Clustering.
Unsupervised Learning

Maria-Florina Balcan
04/15/2019
Clustered, Informal Goals

**Goal**: Automatically partition unlabeled data into groups of similar datapoints.

**Question**: When and why would we want to do this?

**Useful for**:

- Automatically organizing data.
- Understanding hidden structure in data.
- Preprocessing for further analysis.
  - Representing high-dimensional data in a low-dimensional space (e.g., for visualization purposes).
Clustering

• Last time: Partitional objective based clustering
  • Focused on k-means and k-means ++
    • Lloyd’s method
    • Initialization techniques (random, furthest traversal, k-means++)

• Today: hierarchical Clustering.
  • Single linkage, Complete linkage
What value of k???

• Heuristic: Find large gap between k-1-means cost and k-means cost.

• Hold-out validation/cross-validation on auxiliary task (e.g., supervised learning task).

• Try hierarchical clustering.
Hierarchical Clustering

- A hierarchy might be more natural.
- Different users might care about different levels of granularity or even prunings.
Hierarchical Clustering

Top-down (divisive)

- Partition data into 2-groups (e.g., 2-means)
- Recursively cluster each group.

Bottom-Up (agglomerative)

- Start with every point in its own cluster.
- Repeatedly merge the “closest” two clusters.
- Different defs of “closest” give different algorithms.
Bottom-Up (agglomerative)

Have a **distance** measure on pairs of objects.

$$d(x,y)$$ - distance between $$x$$ and $$y$$

E.g., # keywords in common, edit distance, etc

- **Single linkage:**
  $$\text{dist}(C, C') = \min_{x \in C, x' \in C'} \text{dist}(x, x')$$

- **Complete linkage:**
  $$\text{dist}(C, C') = \max_{x \in C, x' \in C'} \text{dist}(x, x')$$

- **Average linkage:**
  $$\text{dist}(C, C') = \frac{\text{avg}}{\text{dist}(x, x')}$$
Single Linkage

Bottom-up (agglomerative)
- Start with every point in its own cluster.
- Repeatedly merge the “closest” two clusters.

Single linkage: $\text{dist}(C, C') = \min_{x \in C, x' \in C'} \text{dist}(x, x')$
Complete Linkage

Bottom-up (agglomerative)
- Start with every point in its own cluster.
- Repeatedly merge the “closest” two clusters.

Complete linkage: \( \text{dist}(S, T) = \max_{x \in S, x' \in T} \text{dist}(x, x') \)

One way to think of it: keep max diameter as small as possible at any level.
Running time for Single and Complete Linkage

• Each algorithm starts with N clusters, and performs N-1 merges.
• For each algorithm, computing $\text{dist}(C, C')$ can be done in time $O(|C| \cdot |C'|)$. (e.g., examining $\text{dist}(x, x')$ for all $x \in C, x' \in C'$)
• Time to compute all pairwise distances and take smallest is $O(N^2)$.
• Overall time is $O(N^3)$.

In fact, can run all these algorithms in time $O(N^2 \log N)$.

What You Should Know

• Partitional Clustering. k-means and k-means ++
  • Lloyd’s method
  • Initialization techniques (random, furthest traversal, k-means++)

• Hierarchical Clustering.
  • Single linkage, Complete linkage