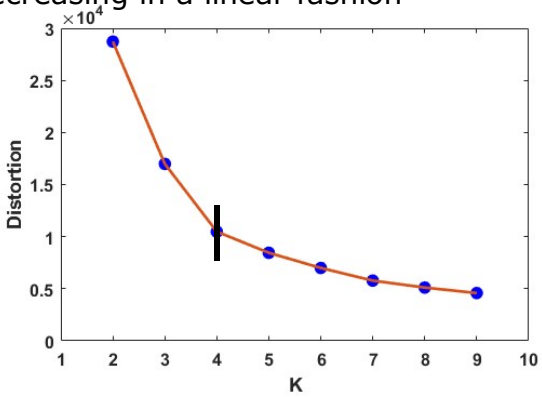


Measure of Dissimilarity: ***Euclidian Distance*** ***Mahalanobis Distance***

15

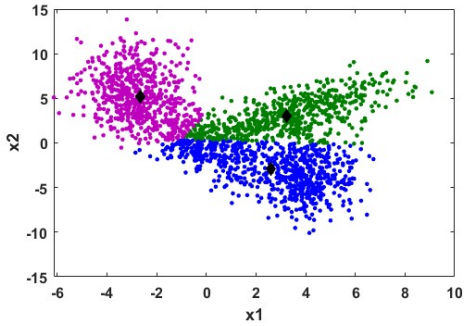
Elbow Method to Choose K

- Determine the distortion measure for different values of K
- Plot the K vs Distortion
- Optimal number of clusters: Select the value of K at the "elbow" i.e. the point after which the distortion start decreasing in a linear fashion



16

Soft Clustering



- Soft clustering: Each example belong to each group with some probability
  - Fuzzyness at the boundary of the clusters
- Gaussian mixture model (GMM) is one of the soft clustering techniques
- GMM can be seen as similar to K-means clustering
- Each cluster is represented as Gaussian density

17

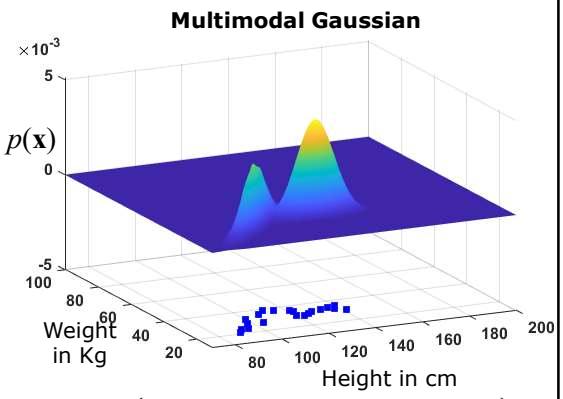
Gaussian Mixture Model (GMM)

- Given: Training data having N samples

$$\mathcal{D}=\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_N\}, \quad \mathbf{x}_n \in \mathbb{R}^d$$

- GMM is a linear superposition of multiple (K) *Gaussian components*:

$$p(\mathbf{x}) = \sum_{k=1}^{K} w_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$



$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}_k|^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^{\mathsf{T}} \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right)$$

18

Gaussian Mixture Model (GMM)

- GMM is a linear superposition of multiple Gaussians:

$$p(\mathbf{x}) = \sum_{k=1}^{K} w_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- For a d-dimensional feature vector representation of data, the parameters of GMM are
  - Mixture coefficients, w_k, $k = 1,2,\ldots, K$
    - *Mixture weight or Strength of each clusters (or mixtures or modes)*
    - Property: $\sum_{k=1}^{K} w_k = 1$
  - d-dimensional mean vector, $\boldsymbol{\mu}_k$, $k = 1,2,\ldots, K$
  - dxd size covariance matrices, $\boldsymbol{\Sigma}_k$, $k = 1,2,\ldots, K$
- Training process objective:
  - Partition the data into K groups
  - To estimate the parameters of the each cluster in GMM

19

Expectation-Maximization (EM) for GMMs

- Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters

1. Initialize the mean vectors $\boldsymbol{\mu}_k$, covariance matrices $\boldsymbol{\Sigma}_k$ and mixing coefficients w_k, and evaluate the initial value of the log likelihood
2. **E-step**: Evaluate the responsibilities $\gamma_k(\mathbf{x})$ using the current parameter values – Assign the data points to each cluster

20

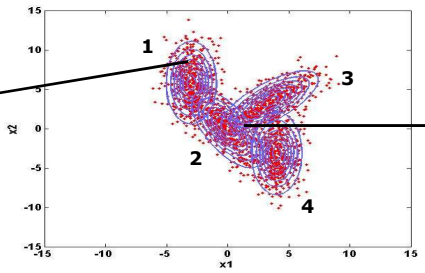
EM Method – Responsibility Term

- A quantity that plays an important role is the responsibility term, $\gamma_k(\mathbf{x})$
- It is given by

$$\gamma_k(\mathbf{x}) = \frac{w_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k=1}^{K} w_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

- w_k : mixture coefficient or prior probability of cluster k,
- $\gamma_k(\mathbf{x})$ gives the posterior probability of the cluster k for the observation $\mathbf{x}$

$\gamma_1(\mathbf{x}_n) = 0.99$
$\gamma_2(\mathbf{x}_n) = 0.01$
$\gamma_3(\mathbf{x}_n) = 0.00$
$\gamma_4(\mathbf{x}_n) = 0.00$



$\gamma_1(\mathbf{x}_m) = 0.08$
$\gamma_2(\mathbf{x}_m) = 0.42$
$\gamma_3(\mathbf{x}_m) = 0.34$
$\gamma_4(\mathbf{x}_m) = 0.16$

21

Expectation-Maximization (EM) for GMMs

- Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters
  1. Initialize the mean vectors $\boldsymbol{\mu}_k$, covariance matrices $\boldsymbol{\Sigma}_k$ and mixing coefficients w_k, and evaluate the initial value of the log likelihood
  2. **E-step**: Evaluate the responsibilities $\gamma_k(\mathbf{x})$ using the current parameter values
  3. **M-step**: Re-estimate the parameters $\boldsymbol{\mu}_k^{new}$, $\boldsymbol{\Sigma}_k^{new}$ and w_k^{new} using the current responsibilities

$$\boldsymbol{\mu}_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k(\mathbf{x}_n)\, \mathbf{x}_n \qquad \boldsymbol{\Sigma}_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k(\mathbf{x}_n)\, (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathsf{T}}$$

$$w_k^{new} = \frac{N_k}{N} \qquad N_k = \sum_{n=1}^{N} \gamma_k(\mathbf{x}_n)$$

- N_k: Effective number of points assigned to the cluster k

22

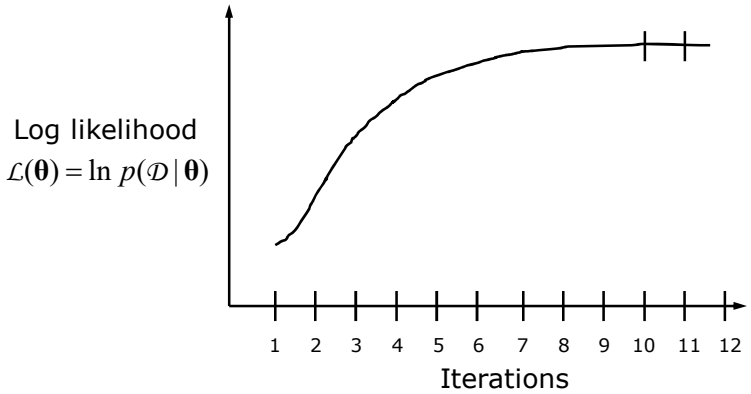
Expectation-Maximization (EM) for GMMs

- Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters
  1. Initialize the mean vectors $\boldsymbol{\mu}_k$, covariance matrices $\boldsymbol{\Sigma}_k$ and mixing coefficients w_k, and evaluate the initial value of the log likelihood
  2. **E-step**: Evaluate the responsibilities $\gamma_k(\mathbf{x})$ using the current parameter values
  3. **M-step**: Re-estimate the parameters $\boldsymbol{\mu}_k^{new}$, $\boldsymbol{\Sigma}_k^{new}$ and w_k^{new} using the current responsibilities
  4. Evaluate the log likelihood and check for convergence of the log likelihood
     - If the convergence criterion is not satisfied return to step 2

23

Expectation-Maximization (EM) for GMMs

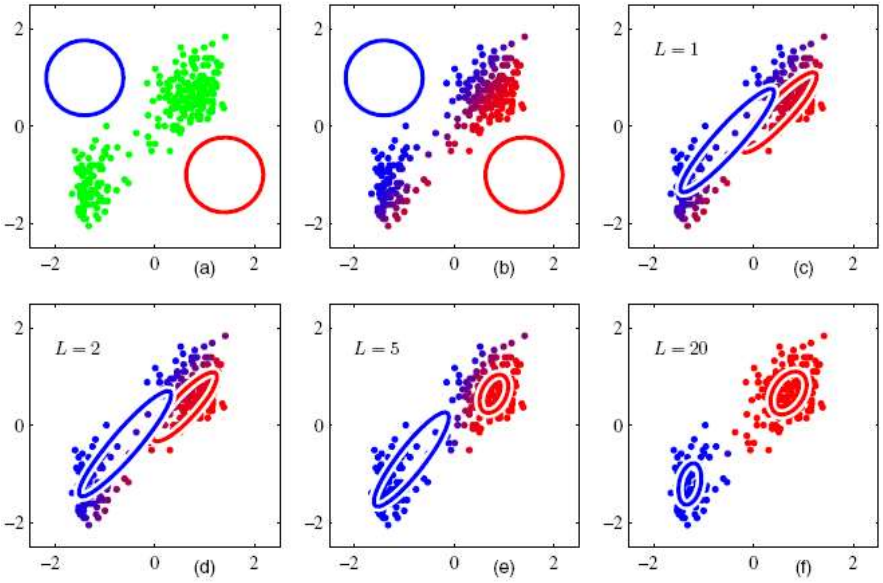
- Convergence criterion: Difference between log likelihoods of successive iterations fall below a threshold (E.g. 10^{-3})



$$\mathcal{L}(\boldsymbol{\theta}) = \ln p(\mathcal{D}\,|\,\boldsymbol{\theta}) = \sum_{n=1}^{N} \ln\left(\sum_{k=1}^{K} w_k \mathcal{N}(\mathbf{x}\,|\,\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right)$$

24

Illustration of Parameter Estimation

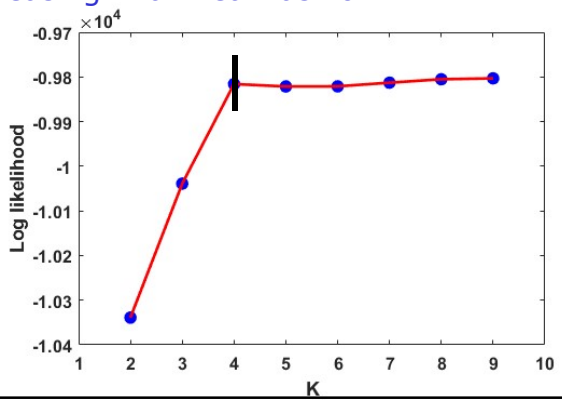


C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

25

Elbow Method to Choose K

- Determine the total data log likelihood for different values of K
- Plot the K vs log likelihood
- Optimal number of clusters: Select the value of K at the "elbow" i.e. the point after which the log likelihood start increasing in a linear fashion



26

K-Medoid Clustering Algorithms

- Related to K-means clustering
- The K-means algorithm is sensitive to outliers because an example with extremely large value may substantially distort the distribution of data
- Solution: One of the data points is chosen as representative of cluster, instead of mean value of the cluster
- Its replaces the means of cluster with modes
- Partitioning around medoids
- A medoid of a finite dataset: The data point from the set, whose average dissimilarity (distance) to all the points is minimal
  - The most centrally located point in the set

27

K-Medoid Clustering Algorithm

- Given: Training data, $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^{N}, \mathbf{x}_n \in \mathbb{R}^d$ and K

1. Initialize the medoid, $\hat{\mathbf{x}}_k$, k=1, 2, …, K using randomly selected K data points in $\mathcal{D}$
2. Assign each data point $\mathbf{x}_n$ to the closest medoid

$$k^* = \arg\min_k \left\|\mathbf{x}_n - \hat{\mathbf{x}}_k\right\|^2$$

3. Update medoids $\hat{\mathbf{x}}_k$, k=1, 2, …, K
   - For each data point $\mathbf{x}_n$ assigned to a cluster k compute the average dissimilarity (distance) of $\mathbf{x}_n$ to all the data points assigned to cluster k

     Average dissimilarity for $\mathbf{x}_n = \dfrac{\sum_{\mathbf{x}_m \in \mathcal{D}_k} \left\|\mathbf{x}_n - \mathbf{x}_m\right\|^2}{N_k}$ N_k: Number of examples in cluster k
   - Select the example with minimum average dissimilarity as *medoid*
4. Repeat the steps 2 and 3 until the convergence

28

K-Medoid Clustering Algorithm

- Convergence criteria:
  - No change in the cluster assignment **OR**
  - The difference between the distortion measure (absolute-error) (J) in the successive iteration falls below the threshold
    - Distortion measure (J) : Sum of the squares of the distance of each example to its corresponding reference point (medoid)

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \left\| \mathbf{x}_n - \hat{\mathbf{x}}_k \right\|^2$$

z_{nk} is 1 if $\mathbf{x}_n$ belongs to cluster k, otherwise 0

- Optimal number of clusters (k) can obtained using elbow method

29

Evaluation of Clustering: Purity Score

- Lets us assume that class index for each example is given
- Purity score: Purity is a measure of the extent to which clusters contain a single class
  - For each cluster, count the number of data points from the most common class in said cluster
  - Take the sum over all clusters and divide by the total number of data points
- Let M be the number of classes, $C_1, C_2, \ldots, C_m, \ldots, C_M$
- Let K be the number of clusters, $k = 1,2,\ldots, K$
- Let N be the number of data points

30

Evaluation of Clustering: Purity Score

- For each cluster k,
  - Count the number of data points from each class
  - Consider the number of data points of most common class

$$\max_m \left|N_k \cap C_m\right|$$

$\left|N_k \cap C_m\right|$ is the number of data points in k^{th} cluster belonging to class m

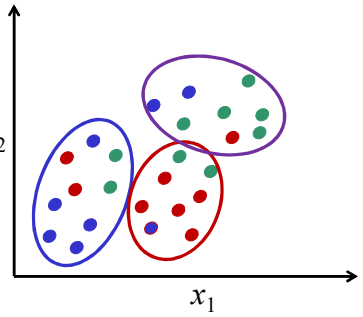
- Take the sum over all clusters, k
- Divide by the total number of data points (N)

$$\text{Purity Score} = \frac{1}{N}\sum_{k=1}^{K}\max_m \left|N_k \cap C_m\right|$$

31

Illustration of Computing Purity Score

- Number of data points, N = 25
- number of classes, M = 3
- number of clusters, K = 3
- *Cluster 1*: Number of examples of Blue Class are more, i.e. **5**
- *Cluster 2*: Number of examples of Red Class are more, i.e. **5**
- *Cluster 3*: Number of examples of Green Class are more, i.e. **5**
- **Purity score: (5+5+5)/25 = 0.60**



32

Text Books

1. J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Third Edition, Morgan Kaufmann Publishers, 2011.

2. S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, 2009.

33