Distance, Similarity, and Clustering

• Various methods for creating clusters of similar items from $N$ multivariate observations $x_1, x_2, \ldots, x_N$.

• Simplest case is when similarity is measured by distance:

$$d(x_i, x_k) = \sqrt{(x_{i,1} - x_{k,1})^2 + (x_{i,2} - x_{k,2})^2 + \cdots + (x_{i,p} - x_{k,p})^2}$$

$$= \sqrt{(x_i - x_k)' (x_i - x_k)}$$

$i, k = 1, 2, \ldots, n$.

• Possibly statistical distance (but $S$ may be hard to find):

$$d(x_i, x_k) = \sqrt{(x_i - x_k)' S^{-1} (x_i - x_k)}$$
• Other (non-Euclidean) measures of distance can be used, provided they satisfy:

\[ d(x, y) = d(y, x) \]
\[ d(x, y) \geq 0 \quad \text{with equality if and only if } x = y \]
\[ d(x, y) \leq d(x, z) + d(z, y) \]

• E.g. Minkowski metric

\[ d(x, y) = \left( \sum_{i=1}^{p} |x_i - y_i|^m \right)^{1/m} \]
• $m = 1$: “city block” metric:

$$d(x, y) = \sum_{i=1}^{p} |x_i - y_i|$$

• $m = 2$: Euclidean distance.

• $m = \infty$:

$$d(x, y) = \max_{i} |x_i - y_i|$$
• With categorical data, distance cannot be defined directly.

• Instead, various ways have been developed to measure similarity, e.g. the number of matches between elements of $x_i$ and $x_k$.

• If the similarities $\tilde{s}_{i,k}$ are scaled so that each item’s similarity to itself is $\tilde{s}_{i,i} = 1$, and the similarity matrix is non-negative definite, then distances defined by

$$d_{i,k} = \sqrt{1 - \tilde{s}_{i,k}}$$

satisfy the properties of distance, including the triangle inequality

$$d_{i,k} \leq d_{i,l} + d_{l,k}$$
Hierarchical Clustering

• Agglomerative or Divisive.

• Agglomerative clustering of $N$ items:

  1. Begin with $N$ clusters, each with a single item.
  2. Merge the two closest clusters.
  3. Repeat step 2 until only one cluster remains.
Linkage

• How to measure the distance between clusters $U$ and $V$?

  – *Single linkage*: minimum distance

    \[
    d_{\text{single}}(U, V) = \min_{i, k: x_i \in U, x_k \in V} d_{i, k}
    \]

  – *Complete linkage*: maximum distance

    \[
    d_{\text{complete}}(U, V) = \max_{i, k: x_i \in U, x_k \in V} d_{i, k}
    \]

  – *Average linkage*: average distance

    \[
    d_{\text{average}}(U, V) = \text{ave}_{i, k: x_i \in U, x_k \in V} d_{i, k}
    \]
Example: Concordant First Letters for Numbers

- Data: first letter in the words for “one” to “ten” in English, Norwegian, Danish, Dutch, German, French, Spanish, Italian, Polish, Hungarian, and Finnish.
Concordance (Table 12.3):

```r
> concordance = read.table("concordance", header = TRUE);
> concordance;

<table>
<thead>
<tr>
<th></th>
<th>En</th>
<th>No</th>
<th>Da</th>
<th>Du</th>
<th>Ge</th>
<th>Fr</th>
<th>Sp</th>
<th>It</th>
<th>Po</th>
<th>Hu</th>
<th>Fi</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>8</td>
<td>8</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>10</td>
<td>9</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>10</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>10</td>
<td>8</td>
<td>9</td>
<td>5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>8</td>
<td>10</td>
<td>9</td>
<td>7</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>9</td>
<td>9</td>
<td>10</td>
<td>6</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>7</td>
<td>6</td>
<td>10</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
```
• Convert to “dissimilarities” (i.e., distances):

> dissimilarities = as.dist(10 - concordance);

• Cluster, and make *dendrogram* plots:

> plot(hclust(dissimilarities, "single"), hang = -1);
> plot(hclust(dissimilarities, "complete"), hang = -1);
> plot(hclust(dissimilarities, "average"), hang = -1);
> plot(hclust(dissimilarities, "ward"), hang = -1);
> # Note inversions with method = "median" or "centroid":
> plot(hclust(dissimilarities, "median"), hang = -1);
> plot(hclust(dissimilarities, "centroid"), hang = -1);

• Interpretation is usually based on clusters at some intermediate height.
Non-hierarchical Clustering

- E.g. $K$-means clustering:

- The number of clusters, $K$, is given, and an initial set of clusters (e.g. from hierarchical clustering).

- For every item:
  - Allocate the item to the cluster with the closest centroid.
  - If it changes cluster, update the centroids.

- Repeat until no more reassignments take place.
Example: Public Utilities

- utilities = read.table("JandW/T12-04.dat");
  sd(utilities[, 1:8]);
  u = apply(utilities[, 1:8], 2, function(x) x/sd(x));
  kmeans(u, 4);
  split(utilities[,9], kmeans(u, 4)$cluster);