1. CLUSTERING

1.1 Introduction

We have discussed *supervised learning* in the previous lectures. We will now deal with *unsupervised learning* or *clustering* where class labeling of the training patterns is not available. Our aim here is to organize the feature vectors into *sensible* clusters so that we can derive useful conclusions. How we organize into sensible clusters depends on the application and the users.

The basic steps in clustering can be listed as follows:

Feature selection:
: Features should be selected in such a way that we can make use of as much information as possible.

Proximity measures:
: This quantifies how *similar* or *dissimilar* two feature vectors are.

Clustering criterion:
: This depends on the interpretation of the term *sensible*.

Clustering algorithms:
: Choice of the specific algorithmic scheme for the particular application.

Validation of results:
: Verifying the correctness of the results using appropriate test.

Interpretation of the results:
: Experts in the application field analyze the results and draw the right conclusions.

1.2 Types of features

Based on the relative significance of the values the features may take, they are categorized into *nominal*, *ordinal*, *interval-scaled* and *ratio-scaled*.

1.3 Types of clusters

Based on the belongingness of a vector to the cluster, the clustering can be categorized into *hard* and *fuzzy* clustering. In hard clustering, each feature vector belongs to a single cluster. In fuzzy clustering, each vector x belongs to more than one cluster simultaneously *up to some degree*.

2. PROXIMITY MEASURES

2.1 Dissimilarity and similarity measures

A *Dissimilarity measure* is a function $d : X \times X \rightarrow R$ such that

$$\exists d_0 \in R : -\infty < d_0 \le d(x, y) < +\infty, \forall x, y \in X$$

$$d(x, x) = d_0, \forall x \in X$$

and

$$d(x, y) = d(y, x), \forall x, y \in X$$

If

$d(x, y) = d_0$ if and only if $x = y$

and $d(x, z) \le d(x, y) + d(x, z), \forall x, y, z \in X$

d is called a *metric DM*. *e.g.*, Euclidean distance.

A *similarity measure* is a function $s : X \times X \to R$ such that

$$\exists s_0 \in R : -\infty < s(x, y) \le s_0 < +\infty, \forall x, y \in X$$

$$s(x, x) = s_0, \forall x \in X$$

and

$$s(x, y) = s(y, x), \forall x, y \in X$$

If

$s(x, y) = s_0$ if and only if $x = y$

and $s(x, y)s(y, z) \le [s(x, y) + s(y, z)]s(x, z), \forall x, y, z \in X$

s is called a *metric SM*.

2.2 Proximity measure between points

2.2.a. *For real valued vectors*

A common general-purpose metric DM is *weighted* L_p norm, given by

$$d_p(x, y) = \left(\sum_{i=1}^{l} w_i |x_i - y_i|^p \right)^{1/p}$$

whose special cases include *weighted Euclidean distance* and *weighted Manhattan distance.*

A sample SM would be the *inner product* given by

$$s_{inner}(x, y) = x^T y = \sum_{i=1}^{l} x_i y_i$$

2.2.b. *For discrete valued vectors*

We can use the SMs and DMs for real valued vectors themselves. In addition, we can also use

Hamming distance – which is a DM that measures the number of places where x and y differ

Tanimoto distance – which is an SM that measures the number of places where x and y are the same divided by the total number of places.

2.2.c. *Fuzzy measures*

When we use fuzzy measures, we end up with a degree of similarity

$$s(x_i, y_i) = \max\{\min\{1 - x_i, 1 - y_i\}, \min\{x_i, y_i\}\}, \text{ where } x_i, y_i \in [0,1].$$

We can specify a similarity measure between two vectors

$$s_f^q(x, y) = \left(\sum_{i=1}^{l} s(x_i, y_i)^q \right)^{1/q}$$

2.3 Proximity measure between point and set

There are two different ways of measuring this. We can measure the proximity by using all points of a cluster or by using a *representative* of the cluster. If we use all points, we can use the *maximum*, *minimum* or *average* over the proximity measure between the point and all points in the given cluster. A cluster can also be represented by a point (compact clusters), hyperplane (elongated clusters) or a hypersphere (hyperspherical clusters). In this case we measure the proximity between the point and the representation of the cluster.

2.4 Proximity measure between two sets

We can use the maximum or minimum proximity between any two points in the two clusters, or the average proximity measure over all points in the two clusters. We can also use the proximity between the representatives of the two clusters.

3. CLUSTERING ALGORITHMS

3.1 Sequential algorithms

These produce a single clustering by processing the feature vectors sequentially.

3.1.a. *Basic Sequential Algorithmic Scheme (BSAS)*

We use a *threshold of similarity*, Θ, in this algorithm. The basic idea is that as each new vector is considered, it is either assigned to an existing cluster or assigned to a newly created cluster. Θ decides whether or not to create a new cluster.

3.1.b. *Modified Basic Sequential Algorithmic Scheme (MBSAS)*

MBSAS is a modified version of BSAS; it *only* processes the feature vectors and creates new clusters if necessary and defers assignment until all clusters have been formed. The two-phase process in MBSAS can be explained as follows:

1. Determine the clusters by scanning list and creating new clusters when necessary, but do not assign any feature vectors to existing clusters
2. Once clusters are determined (each represented by one feature vector), assign each remaining (unassigned) feature vector to its best cluster.

3.1.c. *Two Threshold Sequential Algorithmic Scheme (TTSAS)*

To reduce the sensitivity of the algorithm to Θ, we use two different thresholds Θ_1 and Θ_2. After we find the best cluster C_k for x_i, we assign x_i to only if $d(x_i, C_k) < \Theta_1$ and create new cluster only if $d(x_i, C_k) > \Theta_2$. Otherwise we defer decision on x_i. The deferred vectors are handled separately.

3.2 Hierarchical algorithms

These produce a hierarchy of clusterings and the resultant clustering depends on where the hierarchy is broken. These algorithms are of two types – *agglomerative* and *divisive*. Agglomerative algorithms start with a set of clusters with each vector assigned to one unique cluster. At each step, the algorithm merges two clusters into one where the old clustering is nested in the new clustering. At the final step, we have all the vectors assigned to one single cluster. Divisive algorithms start with a single cluster containing all the points. At each step, they split a cluster into two where the new clustering is nested in the old clustering. At the final step, we have one unique cluster for each vector.

3.2.a. *Generalized Agglomerative Scheme (GAS)*

We start with a clustering in which each vector is assigned a unique cluster. At each step we find the closest pair of clusters (call them *old clusters*), merge them into one cluster, remove both the old clusters and put the new merged cluster in the clustering, and update the representatives if necessary. We do this until we end up with a single cluster.

3.2.b. *Implementation*

We use a proximity matrix P_t that gives the proximity between all pairs of clusters at level t. Each step of GAS can be represented by a proximity matrix. When we update the matrix by introducing a new cluster, we have to compute the proximities between the new cluster and the remaining clusters. For every remaining cluster, we can use either the minimum or the maximum of the proximities between the cluster and the two old clusters. *i.e.*, if clusters C_i and C_j have been merged into C_q, the proximity between cluster C_q and any cluster C_s can be given by

$$\alpha(C_q, C_s) = \min\{ \alpha(C_i, C_s), \alpha(C_j, C_s)\}$$

or

$$\alpha(C_q, C_s) = \max\{ \alpha(C_i, C_s), \alpha(C_j, C_s)\}$$

If we use the minimum, it yields the *single link algorithm* for DM, and *complete link algorithm* for SM. If we use the maximum, it is the reverse.

3.2.c. *Single versus complete link algorithms*

Single link algorithms favor elongated clusters since they look for the closest members across two clusters. Complete link algorithms favor compact clusters since they look for the farthest members across two clusters.

Appendix

Questions

1. The average set-to-set measure α_{avg} is not necessarily a measure even if α is, whereas the mean (representative) set-to-set measure α_{rep} is a measure whenever α is. We would be able to prove the fact mathematically, but why does this happen? Though these two measures are not the same, theoretically they definitely seem analogous.
2. In lecture 9 slide 11, fig.(c), in the second cluster containing x_8, x_9, x_{10} and x_{11}, I think x_{10} and x_{11} should be merged together since they are of distance 2.6 which is less than 4.5. And then we can merge $\{x_{10}, x_{11}\}$ and $\{x_8, x_9\}$. I checked the book. It is the same as the slide.

Most interesting result

Single-link algorithm produces same dendogram regardless of how ties are broken in proximity matrix; complete-link and others do not possess this property

Least interesting result

Brute force approach of clustering is infeasible.

Research topics

1. Application of clustering in phylogeny.
2. Find new *metric* proximity measures.