Announcements

• **Homeworks:**
  – HW3 due

• **Class roadmap:**

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tuesday, Oct. 4</td>
<td>ML Unsupervised I</td>
</tr>
<tr>
<td>Thursday, Oct. 6</td>
<td>ML Unsupervised II</td>
</tr>
<tr>
<td>Tuesday, Oct. 11</td>
<td>ML Linear Regression</td>
</tr>
<tr>
<td>Thursday, Oct. 13</td>
<td>Machine Learning: K - Nearest Neighbors &amp; Naive Bayes</td>
</tr>
</tbody>
</table>
Recap of Supervised/Unsupervised

**Supervised** learning:

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

**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.

Mulvev 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**
- Center-based
- 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
K-means algorithm

• Input: $x_1, x_2, \ldots, x_n, k$

• Step 1: select $k$ cluster centers $c_1, c_2, \ldots, c_k$

• Step 2: for each point $x_i$, assign it to the closest center in Euclidean distance:

$$y(x_i) = \text{argmin}_j ||x_i - c_j||$$

• Step 3: update all cluster centers as the centroids:

$$c_j = \frac{\sum_{x:y(x)=j} x}{\sum_{x:y(x)=j} 1}$$

• Repeat Step 2 and 3 until cluster centers no longer change
Questions on k-means

• What is k-means trying to optimize?

• Will k-means stop (converge)?

• Will it find a global or local optimum?

• How to pick starting cluster centers?

• How many clusters should we use?
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
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
  \[ d(A, B) = \frac{1}{|A||B|} \sum_{x_1 \in A, x_2 \in B} d(x_1, x_2) \]
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...
Single-linkage Example
We’ll merge using complete-linkage

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

Beginning is the same...

\[ d(C_1, C_2) = d(1, 5) = 4 \]
\[ d(C_2, \{7.25\}) = d(4, 7.25) = 3.25 \]
Now we diverge:

**Complete-linkage Example**

Now we diverge:
Complete-linkage Example
When to Stop?

No simple answer:

• Use the binary tree (a dendogram)

• Cut at different levels (get different heights/depths)

http://opentreeoflife.org/