CS 540 Introduction to Artificial Intelligence

Unsupervised Learning I

Yingyu Liang
University of Wisconsin-Madison
Oct 5, 2021

Based on slides by Fred Sala
Recap of Supervised/Unsupervised

**Supervised** learning:

- Make predictions, classify data, perform regression
- Dataset: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Goal: find function $f: X \rightarrow Y$ to predict label on new data

Features / Covariates / Input → Labels / Outputs
Recap of Supervised/Unsupervised

**Unsupervised** learning:

- No labels; generally won’t be making predictions
- Dataset: \( x_1, x_2, \ldots, x_n \)
- Goal: find patterns & structures that help better understand data.

Mulvey and Gingold
Recap of Supervised/Unsupervised

Note that there are other kinds of ML:

– Mixtures: semi-supervised learning, self-supervised
  • Idea: different types of “signal”

– Reinforcement learning
  • Learn how to act in order to maximize rewards
  • Later on in course...
Outline

• Intro to Clustering
  – Clustering Types, Centroid-based, k-means review

• Hierarchical Clustering
  – Divisive, agglomerative, linkage strategies
Unsupervised Learning & Clustering

• Note that clustering is just one type of unsupervised learning (UL)
  – PCA is another unsupervised algorithm

• Estimating probability distributions also UL (GANs)

• Clustering is popular & useful!
Clustering Types

• Several types of clustering

**Partitional**
- Centroid
- Graph-theoretic
- Spectral

**Hierarchical**
- Agglomerative
- Divisive

**Bayesian**
- Decision-based
- Nonparametric
Center-based Clustering

- k-means is an example of partitional center-based
- Recall steps: 1. Randomly pick k cluster centers
Center-based Clustering

- 2. Find closest center for each point
Center-based Clustering

• **3.** Update cluster centers by computing centroids
Center-based Clustering

• Repeat Steps 2 & 3 until convergence
Hierarchical Clustering

Basic idea: build a “hierarchy”

• Want: arrangements from specific to general
• One advantage: no need for k, number of clusters.

• Input: points. Output: a hierarchy
  – A binary tree

Credit: Wikipedia
Agglomerative vs Divisive

Two ways to go:

- **Agglomerative**: bottom up.
  - Start: each point a cluster. Progressively merge clusters

- **Divisive**: top down
  - Start: all points in one cluster. Progressively split clusters

Credit: r2d3.us
Agglomerative Clustering Example

**Agglomerative.** Start: every point is its own cluster
Agglomerative Clustering Example

Get pair of clusters that are closest and merge
Agglomerative Clustering Example

Repeat: Get pair of clusters that are closest and merge
Agglomerative Clustering Example

Repeat: Get pair of clusters that are closest and merge
Merging Criteria

Merge: use closest clusters. Define closest?

• Single-linkage

\[ d(A, B) = \min_{x_1 \in A, x_2 \in B} d(x_1, x_2) \]

• Complete-linkage

\[ d(A, B) = \max_{x_1 \in A, x_2 \in B} d(x_1, x_2) \]

• Average-linkage

\[ d(A, B) = \frac{1}{|A||B|} \sum_{x_1 \in A, x_2 \in B} d(x_1, x_2) \]
Single-linkage Example

We’ll merge using single-linkage

• 1-dimensional vectors.
• Initial: all points are clusters
Single-linkage Example

We’ll merge using single-linkage

\[ d(C_1, \{4\}) = d(2, 4) = 2 \]
\[ d(\{4\}, \{5\}) = d(4, 5) = 1 \]
Single-linkage Example

Continue…

\[ d(C_1, C_2) = d(2, 4) = 2 \]

\[ d(C_2, \{7.25\}) = d(5, 7.25) = 2.25 \]
Single-linkage Example

Continue...

1 - 2 - C1 - 4 - 5 - C2 - 7.25
Single-linkage Example

C_1

C_3

C_4

1 2 4 5

7.25
We’ll merge using complete-linkage

- 1-dimensional vectors.
- Initial: all points are clusters
Complete-linkage Example

Beginning is the same...

\[ d(C_1, C_2) = d(1, 5) = 4 \]
\[ d(C_2, \{7.25\}) = d(4, 7.25) = 3.25 \]
Complete-linkage Example

Now we diverge:
Complete-linkage Example

\[ C_1 \rightarrow C_2 \rightarrow C_4 \rightarrow C_3 \rightarrow 7.25 \]
When to Stop?

No simple answer:

- Use the binary tree (a dendogram)
- Cut at different levels (get different heights/depths)

http://opentreeoflife.org/