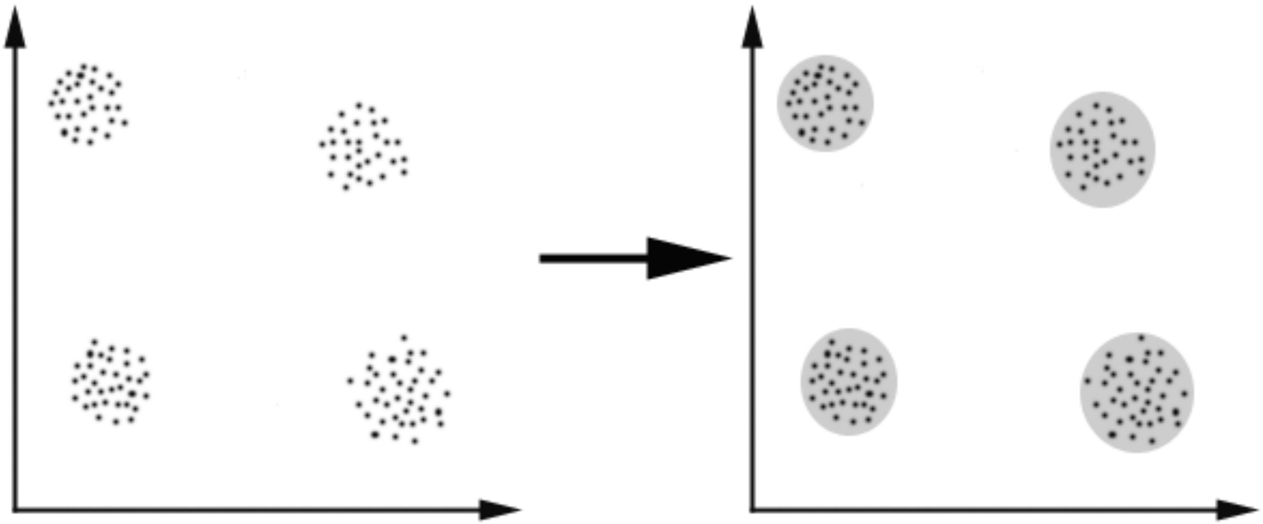
HIERARCHICAL CLUSTERING

Single-link clustering example

Note: For CD1 course, we only consider Single-Linkage Clustering

Introduction

- What is clustering?
  - Most important unsupervised learning problem
  - Find structure in a collection of unlabeled data
  - The process of organizing objects into groups whose members are similar in some way



Example of distance based clustering

Goals of Clustering

- Data reduction:
  - Finding representatives for homogeneous group
- Natural data types:
  - Finding natural clusters and describe their property
- Useful data class:
  - Finding useful and suitable groupings
- Outlier detection
  - Finding usual data objects

Applications

- Marketing
- Biology
- Libraries
- Insurance
- City-planning
- Earthquake studies

Requirements

- Scalability
- Dealing with different types of attributes
- Discovering clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Ability to deal with noise and outliers
- Insensitivity to order of input records
- High dimensionality
- Interpretability and usability

Clustering Algorithms

- Exclusive Clustering
  - K-means
- Overlapping Clustering
  - Fuzzy C-means
- Hierarchical Clustering
  - Hierarchical Clustering
- Probabilistic Clustering
  - Mixture of Gaussians

Hierarchical Clustering (Agglomerative)

- Given a set of N items to be clustered, and an N*N distance matrix, the basic process of hierarchical clustering is:
  - Step 1. Assign each data as a cluster, so we have N clusters from N items. Distance between clusters=distance between the items they contain
  - Step 2. Find the closest pair of clusters and merge them into a single cluster (become N-1 clusters)
  - Step 3. Compute the distances between the new cluster and each of the old cluster
  - Step 4. Repeat step 2 and 3 until all clusters are combined into a single cluster of size N.

Remember its Connection with Kruskal's MST Algorithm

Illustration

Ryan Baker

Different Algorithms to calculate distances

- Single-linkage clustering
  - **Shortest** distance from any member of one cluster to any member of the other cluster
- Complete-linkage clustering
  - **Greatest** distance from any member of one cluster to any member of the other cluster
- Average-linkage clustering
  - **Average** distance from …
- UCLUS method by R.D'Andrade
  - **Median** distance from …

Single-linkage clustering example

- Cluster cities

http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/hierarchical.html

To Start

- Calculate the N*N proximity matrix D=[d(i,j)]

	BA	FI	MI	NA	RM	TO
BA	0	662	877	255	412	996
FI	662	0	295	468	268	400
MI	877	295	0	754	564	138
NA	255	468	754	0	219	869
RM	412	268	564	219	0	669
TO	996	400	138	869	669	0

- The clustering are assigned sequence numbers k from 0 to (n-1) and L(k) is the level of the kth clustering.

Algorithm Summary

- Step1. Begin with disjoint clustering having level L(0)=0 and sequence number m=0
- Step 2. Find the most similar(smallest distance) pair of clusters in the current clustering (r),(s) according to

  d[(r),(s)]=min d[(i),(j)]
- Step 3. Increment the sequence number from m→m+1 Merge clusters r, s to a single cluster. Set the level of this new clustering m to

  L(m)= d[(r),(s)]
- Step 4. Update the proximity matrix, D by deleting the rows and columns of (r), (s) and adding a new row and column of the combined (r, s). The proximity of the new cluster (r, s) and old cluster (k) is defined by

  d[(k),(r, s)]=min {d[(k), (r)], d[(k), (s)] }
- Step 5. Repeat from step 2 if m<N-1, else stop as all objects are in one cluster now

Iteration 0

- The table is the distance matrix D=[d(I,j)]. m=0 and L(0)=0 for all clusters.

	BA	FI	MI	NA	RM	TO
BA	0	662	877	255	412	996
FI	662	0	295	468	268	400
MI	877	295	0	754	564	138
NA	255	468	754	0	219	869
RM	412	268	564	219	0	669
TO	996	400	138	869	669	0

Iteration 1

- Merge MI with TO into MI/TO, L(MI/TO)=138 m=1

	BA	FI	MI/TO	NA	RM
BA	0	662	877	255	412
FI	662	0	295	468	268
MI/TO	877	295	0	754	564
NA	255	468	754	0	219
RM	412	268	564	219	0

Iteration 2

- merge NA, RM→NA/RM, L(NA/RM)=219, m=2

	BA	FI	MI/TO	NA/RM
BA	0	662	877	255
FI	662	0	295	268
MI/TO	877	295	0	564
NA/RM	255	268	564	0

Iteration 3

- Merge BA and NA/RM into BA/NA/RM
- L(BA/NA/RM)=255, m=3

	BA/NA/RM	FI	MI/TO
BA/NA/RM	0	268	564
FI	268	0	295
MI/TO	564	295	0

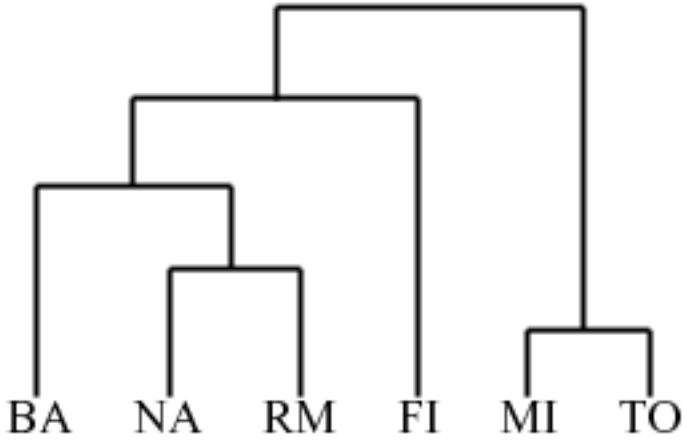
Iteration 4.

- Merge FI with BA/NA/RM into FI/BA/NA/RM
- L(FI/BA/NA/RM)=268, M=4

	BA/FI/NA/RM	MI/TO
BA/FI/NA/RM	0	295
MI/TO	295	0

Hierarchical tree (Dendrogram)

- The process can be summarized by the following hierarchical tree



Demo

- http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/AppletH.html

Note: Complete Link and Average Link are for your info only.

Complete-link clustering

- **Complete-link distance** between clusters C_i and C_j is the ***maximum distance*** between any object in C_i and any object in C_j

- The distance is **defined by the two most dissimilar objects**

$$D_{cl}\left(C_i, C_j\right) = \max_{x,y}\left\{d(x,y) \middle| x \in C_i, y \in C_j\right\}$$

Group average clustering

- Group average distance between clusters Ci and Cj is the average distance between any object in Ci and any object in Cj

$$D_{avg}\left(C_i, C_j\right) = \frac{1}{\left|C_i\right| \times \left|C_j\right|} \sum_{x \in C_i, y \in C_j} d(x, y)$$

Comparison

Distance Algorithm	Advantage	Disadventage
Single-link	Can handle non-elliptical shapes	•Sensitive to noise and outliers • It produces long, elongated clusters
Complete-link	• More balanced clusters • Less susceptible to noise	• Tends to break large clusters • All clusters tend to have the same diameter-small clusters are merged with large ones
Group Average	• Less susceptible to noise and outliers	Biased towards globular clusters

Resources

- Princeton web math
  - http://web.math.princeton.edu/math_alive/5/Notes2.pdf
- A tutorial on clustering algorithms
  - http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/hierarchical.html
- Andrew Moore
  - K-means and Hierarchical clustering http://www.autonlab.org/tutorials/kmeans.html
- Ryan S.J.d. Baker
  - Big Data Education, video lecture week 7, couresa https://class.coursera.org/bigdata-edu-001/lecture
- Chris Caldwell
  - Graph theory tutorials
  - http://www.utm.edu/departments/math/graph/