Dataset: n observations (rows)
p variables (columns)

Today: High-D Continuous Data, Clustering

Sam Ventura

36-315

Today: Distance Matrices,
Hierarchical Clustering, Dendrograms

Department of Statistics
Carnegie Mellon University

March 28, 2016

d

Distance = Metric = Distance Metric = Distance Function

Function that defines distance between pairs of observations in a dataset

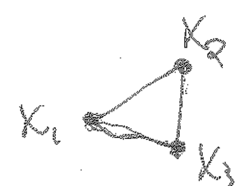
Properties: X_i, X_j are observations (rows) in orig. data

Non-negativity: $d(X_i, X_j) \geq 0$

Identity: $d(X_i, X_j) = 0 \iff X_i = X_j$

Symmetry: $d(X_i, X_j) = d(X_j, X_i)$

Triangle Inequality: $d(X_1, X_3) \leq d(X_1, X_2) + d(X_2, X_3)$



Examples:

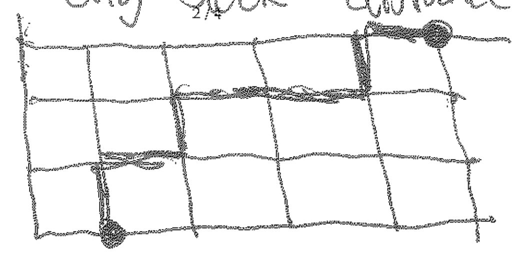
Euclidean distance ⇒ pythagorean

$$d(X_i, X_j) = \sqrt{\sum_{k=1}^{p} (X_{i,p} - X_{j,p})^2}$$

"as the bird flies"

Manhattan Distance

"city block" distance



→ n rows, n colums

$$\begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \end{bmatrix}$$

Distance Matrices

A **distance matrix** is a data structure that efficiently organizes the pairwise distances between all observations in a dataset.

Pairwise distances are organized into the lower-triangle of a matrix, D

The $(i,j)^{th}$ element of the matrix contains the distance between x_i and x_j:

$D[i,j] = d(x_i, x_j)$ $= d(x_j, x_i) = D[j,i]$ → also is symmetric

D

Examples:

Suppose original has 4 observations. ⇒ Distance matrix will be 4x4

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 3 & \pi & 2 \\ 2 & 3 & 0 & 1/2 & \\ 3 & \pi & 1/2 & 0 & x \\ 4 & 2 & & x & 0 \end{array}$$

→ Diagonal of D = 0

→ cell (i,j) = cell (j,i)

→ D is symmetric.

→ Non-negative

3/4

Visualizing Distances / High-Dim Structure

Goal: link observations into groups / "clusters"

There is no easy way to visualize how far apart observations are in high-dimensional space.

As we saw on the last Lab / current HW, we can project the data into lower-dim space and visualize the results. Why might this be a bad idea? → losing information!

→ similar to min. spanning tree

Another option: Use **hierarchical linkage clustering.**

0. start w/ all obs. in data set in their own group (n groups)
1. find distance between all pairs of obs. (distance matrix)
2. Link up the two "closest" groups (obs)
3. Re-find distances / "updating" distances

alternate between #2 and #3 until we have all obs. in one group

ways to define "closest"

Single linkage: the distance between two groups is the shortest possible distance between two points, one from each group

Complete linkage: the distance between two groups is the largest possible distance between two points, one from each group

4/4

After each iteration, new grouping of our n observations

All iterations together = "hierachy" of groupings

↳ "hierarchical clustering"