Data Mining:
Clustering/Unsupervised Learning
Hierarchical Clustering

Hierarchical Clustering

Hierarchical Clustering. **Hierarchical clustering** methods construct a **dendrogram** of the input dataset. Unlike partitional clustering methods, **hierarchical clustering** methods do not produce specific partitions of data into disjoint clusters. Rather, the data is organized in the dendrogram by percieved similarity.

*By **cutting** the dendrogram horizontaly, we obtain clusters. The clusters are associated with the specific cut.* The cuts are typically guided by a **threshold** on the similarity of data points within the cluster.

Dendrogram. A **dendrogram** of a dataset is a **labeled binary tree** with the following properties:

- The **leaves** of the **dendrogram** are individual datapoints from the input dataset D. Each point of the dataset is associated with **exactly one leaf**.
- The **internal nodes** of the **dendrogram** are labeled. Typically, labels are **real numbers**.
- Each internal node represents a cluster of data points assembled at a specific threshold. The label of the node represents the threshold (similarity between the data points from the two *child clusters*).

Cut. A **dendrogram** is **cut** using some **threshold** α as follows:

> All nodes of the **dendrogram** with labels greater than α are removed from the **dendrogram**, together with any adjacent edges.The resulting **forest** represents the clusters found by a **hierarchical clustering method** that constructed the **dendrogram**, at the threshold α.

Hierarchical Clustering Algorithms

There are two types of **hierarchical clustering algorithms**:

Divisive hierarchical clustering: these algorithms start by treating an entire dataset as a single cluster. On each step they find a way to split one of the currently observed clusters into a pair and construct the appropriate part of the **dendrogram**. **Divisive clustering algorithms** are **top down**.

Agglomerative hierarchical clustering: these algorithms start by treating each point in the dataset as a single cluster. On each step, the algorithm makes a decision to merge two existing clusters, and constructs the apprpriate portion of the **dendrogram**. **Agglomerative clustering algorithms** are **bottom up**.

We consider **Agglomerative algorithms** further.

Agglomerative Hierachical Clustering Algorithms

Input. Dataset $D = \{x_1, \ldots x_n\}$.

Output. A **dendrogram** T of the dataset D.

Algorithm Idea. The algorithm proceeds as follows:

1. On step 1, each point $x \in D$ is assigned to its own cluster.
2. On each step, the algorithm computes the **distance matrix** for the current list of clusters.
3. It then selects a pair of clusters with the shortest distance, and merges these two clusters into one (constructing the apprpriate part of the dendrogram).
4. The algorithm stops when all points are merged into a single cluster.

Algorithm. The pseudocode for the algorithm is shown in Figure 1.

Distances Between Clusters

Single-link method. The distance between two clusters is the **distance between two closest points** in the clusters.

Complete-link method. The distance between two clusters is the **distance between two points that are furtherest away from each other** in the two clusters.

Average-link method. The distance between two clusters is the **average of all pairwise distances.**

```
Algorithm Agglomerative(D).
begin
  foreach x_i ∈ D do C_{1i} = {x_i};
  C_1 := {C_{11}, ..., C_{1n}};
  i := 1;
  while |C_i| > 1 do
     for j = 1 to |C_i| do
        for k = j + 1 to |C_i| do
           d[j,k] := dist(C_{ij}, C_{ik});
        endfor
     endfor
     (s,r) := arg min(d[j,k]);
     for j := 1 to |C_i| do
        if j ≠ r and j ≠ s then C_{i+1,j} := C_{ij}
        else if j = r then C_{i+1,j} := C_{ir} ∪ C_{is};
  end while
end
```

Figure 1: Agglomerative Clustering Algorithm.

Centroid method. The distance between two clusters is the **distance between the cluster centroids**.

Ward's methods. The distance between two clusters is the **increase in the sum of squared error** of distances.

Cluster Evaluation

Note: For supervised learning, we had *training sets*, so we knew, when our predictors made mistakes. For unsupervised learning methods, we do not have the notion of **ground truth** built into the problem.

User Inspection. Human experts evaluate the quality of each cluter. **Pros.** Lets "customers" have a say. **Cons.** Subjective.

Ground Truth. *Use classification datasets.* Use class labels as cluster identifiers. Check is clustering methods correctly assemble the clusters. Accuracy can be computed in the same way as for classification problems.

Enthropy. Use classification datasets, and measure the quality of the clustering via **enthropy**.

Purity. Use classifciation datasets and measure the quality of clustering via **purity**. Let $D = D_1 \cup \ldots \cup D_k$ is the partitioning of D into clusters, let $c_1, \ldots c_k$ be classes.

$$purity(D_i) = \max_j(Pr_i(c_j)).$$

That is, the **purity** of a cluster is the *proportion of the datapoints belonging to the plurality class*.

Machine Learning. Use cluster assignment as class labels, and use supervised learning methods to see if each cluster can be learned. Evaluate the accuracy of classification.

References

[1] S.P. Lloyd. Least Squares quantization in PCM. *Unpublished Bell Labs Tech. Note.* (1957). Portions presented at *Institute for Mathematical Statistics Meeting*, 1957. *IEEE Transactions on Information Theory*, vol. IT-28, pp. 129—137, MArch 1982.

[2] E. Forgey. Cluster Analysis of Multivariate Data: Efficiency vs. Interpretability of Calssification. *Biometrics*, vol. 21, p. 768 (abstract), 1965.