**Cluster analysis: Basic concepts and algorithms**

**Jin Miao, Shuxuan Zeng**

Part I: Introduction and overview

**Concept:** cluster analysis divides data into groups that are meaningful or useful.

**Managerial implications:** market segmentation

**Why CA?**

1) Summarization;

2) Compression;

3) Efficiency in finding nearest neighbors.

**Goal:** within a group, the more similarity, the better; between groups, the greater differences, the better.

**Types of classification:**

1) Hierarchical vs. partitional;

2) Exclusive vs. overlapping vs. fuzzy;

3) Complete vs. partial

**Types of clusters:**

1) Well-separated

2) prototype-based

3) graph-based

4) density-based

Part II: K-means

**Algorithm:**

1. Select K initial points as centroids
2. Form K clusters and assign each point to the cluster where its closest centroid is
3. Recomputed centroid of each cluster
4. Repeat 2) and 3) until centroids do not change

**Centroids and objective function**

We are going to have K clusters, C1, C2, …, Ck, with Ci representing the i-th cluster. **ci** denotes the centroid of i-th cluster. For the i-th cluster, there are mi objects in it.

As in the second step, we have to assign each object to its closest centroid. The distance could vary by definition. Euclidean (L2) distances are commonly used for points in Euclidean space. For any point **x**, dist (**ci**, **x**) is defined in advance. Popular examples for it are like: Maharran (L1), Euclidean distance (L2), cosine or Bregman divergence.

The objective function is SSE = 2

Our goal is to minimize the objective function.

Centroid of each cluster is computed as **ci** =

When choosing the centroids of the clusters, a common practice is to choose initial centroids randomly. But the resulting clusters may be poor and also different by choosing different centroids.

**Time and space complexity**

The storage: O((m+K)n), where m is the number of points, n is the number of attributes.

The time: O(I\*K\*m\*n), where I is the number of iterations required for convergence.

An alternative for the steps for the algorithm is that we could choose to update centroid incrementally, which means that after assigning a point to its closest cluster, we can update the centroid of that cluster correspondingly. This step results in better accuracy and faster convergence.

**Strength and weakness**

Strength:

* Simple and could be used for a wide range of data types;
* Efficient

Weakness:

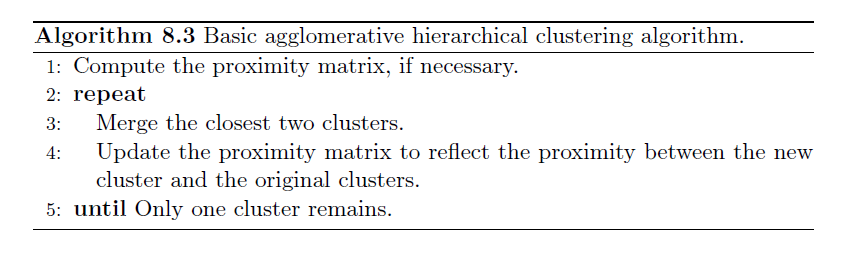
* Not suitable for all types of data
* Cannot handle non-globular clusters or clusters of different sizes and densities
* Has trouble clustering data which contains outliers

Part III: Agglomerative Hierarchical Clustering



Agglomerative: from individual clusters, merge the closest pairs

Divisive: from all-inclusive cluster, split a cluster until singletons



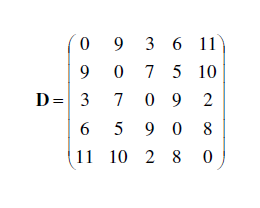
**Key Question: How to update cluster proximity?**

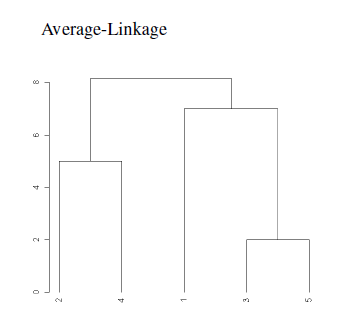
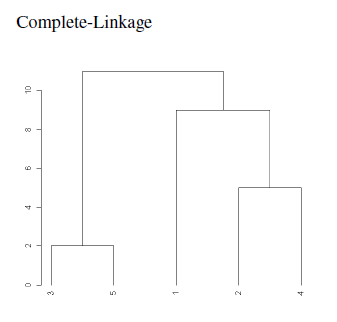
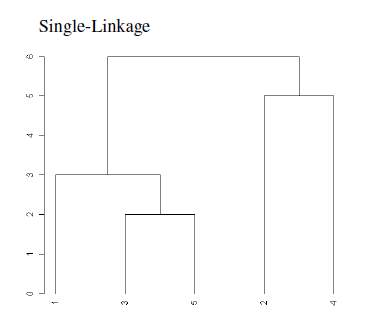
Single Link (MIN): The minimum of the distance between any two points in the two different clusters

Complete Link (MAX): The maximum of the distance between any two points in the two different clusters

Group Average: the average pairwise proximity among all pairs of points in different clusters

Proximity Matrix (Distance Matrix)





**Comparison between MIN, MAX, average**

MIN: good at non-elliptical shapes; sensitive to noise and outlier

Average: intermediate

MAX: robust to noise and outlier; break large clusters and favour globular shapes

**Ward’s Method**

The proximity between two clusters is defined as the increase in the squared error from merging two clusters. (similar to average method)

Variation within groups + variation between groups.

Problem: Inversions (two clusters merged may be more similar than the pairs of clusters that were merged in a previous step)

**Critiques for Hierarchical Clustering**

Lack of a global objective function (decisions are made locally)

Cluster sizes (weighted vs. unweights in terms of data points)

Merging decisions are final (Local Optimal 🡪 Global Optimal )

Expensive in computational complexity

Noise and outliers (undoing is prohibited)

Part IV: DBSCAN

**Definition of density, core points, border points, noise points**

**Density:**

For a point x, its density is defined as the number of points within a specified radius, Eps, of x (including itself).

**Core points:**

Given a distance parameter Eps, if the point x’s density (no. of points within distance Eps of x, including x) is greater or equal to a certain threshold, MinPts, x is a core point.

**Border point:**

A border point is not a core point, but falls within the neighborhood of a core point (within radius Eps of a core point).

**Noise point:**

A point which is neither a core point nor a border point.

**DBSCAN algorithm**

1. Label all points core points, border points, or noise points.
2. Eliminate noise points
3. Put an edge between core points that are within distance Eps of each other
4. Make each group of connected core points a cluster
5. Assign border points to one of the clusters where its associated core points are

**Time and complexity**

Time: O(m\* time to find points in the Eps-neighborhood), in the worst case, this complexity is O(m2)

Space: O(m)

**Selection of parameters**

Basic approach is to look at the behavior of the distance from a point to its k-th nearest neighbor, which it is called the k-dist. As for k-dist of different points, we expect to see a sharp change for k-dist, for which the value could be used as Eps. And the value k could be treated as MinPts. For DBSCAN algorithm, the commonly used value is k=4, which is reasonable for most 2 dimensional data sets.

**Strength and weakness**

Strength:

Can find many clusters that could not be found using K-means

Weakness:

Has trouble when the clusters have widely varying densities

Has trouble with high-dimensional data

Part V: Cluster Evaluation (Cluster Validation)

**Types** Supervised (Internal Indices); Unsupervised (External Indices); Relative (comparison)

**Unsupervised Evaluation**

**Graph-based view of cohesion and separation**

**Prototype-based view of cohesion and separation**

**Measuring Cluster Validity via Correlation of Proximity Matrix**

*The correlation between the similarity matrix and an ideal version of the similarity matrix based on the cluster labels*

**Hierarchical Clustering**

*CoPhenetic Correlation Coefficients*

**Number of Clusters -- SSE**

**Supervised Evaluation**

Classification-oriented: *entropy, purity, precision, recall, F-measure*

Similarity-oriented: correlation between cluster and class matrices

**General Remarks**

Cluster validity measures are hard to interpret.

It is difficult to calculate statistical significance levels for both unsupervised, supervised and relative measures.