S2-20_DSECLZC415: Data Mining

(Lecture #13 – Cluster Analysis)

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

DISCLAIMERS

- *The slides presented here are obtained from the authors of the books and from various other contributors. I hereby acknowledge all the contributors for their material and inputs.*
- *I have added and modified a few slides to suit the requirements of the course.*

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

Data Mining

Cluster Analysis

Types of Clusterings

Partitional Clustering

- A division of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- Divisions can be
  - Distance based
  - Density based

Hierarchical clustering

- A set of nested clusters organized as a hierarchical tree

Comparison of DBSCAN and K-means

Both are partitional.

K-means is complete; DBSCAN is not.

K-means has a prototype-based notion of a cluster; DBSCAN uses a density-based notion.

K-means can find clusters that are not well-separated. DBSCAN will merge clusters that touch.

DBSCAN handles clusters of different shapes and sizes; K-means prefers globular clusters.

DBSCAN can handle noise and outliers; K-means performs poorly in the presence of outliers

Comparison of DBSCAN and K-means

K-means can only be applied to data for which a centroid is meaningful; DBSCAN requires a meaningful definition of density

Both techniques were designed for Euclidean data, but extended to other types of data

K-means has an O(n) time complexity; DBSCAN is O(n^2)

Because of random initialization, the clusters found by K-means can vary from one run to another; DBSCAN always produces the same clusters

DBSCAN automatically determines the number of clusters; K-means does not

K-means has only one parameter, DBSCAN has two.

innovate achieve lead

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

Extensions to Hierarchical Clustering

Limitations to Hierarchical Clustering

- Major weakness of agglomerative clustering methods
  - Can never undo what was done previously
  - Do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects
- Integration of hierarchical & distance-based clustering
  - BIRCH : uses CF-tree and incrementally adjusts the quality of sub-clusters

BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies)

- Zhang, Ramakrishnan & Livny, SIGMOD'96
- Incrementally construct a CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering
  - Phase 1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)
  - Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree
- *Scales linearly*: finds a good clustering with a single scan and improves the quality with a few additional scans
- *Weakness:* handles only numeric data, and sensitive to the order of the data record

Clustering Feature Vector in BIRCH

Clustering Feature (CF): ***CF* = *(N, LS, SS)***

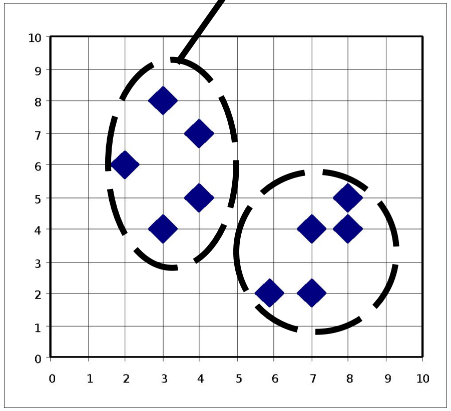
N: **Number of data points**

LS: linear sum of N points: $\sum_{i=1}^{N} X_i$

SS: square sum of N points

$$\sum_{i=1}^{N} X_i^{\,2}$$

CF = (5, (16,30),(54,190))



(3,4)
(2,6)
(4,5)
(4,7)
(3,8)

CF-Tree in BIRCH

Clustering feature:

- Summary of the statistics for a given subcluster: the 0-th, 1st, and 2nd moments of the subcluster from the statistical point of view
- Registers crucial measurements for computing cluster and utilizes storage efficiently

A CF tree is a height-balanced tree that stores the clustering features for a hierarchical clustering

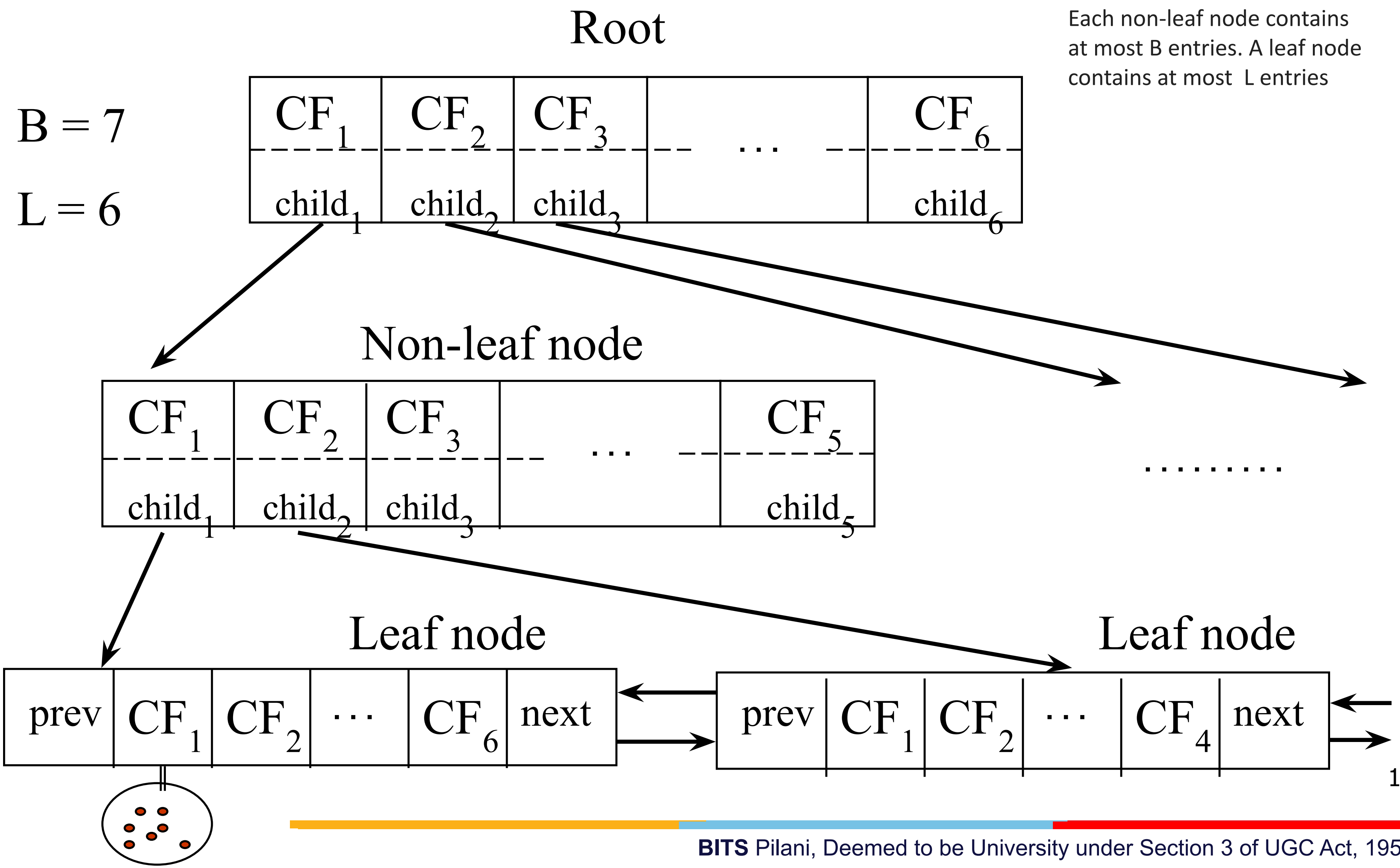
- A nonleaf node in a tree has descendants or "children"
- The nonleaf nodes store sums of the CFs of their children

A CF tree has two parameters

- Branching factor: max # of children
- Threshold: max diameter of sub-clusters stored at the leaf nodes

The CF Tree Structure

Each non-leaf node contains at most B entries. A leaf node contains at most L entries


Root
B = 7
L = 6
CF_1 CF_2 CF_3 … CF_6
$child_1$ $child_2$ $child_3$ $child_6$
Non-leaf node
CF_1 CF_2 CF_3 … CF_5
$child_1$ $child_2$ $child_3$ $child_5$
……….
Leaf node
prev CF_1 CF_2 … CF_6 next
Leaf node
prev CF_1 CF_2 … CF_4 next

BIRCH Steps

The first phase builds a CF tree out of the data points, a height-balanced tree data structure

In the second step, the algorithm scans all the leaf entries in the initial CF tree to rebuild a smaller CF tree, while removing outliers and grouping crowded subclusters into larger ones.

In step three an existing clustering algorithm is used to cluster all leaf entries. Here algorithm is applied directly to the subclusters represented by their CF vectors.

In (optional) step 4 the centroids of the clusters produced in step 3 are used as seeds and redistribute the data points to its closest seeds to obtain a new set of clusters

The Birch Algorithm

Cluster Diameter

$$\sqrt{\frac{1}{n(n-1)}\sum(x_i - x_j)^2}$$

In first phase, For each point in the input

- Find closest leaf entry
- Add point to leaf entry and update CF
- If entry diameter > max_diameter, then split leaf, and possibly parents

Algorithm is O(n)

Concerns

- Sensitive to insertion order of data points
- Since we fix the size of leaf nodes, so clusters may not be so natural
- Clusters tend to be spherical given the radius and diameter measures

Centroid, Radius and Diameter of a Cluster

- Centroid: the “middle” of a cluster

$$C = \frac{\Sigma_{i=1}^{N}(x_i)}{N} = \frac{LS}{n}$$

- Radius: square root of average distance from any point of the cluster to its centroid

$$R = \sqrt{\frac{\Sigma_{i=1}^{N}(x_i - c\ \ )^2}{N}} \qquad R = \sqrt{\frac{nSS - LS^2}{n^2}}$$

- Diameter: square root of average mean squared distance between all pairs of points in the cluster

$$D = \sqrt{\frac{\Sigma_{i=1}^{N}\Sigma_{j=1}^{N}(x_i - x_j)^2}{N(N-1)}} = \sqrt{\frac{2nSS - 2LS^2}{n(n-1)}}$$

innovate achieve lead

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

OPTICS

OPTICS: A Cluster-Ordering Method (1999)

OPTICS: Ordering Points To Identify the Clustering Structure

- Ankerst, Breunig, Kriegel, and Sander (SIGMOD'99)
- Produces a special order of the database wrt its density-based clustering structure
- This cluster-ordering contains info equivalent to the density-based clusterings corresponding to a broad range of parameter settings
- Good for both automatic and interactive cluster analysis, including finding intrinsic clustering structure
- Can be represented graphically or using visualization techniques

OPTICS algorithm

Ordering points to identify the clustering structure (OPTICS) is an algorithm for finding density-based clusters

OPTICS overcomes the (DBSCAN) problem of detecting meaningful clusters in data of varying density

Points of the dataset are (linearly) ordered such that spatially closest points become neighbors. A special distance is plotted for each point

OPTICS: Some Extension from DBSCAN

Complexity: O(*NlogN*)

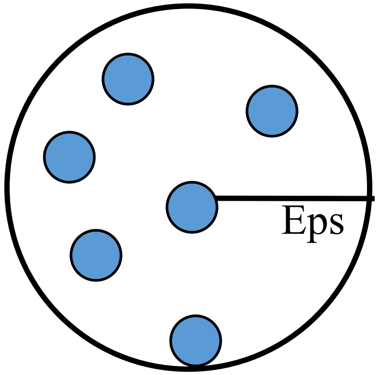
Core Distance:

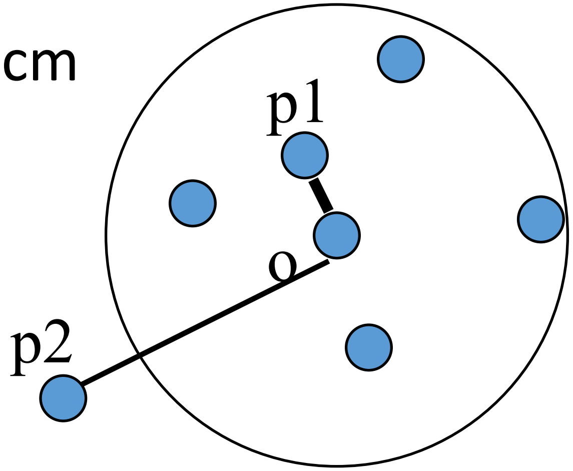
- min eps s.t. point is core

Reachability Distance

Max (core-distance (o), d(o, p))

r(p1, o) = 3cm. r(p2,o) = 4cm

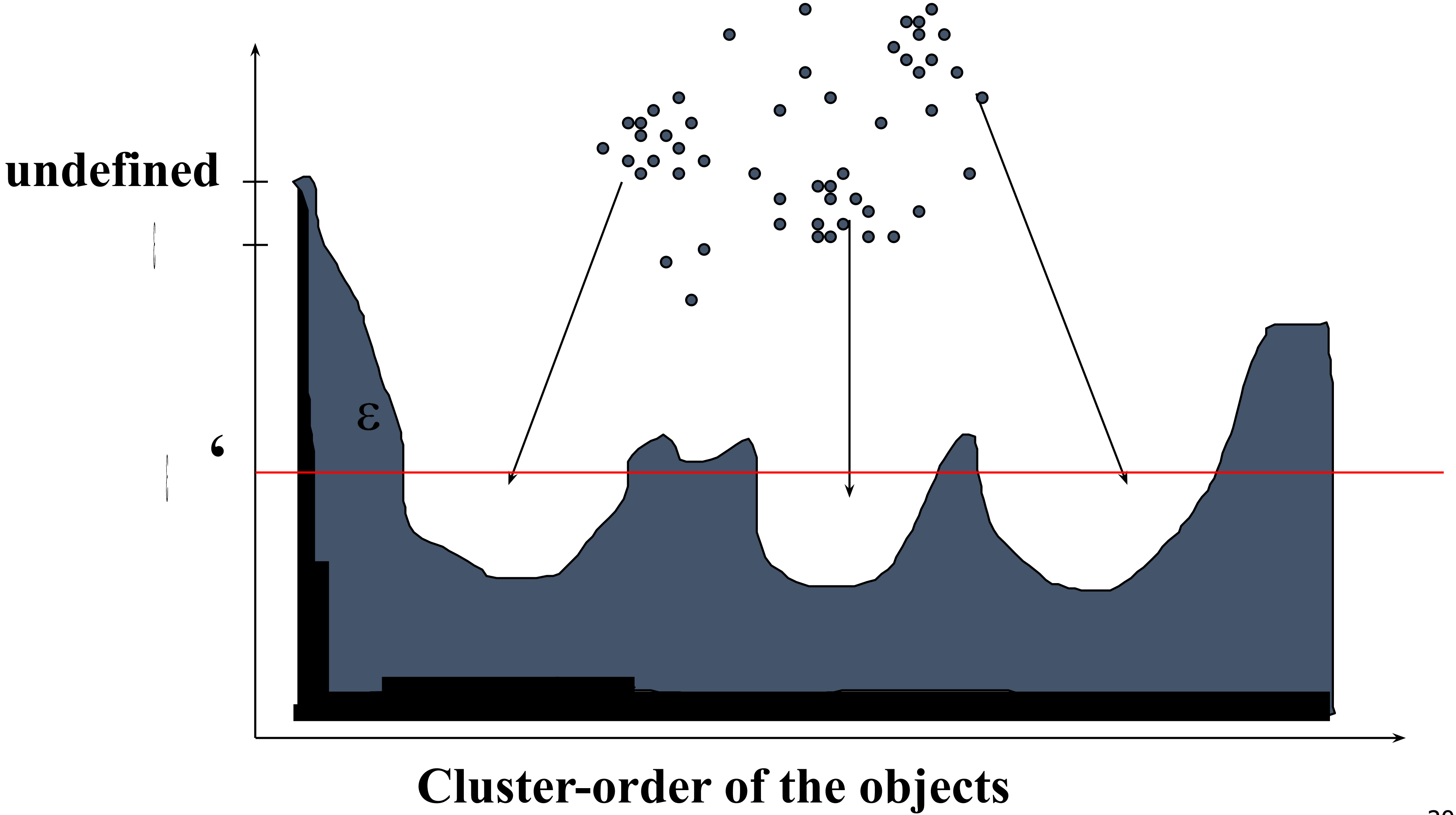




MinPts = 5

= 3 cm

Reachability-distance



innovate achieve lead

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

Cluster Validation

Cluster Validity

For supervised classification we have a variety of measures to evaluate how good our model is

- Accuracy, precision, recall

For cluster analysis, the analogous question is how to evaluate the "goodness" of the resulting clusters?

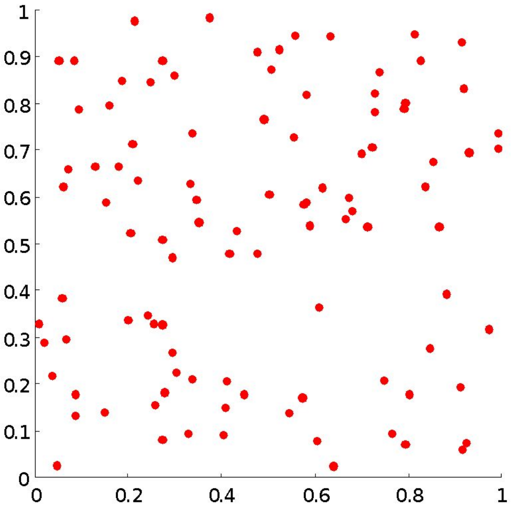
But "clusters are in the eye of the beholder"!

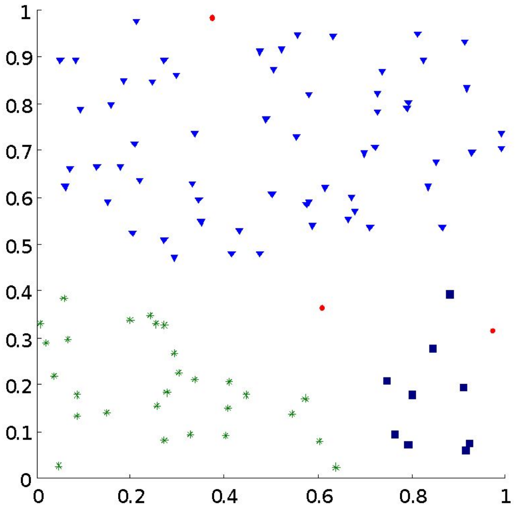
Then why do we want to evaluate them?

- To avoid finding patterns in noise
- To compare clustering algorithms
- To compare two sets of clusters
- To compare two clusters

Clusters found in Random Data

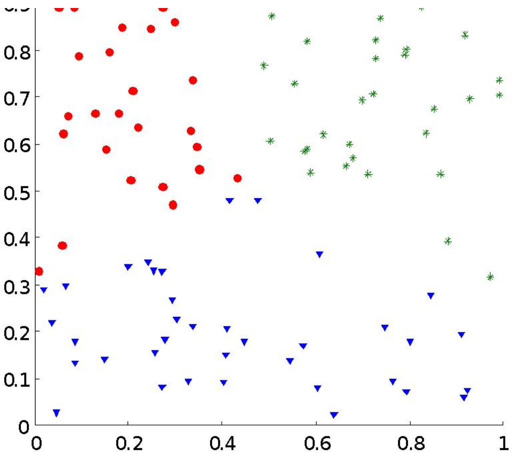
Random Points

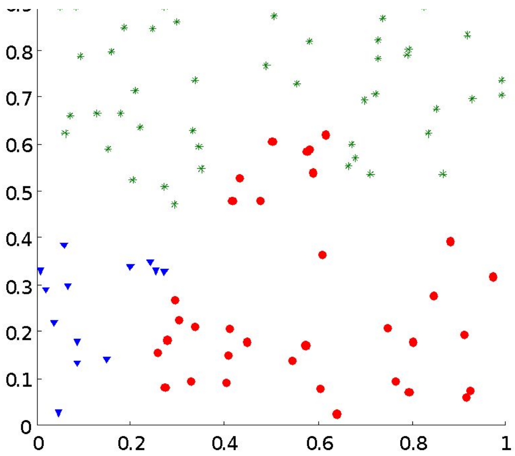




DBSCAN

K-means





Complete Link

Different Aspects of Cluster Validation

1. Determining the clustering tendency of a set of data, i.e., distinguishing whether non-random structure actually exists in the data.
2. Comparing the results of a cluster analysis to externally known results, e.g., to externally given class labels.
3. Evaluating how well the results of a cluster analysis fit the data *without* reference to external information.
   - Use only the data
4. Comparing the results of two different sets of cluster analyses to determine which is better.
5. Determining the ‘correct’ number of clusters.

For 2, 3, and 4, we can further distinguish whether we want to evaluate the entire clustering or just individual clusters.

Clustering Tendency

All clustering algorithms find some clusters, whether data has natural clusters or not.

We need a mechanism to check if at least some clusters are of good quality

Alternatively, we can directly check the data for clustering tendency. A common approach is to use statistical tests for spatial randomness (in Euclidean space) among data points.

Hopkins Statistic : Use p points from data and generate a set of p random points in data space. Let u_i be nearest neighbor distances in generated data and w_i be nearest neighbor distances in supplied data. Hopkin statistic H is defined as

$$H = \frac{\sum_{i=1}^{p} w_i}{\sum_{i=1}^{p} u_i + \sum_{i=1}^{p} w_i}$$

H is closer to 0 => data is highly clustered;
H is closer to 0.5 => data is uniformly distributed in data space

Evaluation measures for Cluster Validity

Unsupervised Measures without resort to external information, also referred *internal indices* e.g. SSE.

Measures can be

- Cluster Cohesion (compactness, tightness) - how closely related the objects in a cluster are?
- Cluster Separation (isolation) - how distinct or well-separated a cluster is from other clusters

Supervised Measures match the clustering output to some external structure, also referred *external indices* e.g. entropy

- Measures how well cluster labels match externally supplied class labels.

Relative - compare different clustering outcomes. can be based on supervised or unsupervised measures, e.g. two K-means clustering outputs can be compared using either SSE or entropy.

Internal Measures: SSE

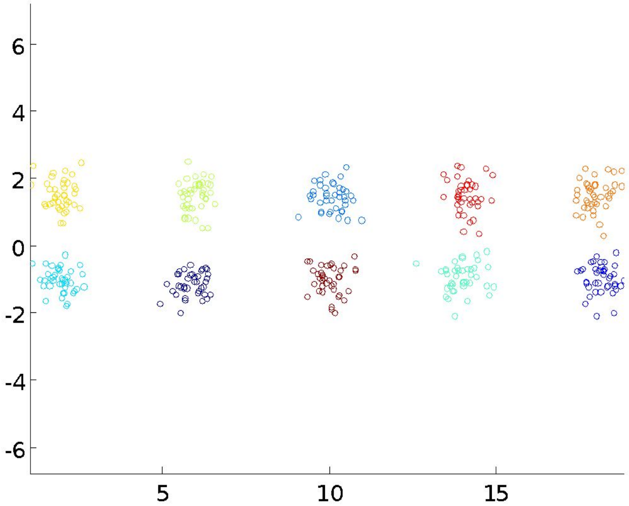
Clusters in more complicated figures aren't well separated

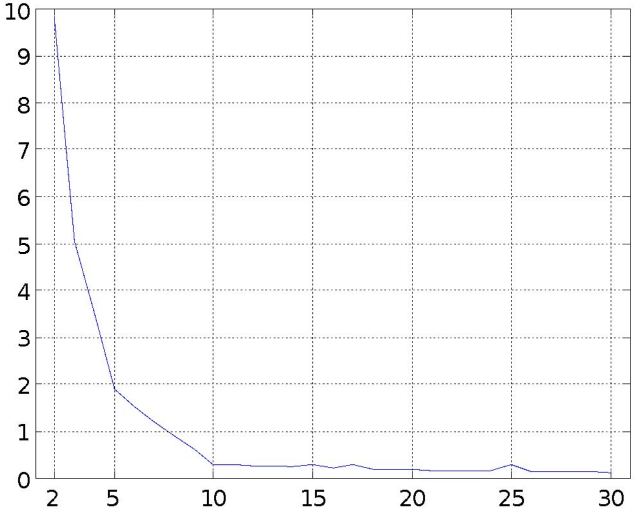
Internal Index: Used to measure the goodness of a clustering structure without respect to external information

- SSE

SSE is good for comparing two clusterings or two clusters (average SSE).

Can also be used to estimate the number of clusters





Internal Measures: Cohesion and Separation

Cluster Cohesion: Measures how closely related are objects in a cluster

- Example: SSE

Cluster Separation: Measure how distinct or well-separated a cluster is from other clusters

- Example: Squared Error
- Cohesion is measured by the within cluster sum of squares (SSE)

$$WSS = \sum_i \sum_{x \in C_i} (x - c_i)^2$$

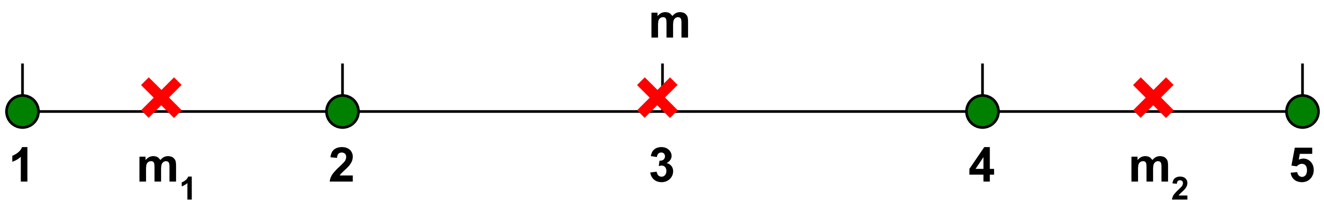
- Separation is measured by the between cluster sum of squares

$$BSS = \sum_i |C_i| (c - c_i)^2$$

- Where $|C_i|$ is the size of cluster i

Internal Measures: Cohesion and Separation

- Example: SSE
  - BSS + WSS = constant



K=1 cluster:

$$SSE = WSS = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$$

$$BSS = 4 \times (3-3)^2 = 0$$

$$Total = 10 + 0 = 10$$

K=2 clusters:

$$SSE = WSS = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$

$$BSS = 2 \times (3-1.5)^2 + 2 \times (4.5-3)^2 = 9$$

$$Total = 1 + 9 = 10$$

Measuring Clustering Quality: Extrinsic Methods

Clustering quality measure: $Q(C, C_g)$, for a clustering C given the ground truth C_g.

Q is good if it satisfies the following **4** essential criteria

- Cluster homogeneity: the purer, the better
- Cluster completeness: should assign objects belong to the same category in the ground truth to the same cluster
- Rag bag: putting a heterogeneous object into a pure cluster should be penalized more than putting it into a *rag bag* (i.e., "miscellaneous" or "other" category)
- Small cluster preservation: splitting a small category into pieces is more harmful than splitting a large category into pieces

Measuring Clustering Quality

Two methods: extrinsic vs. intrinsic

Extrinsic: supervised, i.e., the ground truth is available

- Compare a clustering against the ground truth using certain clustering quality measure
- Ex. precision and recall metrics (averaged over all classes)

Intrinsic: unsupervised, i.e., the ground truth is unavailable

- Evaluate the goodness of a clustering by considering how well the clusters are separated, and how compact the clusters are
- Ex. Silhouette coefficient

Classification-based Measures of Cluster Validity

Precision: The fraction of a cluster that consists of objects of a specified class. The precision of cluster i with respect to a class j is precision(i,j) = p_{ij}

Recall: The extent to which a cluster contains all objects of a specified class. The recall of cluster i with respect to class j is recall(i, j) = m_{ij}/m_j where m_j is the number of objects in class j.

F-measure: A combination of both precision and recall that measures the extent to which a cluster contains only objects of a particular class and all objects of that class. The F-measure of cluster i with respect to class j is

F(i,j) = (2*precision(i,j)*recall(i,j))/(precision(i,j)+recall(i,j))

External Measures of Cluster Validity: Precision/Recall

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Totals
1	3	5	40	506	96	27	677
2	4	7	280	29	39	2	361
3	1	1	1	7	4	671	685
4	10	162	3	119	73	2	369
5	331	22	5	70	13	23	464
6	5	358	12	212	48	13	648
Total	354	555	341	943	273	738	

K-means clustering result for a newspaper articles document data set

For Cluster 1 and Metro class,
Precision = 506/677 = 0.75
Recall = 506/943 = 0.54
F-value = 0.63

For Cluster 3 and Sports class,
Precision = 671/685 = 0.98
Recall = 671/738 = 0.91
F-value = 0.94

Measuring Cluster Validity Via Correlation

- Two matrices
  - Proximity Matrix
  - Ideal Similarity Matrix
    - One row and one column for each data point
    - An entry is 1 if the associated pair of points belong to the same cluster
    - An entry is 0 if the associated pair of points belongs to different clusters
- Compute the correlation between the two matrices
  - Since the matrices are symmetric, only the correlation between n(n-1) / 2 entries needs to be calculated.
- High correlation indicates that points that belong to the same cluster are close to each other.
- Not a good measure for some density or contiguity based clusters.

Determine the Number of Clusters

Empirical method

- # of clusters ≈√(n/2) for a dataset of n points

Elbow method

- Use the turning point in the curve of sum of within cluster variance w.r.t the # of clusters

Cross validation method

- Divide a given data set into *m* parts
- Use *m* – 1 parts to obtain a clustering model
- Use the remaining part to test the quality of the clustering
  - E.g., For each point in the test set, find the closest centroid, and use the sum of squared distance between all points in the test set and the closest centroids to measure how well the model fits the test set
- For any k > 0, repeat it *m* times, compare the overall quality measure w.r.t. different *k's*, and find # of clusters that fits the data the best

Final Comment on Cluster Validity

"The validation of clustering structures is the most difficult and frustrating part of cluster analysis.

Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage."

Algorithms for Clustering Data, Jain and Dubes

Prescribed Text Books

	Author(s), Title, Edition, Publishing House
T1	Tan P. N., Steinbach M & Kumar V. “Introduction to Data Mining” Pearson Education
T2	Data Mining: Concepts and Techniques, Third Edition by Jiawei Han, Micheline Kamber and Jian Pei Morgan Kaufmann Publishers