Hierarchical Clustering

4-4-16
Hierarchical clustering: the setting

Unsupervised learning

- no labels/output, only x/input

Clustering

- Group similar points together
Machine learning taxonomy

**Supervised**
- Output known for training set
- Highly flexible; can learn many agent components

- **Regression**
  - Decision trees
  - Naive Bayes
  - K-nearest neighbors
  - SVM

- **Classification**

**Semi-Supervised**
- Occasional feedback
- Learn the agent function (policy learning)

- **value iteration**
- **Q-learning**
- **MCTS**

**Unsupervised**
- No feedback
- Learn representations

- **Clustering**
  - Hierarchical
  - K-means
  - GNG

- **Dimensionality reduction**
  - PCA
The goal of clustering

Given a bunch of data, we want to come up with a representation that will simplify future reasoning.

Key idea: group similar points into clusters.

Examples:

- Identifying objects in sensor data
- Detecting communities in social networks
- Constructing phylogenetic trees of species
- Making recommendations from similar users
Hierarchical clustering

- Organizes data points into a hierarchy.
- Every level of the binary tree splits the points into two subsets.
- Points in a subset should be more similar than points in different subsets.
- The resulting clustering can be represented by a dendrogram.
Direction of clustering

Agglomerative (bottom-up)

• Each point starts in its own cluster.
• Repeatedly merge the two most-similar clusters until only one remains.

Divisive (top-down)

• All points start in a single cluster.
• Repeatedly split the data into the two most self-similar subsets.

Either version can stop early if a specific number of clusters is desired.
Agglomerative clustering

● Each point starts in its own cluster.
● Repeatedly merge the two most-similar clusters until only one remains.

How do we decide which clusters are most similar?

● Distance between closest points in each cluster (single link).
● Distance between farthest points in each cluster (complete link).
● Distance between centroids (average link).
  ○ The centroid is the average position of a cluster: the mean value of every coordinate.
Agglomerative clustering exercise

Which clusters should be merged next?

Under single link?

Under complete link?

Under average link?
Divisive clustering

- All points start in a single cluster.
- Repeatedly split the data into the two most self-similar subsets.

How do we split the data into subsets?

- We need a subroutine for 2-clustering.
- Options include k-means and EM (Wednesday’s topics).
Similarity vs. Distance

We can perform clustering using either a similarity function or a distance function to compare points.

- maximizing similarity ≈ minimizing distance

Example similarity function:
- cosine of the angle between two vectors

Distance metrics have extra constraints:
- Triangle inequality.
- Distance is zero if and only if the points are the same.
Distance metrics

- Euclidean distance
- Generalized euclidean distance
  - p-norm
- Edit distance
  - Good for categorical data.
  - Example: gene sequences.
p-norm

$$\|x\|_p := \left( \sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

- \( p=1 \) Manhattan distance
- \( p=2 \) Euclidean distance
- \( p=\infty \) largest distance in any dimension
Strengths and weaknesses of hierarchical clustering

+ Creates easy-to-visualize output (dendrograms).
+ We can pick what level of the hierarchy to use after the fact.
+ It’s often robust to outliers.
- It’s extremely slow: the basic agglomerative clustering algorithm is $O(n^3)$.
- Each step is greedy, so the overall clustering may be far from optimal.
- Bad for online applications, because adding new points requires recomputing from the start.
Partition-based clustering

- Select the number of clusters, $k$, in advance.
- Split the data into $k$ clusters.
- Iteratively improve the clusters.
Examples of partition-based clustering

k-means

- Pick k random centroids.
- Assign points to the nearest centroid.
- Recompute centroids.
- Repeat until convergence.

EM:

- Assume points drawn from a distribution with unknown parameters.
- Iteratively assign points to most-likely clusters, and update the parameters of each cluster.