Today

• Clustering
  - Distance functions
  - K-Means
  - Spectral clustering
  - Hierarchical clustering
Today

• Clustering
  - Distance functions
  - K-Means
  - Spectral clustering
  - Hierarchical clustering
Classification

- **Classification:**
  - Supervised learning
  - Make a rational *prediction* given evidence
  - We’ve seen several methods for this, e.g.
    - Perceptron, Nearest-Neighbor, Support Vector Machine, Boosting
  - Useful when you have *labeled* data (or can get it)
Clustering

- Clustering:
  - Unsupervised learning
  - Requires unlabeled data
  - Detect patterns, e.g.
    - Group emails or search results
    - Find customer shopping patterns
    - Regions of images
  - Useful when don’t know what you’re looking for
  - But: can get gibberish!
**Image segmentation**

- Goal: Break up the image into meaningful or perceptually similar regions
Clustering examples

Clustering gene expression data

- Microarrays measures the activities of all genes in different conditions
- Clustering genes can help determine new functions for unknown genes
Clustering

- **Basic idea:** group together similar instances
- **Example:** 2D point patterns

- What could “similar” mean?
  - One option: small Euclidean distance (squared)
    \[
    \text{dist}(\vec{x}, \vec{y}) = ||\vec{x} - \vec{y}||^2_2
    \]
  - Clustering results are crucially dependent on the measure of similarity (or distance) between “points” to be clustered
• With this objective, it is a “chicken and egg” problem:
  - If we knew the **cluster centers**, we could allocate points to groups by assigning each to its closest center.
  - If we knew the **group memberships**, we could get the centers by computing the mean per group.
What is a natural grouping among these objects?

Clustering is subjective

Simpson's Family  School Employees  Females  Males
What is Similarity?

- The real meaning of similarity is a philosophical question. We will take a more pragmatic approach.
- Depends on representation and algorithm. For many rep./alg., easier to think in terms of a distance (rather than similarity) between vectors.

Hard to define! But we know it when we see it
**Definition:** Let $O_1$ and $O_2$ be two objects from the universe of possible objects. The distance (dissimilarity) between $O_1$ and $O_2$ is a real number denoted by $D(O_1, O_2)$. 

- $0.23$
- $3$
- $342.7$
A few examples:

- **Euclidean distance**

$$d(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$$

- **Correlation coefficient**

$$s(x, y) = \frac{\sum_i (x_i - \mu_x)(y_i - \mu_y)}{\sigma_x \sigma_y}$$

- **Similarity rather than distance**
- **Can determine similar trends**
What properties should a distance measure have?

- **Symmetric**
  - $D(A,B) = D(B,A)$
  - Otherwise, we can say $A$ looks like $B$ but $B$ does not look like $A$

- **Positivity, and self-similarity**
  - $D(A,B) \geq 0$, and $D(A,B) = 0$ iff $A = B$
  - Otherwise there will different objects that we cannot tell apart

- **Triangle inequality**
  - $D(A,B) + D(B,C) \geq D(A,C)$
  - Otherwise one can say “$A$ is like $B$, $B$ is like $C$, but $A$ is not like $C$ at all”
Desirable Properties of a Clustering Algorithm

- Scalability (in terms of both time and space)
- Ability to deal with different data types
- Minimal requirements for domain knowledge to determine input parameters
- Interpretability and usability

- Optional
  - Incorporation of user-specified constraints
Clustering algorithms

- **Partitioning algorithms**
  - Construct various partitions and then evaluate them by some criterion
    - K-means
    - Mixture of Gaussians
    - Spectral Clustering

- **Hierarchical algorithms**
  - Create a hierarchical decomposition of the set of objects using some criterion
    - Bottom-up – agglomerative
    - Top-down – divisive
• Clustering
  - Distance functions
  - K-Means
  - Spectral clustering
  - Hierarchical clustering
K-Means

• **An iterative clustering algorithm**
  - **Initialize:** Pick $K$ random points as cluster centers (means)
  - **Alternate:**
    - Assign data instances to closest mean
    - Assign each mean to the average of its assigned points
  - **Stop when no points’ assignments change**
K-Means Clustering: Example

• Pick $K$ random points as cluster centers (means)

Shown here for $K=2$
K-Means Clustering: Example

Iterative Step 1

- Assign data points to closest cluster centers
Iterative Step 2

- Change the cluster center to the average of the assigned points
K-Means Clustering: Example

• Repeat until convergence
K-Means Clustering: Example
K-Means Clustering: Example
Properties of K-Means Algorithms

• Guaranteed to converge in a finite number of iterations

• Running time per iteration:
  1. Assign data points to closest cluster center
     $O(KN)$ time
  2. Change the cluster center to the average of its assigned points
     $O(N)$ time
Objective

\[
\min_{\mu} \min_{C} \sum_{i=1}^{k} \sum_{x \in C_i} |x - \mu_i|^2
\]

1. Fix \( \mu \), optimize \( C \):

\[
\min_{C} \sum_{i=1}^{k} \sum_{x \in C_i} |x - \mu_i|^2 = \min_{c} \sum_{i} |x_i - \mu_{x_i}|^2
\]

2. Fix \( C \), optimize \( \mu \):

\[
\min_{\mu} \sum_{i=1}^{k} \sum_{x \in C_i} |x - \mu_i|^2
\]

– Take partial derivative of \( \mu_i \) and set to zero, we have

\[
\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x
\]

K-Means takes an alternating optimization approach, each step is guaranteed to decrease the objective – thus guaranteed to converge
Goal of Segmentation is to partition an image into regions each of which has reasonably homogenous visual appearance.
Example: K-Means for Segmentation

Goal of Segmentation is to partition an image into regions each of which has reasonably homogenous visual appearance.

K=2
K=3
Original
Example: K-Means for Segmentation

Goal of Segmentation is to partition an image into regions each of which has reasonably homogenous visual appearance.
Example: Vector quantization

**FIGURE 14.9.** Sir Ronald A. Fisher (1890 – 1962) was one of the founders of modern day statistics, to whom we owe maximum-likelihood, sufficiency, and many other fundamental concepts. The image on the left is a 1024×1024 grayscale image at 8 bits per pixel. The center image is the result of 2×2 block VQ, using 200 code vectors, with a compression rate of 1.9 bits/pixel. The right image uses only four code vectors, with a compression rate of 0.50 bits/pixel.
K-Means Clustering: Some Issues

- How to set k?
- Sensitive to initial centers
- Sensitive to outliers
- Detects spherical clusters
- Assuming means can be computed

Slide credit: K Grauman
Today

• Clustering
  - Distance functions
  - K-Means
  - Spectral clustering
  - Hierarchical clustering
Graph-Theoretic Clustering

Goal: Given data points $X_1, \ldots, X_n$ and similarities $W(X_i, X_j)$, partition the data into groups so that points in a group are similar and points in different groups are dissimilar.

Similarity Graph: $G(V,E,W)$

- $V$ – Vertices (Data points)
- $E$ – Edge if similarity $> 0$
- $W$ - Edge weights (similarities)

Partition the graph so that edges within a group have large weights and edges across groups have small weights.
Graphs Representations

![Graph Diagram](image)

Adjacency Matrix

\[
\begin{bmatrix}
0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & 0
\end{bmatrix}
\]

Slide credit: B. Freeman and A. Torralba
A Weighted Graph and its Representation

Affinity Matrix

\[ W = \begin{bmatrix}
1 & .1 & .3 & 0 & 0 \\
.1 & 1 & .4 & 0 & .2 \\
.3 & .4 & 1 & .6 & .7 \\
0 & 0 & .6 & 1 & 1 \\
0 & .2 & .7 & 1 & 1 \\
\end{bmatrix} \]

\( W_{ij} \) : probability that i & j belong to the same cluster

Slide credit: B. Freeman and A. Torralba
• Similarity Graphs: Model local neighborhood relations between data points

• E.g. epsilon-NN

\[ W_{ij} = \begin{cases} 
1 & \| x_i - x_j \| \leq \epsilon \\
0 & \text{otherwise}
\end{cases} \]

or mutual k-NN graph (\( W_{ij} = 1 \) if \( x_i \) or \( x_j \) is \( k \) nearest neighbor of the other)

Controls size of neighborhood

\[ W_{ij} = \begin{cases} 
\epsilon^{-1} & \| x_i - x_j \| \leq \epsilon \\
0 & \text{otherwise}
\end{cases} \]
• Similarity Graphs: Model local neighborhood relations between data points

• E.g. Gaussian kernel similarity function

\[
W_{ij} = e^{-\frac{||x_i - x_j||^2}{2\sigma^2}}
\]

Controls size of neighborhood

\[G = \{V,E\}\]
• **Small $\sigma$:** group only nearby points
• **Large $\sigma$:** group far-away points
Feature grouping by “relocalisation” of eigenvectors of the proximity matrix

Three points in feature space

\[ W_{ij} = \exp(-||z_i - z_j||^2 / s^2) \]

With an appropriate \( s \)

\[ W= \]

The eigenvectors of \( W \) are:

The first 2 eigenvectors group the points as desired...
Example eigenvector

Points

Affinity matrix

eigenvector

Slide credit: B. Freeman and A. Torralba
Example eigenvector

Affinity matrix

points
eigenvector

Slide credit: B. Freeman and A. Torralba
• Set of edges whose removal makes a graph disconnected
• Cost of a cut: sum of weights of cut edges
• A graph cut gives us a partition (clustering)
  - What is a “good” graph cut and how do we find one?
Minimum cut

• A cut of a graph $G$ is the set of edges $S$ such that removal of $S$ from $G$ disconnects $G$.

**Cut**: sum of the weight of the cut edges:

$$\text{cut}(A, B) = \sum_{u \in A, v \in B} W(u, v),$$

with $A \cap B = \emptyset$.
We can do segmentation by finding the minimum cut in a graph.

- Efficient algorithms exist for doing this.
• We can do segmentation by finding the **minimum cut** in a graph
  - Efficient algorithms exist for doing this
• Weight of cut is directly proportional to the number of edges in the cut.
Normalized cuts

Write graph as $V$, one cluster as $A$ and the other as $B$

\[ \text{cut}(A,B) = \sum_{u \in A, v \in B} W(u,v), \]

with $A \cap B = \emptyset$

assoc($A,V$) is sum of all edges with one end in $A$.

\[ \text{assoc}(A,V) = \sum_{u \in A, v \in B} W(u,v) \]

\[ \text{Ncut}(A,B) = \frac{\text{cut}(A,B)}{\text{assoc}(A,V)} + \frac{\text{cut}(A,B)}{\text{assoc}(B,V)} \]

J. Shi and J. Malik. **Normalized cuts and image segmentation.** PAMI 2000
Normalized cut

- Let $W$ be the adjacency matrix of the graph
- Let $D$ be the diagonal matrix with diagonal entries $D(i, i) = \Sigma_j W(i, j)$
- Then the normalized cut cost can be written as

$$\frac{y^T(D - W)y}{y^T Dy}$$

where $y$ is an indicator vector whose value should be 1 in the $i$-th position if the $i$-th feature point belongs to $A$ and a negative constant otherwise

J. Shi and J. Malik. Normalized cuts and image segmentation. PAMI 2000
Finding the exact minimum of the normalized cut cost is NP-complete, but if we relax $y$ to take on arbitrary values, then we can minimize the relaxed cost by solving the generalized eigenvalue problem

$$(D - W)y = \lambda Dy$$

The solution $y$ is given by the generalized eigenvector corresponding to the second smallest eigenvalue.

Intuitively, the $i$-th entry of $y$ can be viewed as a “soft” indication of the component membership of the $i$-th feature.

- Can use 0 or median value of the entries as the splitting point (threshold), or find threshold that minimizes the Ncut cost.
1. Given an image or image sequence, set up a weighted graph $G = (V, E)$, and set the weight on the edge connecting two nodes being a measure of the similarity between the two nodes.

2. Solve $(D - W)x = \lambda Dx$ for eigenvectors with the smallest eigenvalues.

3. Use the eigenvector with second smallest eigenvalue to bipartition the graph.

4. Decide if the current partition should be sub-divided, and recursively repartition the segmented parts if necessary.

Applying k-means to Laplacian eigenvectors allows us to find cluster with non-convex boundaries.

Both perform same

Spectral clustering is superior
K-Means vs. Spectral Clustering

- Applying k-means to Laplacian eigenvectors allows us to **find clusters with non-convex boundaries**.
• Applying k-means to Laplacian eigenvectors allows us to find cluster with non-convex boundaries.
Examples

Ng et al., 2001

[Ng et al., 2001]
Some Issues

• Choice of number of clusters $k$
  - Most stable clustering is usually given by the value of $k$ that maximizes the eigengap (difference between consecutive eigenvalues)

$$\Delta_k = |\lambda - \lambda_{k-1}|$$
Some Issues

- Choice of number of clusters $k$
- Choice of similarity
  - Choice of kernel for Gaussian kernels, choice of $\sigma$

Good similarity measure

Poor similarity measure
Some Issues

- Choice of number of clusters $k$
- Choice of similarity
  - Choice of kernel
    - for Gaussian kernels, choice of $\sigma$
- Choice of clustering method
  - $k$-way vs. recursive 2-way
Today

• Clustering
  - Distance functions
  - K-Means
  - Spectral clustering
  - Hierarchical clustering
Hierarchical Clustering

- **Bottom-Up (agglomerative):** Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

- The number of dendrograms with $n$ leaves = $(2n -3)!/[(2(n -2)) (n -2)!]$

<table>
<thead>
<tr>
<th>Number of leafs</th>
<th>Number of possible Dendrograms</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>105</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>10</td>
<td>34,459,425</td>
</tr>
</tbody>
</table>
We begin with a distance matrix which contains the distances between every pair of objects in our dataset.

\[
\begin{array}{cccccc}
0 & 8 & 8 & 7 & 7 \\
8 & 0 & 2 & 4 & 4 \\
8 & 2 & 0 & 3 & 3 \\
7 & 4 & 4 & 0 & 1 \\
7 & 4 & 3 & 1 & 0 \\
\end{array}
\]
Bottom-Up (agglomerative):
Start with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.
Bottom-Up (agglomerative):
Start with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.
**Bottom-Up (agglomerative):**  
Start with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.
Bottom-Up (agglomerative):
Start with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

But how do we compute distances between clusters rather than objects?
Computing distance between clusters: Single Link

- Cluster distance = distance of two closest members in each class

- Potentially long and skinny clusters
Computing distance between clusters: Complete Link

- Cluster distance = distance of two farthest members in each class
- Tight clusters
Computing distance between clusters: Average Link

- Cluster distance = average distance of all pairs

- The most widely used measure
- Robust against noise
Agglomerative Clustering

Good
• Simple to implement, widespread application
• Clusters have adaptive shapes
• Provides a hierarchy of clusters

Bad
• May have imbalanced clusters
• Still have to choose number of clusters or threshold
• Need to use an “ultrametric” to get a meaningful hierarchy
What is a good clustering?

• **Internal criterion**: A good clustering will produce high quality clusters in which:
  - the intra-class (that is, intra-cluster) similarity is high
  - the inter-class similarity is low
  - The measured quality of a clustering depends on both the obj. representation and the similarity measure used

• **External criteria** for clustering quality
  - Quality measured by its ability to discover some or all of the hidden patterns or latent classes in gold standard data
  - Assesses a clustering with respect to ground truth
  - Example:
    - Purity
    - entropy of classes in clusters (or mutual information between classes and clusters)
External Evaluation of Cluster Quality

- Simple measure: purity, the ratio between the dominant class in the cluster and the size of cluster
  - Assume documents with C gold standard classes, while our clustering algorithms produce K clusters, $\omega_1, \omega_2, \ldots, \omega_K$ with $n_i$ members.

\[
Purity(w_i) = \frac{1}{n_i} \max_{j} (n_{ij}) \quad j \in C
\]

- Example

Cluster I: Purity = 1/6 (max(5, 1, 0)) = 5/6
Cluster II: Purity = 1/6 (max(1, 4, 1)) = 4/6
Cluster III: Purity = 1/5 (max(2, 0, 3)) = 3/5
External Evaluation of Cluster Quality

• Let:

$$TC = TC_1 \cup TC_2 \cup \ldots \cup TC_n$$

$$CC = CC_1 \cup CC_2 \cup \ldots \cup CC_m$$

be the target and computed clusterings, respectively.

• $TC = CC =$ original set of data

• Define the following:
  - $a$: number of pairs of items that belong to the same cluster in both $CC$ and $TC$
  - $b$: number of pairs of items that belong to different clusters in both $CC$ and $TC$
  - $c$: number of pairs of items that belong to the same cluster in $CC$ but different clusters in $TC$
  - $d$: number of pairs of items that belong to the same cluster in $TC$ but different clusters in $CC$
External Evaluation of Cluster Quality

\[
P = \frac{a}{a + c}
\]

\[
R = \frac{a}{a + d}
\]

\[
F = \frac{2 \times P \times R}{P + R}
\]

Rand Index

\[
\frac{a+b}{a+b+c+d}
\]

Measure of clustering agreement: how similar are these two ways of partitioning the data?

F-measure

Slide credit: Christophe Giraud-Carrier
External Evaluation of Cluster Quality

\[
\text{Rand Index} = \frac{a+b}{a+b+c+d} \\
\text{Adjusted Rand Index} = \frac{2(ab - cd)}{(a + c)(c + b) + (a + d)(d + b)}
\]

Extension of the Rand index that attempts to account for items that may have been clustered by chance.

Slide credit: Christophe Giraud-Carrier
External Evaluation of Cluster Quality

\[
\text{Entropy}(CC_i) = \sum_{TC_j \in TC} -p(TC_j \mid CC_i) \log p(TC_j \mid CC_i)
\]

\[
\text{AvgEntropy}(CC) = \sum_{i=1}^{m} \frac{|CC_i|}{|CC|} \text{Entropy}(CC_i)
\]

**Average Entropy**

Measure of purity with respect to the target clustering