What is clustering?

- Are there any “grouping” them?
- What is each group?
- How many?
- How to identify them?
What is clustering?

- Clustering: the process of grouping a set of objects into classes of similar objects
  - high intra-class similarity
  - low inter-class similarity
  - It is the commonest form of unsupervised learning

- Unsupervised learning = learning from raw (unlabeled, unannotated, etc) data, as opposed to supervised data where a classification of examples is given

- A common and important task that finds many applications in Science, Engineering, information Science, and other places
  - Group genes that perform the same function
  - Group individuals that has similar political view
  - Categorize documents of similar topics
  - Ideality similar objects from pictures
Examples

- People
- Images
- Language
- species
Issues for clustering

- What is a natural grouping among these objects?
  - Definition of "groupness"

- What makes objects “related”?
  - Definition of "similarity/distance"

- Representation for objects
  - Vector space? Normalization?

- Clustering Algorithms
  - K-means
  - Mixture models

- How many clusters?
  - Fixed a priori?
  - Completely data driven?
    - Avoid “trivial” clusters - too large or small

- Formal foundation and convergence
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.

© Eric Xing @ CMU, 2013
What properties should a distance measure have?

- $D(A,B) = D(B,A)$  
  **Symmetry**

- $D(A,A) = 0$  
  **Constancy of Self-Similarity**

- $D(A,B) = 0$ iff $A = B$  
  **Positivity Separation**

- $D(A,B) \leq D(A,C) + D(B,C)$  
  **Triangular Inequality**
Intuitions behind desirable distance measure properties

- \( D(A,B) = D(B,A) \) \hspace{1cm} \text{Symmetry}
  - Otherwise you could claim "Alex looks like Bob, but Bob looks nothing like Alex"

- \( D(A,A) = 0 \) \hspace{1cm} \text{Constancy of Self-Similarity}
  - Otherwise you could claim "Alex looks more like Bob, than Bob does"

- \( D(A,B) = 0 \) \text{Ilf} A = B \hspace{1cm} \text{Positivity Separation}
  - Otherwise there are objects in your world that are different, but you cannot tell apart.

- \( D(A,B) \leq D(A,C) + D(B,C) \) \hspace{1cm} \text{Triangular Inequality}
  - Otherwise you could claim "Alex is very like Bob, and Alex is very like Carl, but Bob is very unlike Carl"
Distance Measures: Minkowski Metric

- Suppose two objects $x$ and $y$ both have $p$ features
  
  $$x = (x_1, x_2, \cdots, x_p)$$
  $$y = (y_1, y_2, \cdots, y_p)$$

- The Minkowski metric is defined by
  
  $$d(x, y) = \sqrt[p]{\sum_{i=1}^{p} |x_i - y_i|^r}$$

- Most Common Minkowski Metrics
  
  1. $r = 2$ (Euclidean distance)
     
     $$d(x, y) = \sqrt{\sum_{i=1}^{p} |x_i - y_i|^2}$$
  
  2. $r = 1$ (Manhattan distance)
     
     $$d(x, y) = \sum_{i=1}^{p} |x_i - y_i|$$
  
  3. $r = +\infty$ ("sup" distance)
     
     $$d(x, y) = \max_{1 \leq i \leq p} |x_i - y_i|$$
1: Euclidean distance: $\sqrt{4^2 + 3^2} = 5$.
2: Manhattan distance: $4 + 3 = 7$.
3: "sup" distance: $\max\{4,3\} = 4$. 
Hamming distance

- Manhattan distance is called *Hamming distance* when all features are binary.

  - Gene Expression Levels Under 17 Conditions (1-High, 0-Low)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>GeneA</em></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td><em>GeneB</em></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Hamming Distance: \( \#(01) + \#(10) = 4 + 1 = 5 \).
Similarity Measures: Correlation Coefficient

Gene A

Gene B

Gene A

Gene B

Expression Level

Expression Level

Expression Level

Time

Time

Time
Similarity Measures: Correlation Coefficient

- Pearson correlation coefficient

\[
s(x, y) = \frac{\sum_{i=1}^{p} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{p} (x_i - \bar{x})^2 \times \sum_{i=1}^{p} (y_i - \bar{y})^2}}
\]

where \( \bar{x} = \frac{1}{p} \sum_{i=1}^{p} x_i \) and \( \bar{y} = \frac{1}{p} \sum_{i=1}^{p} y_i \).

\[|s(x, y)| \leq 1\]

- Special case: cosine distance

\[
s(x, y) = \frac{\bar{x} \cdot \bar{y}}{||\bar{x}|| \cdot ||\bar{y}||}
\]
Clustering Algorithms

- **Partitional algorithms**
  - Usually start with a random (partial) partitioning
  - Refine it iteratively
    - K means clustering
    - Mixture-Model based clustering

- **Hierarchical algorithms**
  - Bottom-up, agglomerative
  - Top-down, divisive
Partitioning Algorithms

- Partitioning method: Construct a partition of \( n \) objects into a set of \( K \) clusters

- Given: a set of objects and the number \( K \)

- Find: a partition of \( K \) clusters that optimizes the chosen partitioning criterion
  - Globally optimal: exhaustively enumerate all partitions
  - Effective heuristic methods: K-means and K-medoids algorithms
K-means Clustering: Step 1
K-means Clustering: Step 2
K-means Clustering: Step 3
K-means Clustering: Step 4
K-means Clustering: Step 5
K-Means

Algorithm

1. Decide on a value for $k$.
2. Initialize the $k$ cluster centers randomly if necessary.
3. Decide the class memberships of the $N$ objects by assigning them to the nearest cluster centroids (aka the center of gravity or mean)

$$
\vec{\mu}_k = \frac{1}{C_k} \sum_{i \in C_k} \vec{x}_i
$$

4. Re-estimate the $k$ cluster centers, by assuming the memberships found above are correct.
5. If none of the $N$ objects changed membership in the last iteration, exit. Otherwise go to 3.
Problem with K-means
Mixture Models

- A density model $p(x)$ may be multi-modal.
- We may be able to model it as a mixture of uni-modal distributions (e.g., Gaussians).
- Each mode may correspond to a different sub-population (e.g., male and female).
Gaussian Mixture Models (GMMs)

- Consider a mixture of $K$ Gaussian components:

\[
p(x_n | \mu, \Sigma) = \sum_k \pi_k N(x_n | \mu_k, \Sigma_k)
\]

- This model can be used for unsupervised clustering.
  - This model (fit by AutoClass) has been used to discover new kinds of stars in astronomical data, etc.
Learning mixture models

- In fully observed iid settings, the log likelihood decomposes into a sum of local terms.
  \[ \ell_c(\theta; D) = \log p(x, z | \theta) = \log p(z | \theta_z) + \log p(x | z, \theta_x) \]

- With latent variables, all the parameters become coupled together via **marginalization**
  \[ \ell_c(\theta; D) = \log \sum_z p(x, z | \theta) = \log \sum_z p(z | \theta_z) p(x | z, \theta_x) \]

- Are they equally difficult?
MLE for GMM

- If we are doing MLE for completely observed data

- Data log-likelihood

\[
\ell(\theta; D) = \log \prod_n p(z_n, x_n) = \log \prod_n p(z_n | \pi) p(x_n | z_n, \mu, \sigma)
\]

\[
= \sum_n \log \prod_k \pi_k^{z_n} + \sum_n \log \prod_k N(x_n; \mu_k, \sigma_k)^{z_n}
\]

\[
= \sum_n \sum_k z_n \log \pi_k - \sum_n \sum_k z_n \frac{1}{2\sigma^2}(x_n - \mu_k)^2 + C
\]

- MLE

\[
\hat{\pi}_{k,MLE} = \arg \max_{\pi} \ell(\theta; D),
\]

\[
\hat{\mu}_{k,MLE} = \arg \max_{\mu} \ell(\theta; D)
\]

\[
\hat{\sigma}_{k,MLE} = \arg \max_{\sigma} \ell(\theta; D) \quad \Rightarrow \quad \hat{\mu}_{k,MLE} = \frac{\sum_n z_n^k x_n}{\sum_n z_n^k}
\]

- What if we do not know \(z_n\)?

\[
z_n \rightarrow p(z_n^k = 1 | x, \mu^{(i)}, \Sigma^{(i)})
\]
The Expectation-Maximization (EM) Algorithm

- Start:
  - "Guess" the centroid $\mu_k$ and covariance $\Sigma_k$ of each of the $K$ clusters

- Loop
The Expectation-Maximization (EM) Algorithm

- A “soft” k-means

E:

$$\tau_{n}^{k(t)} = \langle z_{n}^{k} \rangle_{q^{(t)}} = p(z_{n}^{k} = 1 \mid x, \mu^{(t)}, \Sigma^{(t)})$$

M:

$$\pi_{k}^{*} = \frac{\sum_{n} \tau_{n}^{k(t)}}{N} = \frac{\left\langle n_{k} \right\rangle}{N}$$

$$\mu_{k}^{(t+1)} = \frac{\sum_{n} \tau_{n}^{k(t)} x_{n}}{\sum_{n} \tau_{n}^{k(t)}}$$

$$\Sigma_{k}^{(t+1)} = \frac{\sum_{n} \tau_{n}^{k(t)} (x_{n} - \mu_{k}^{(t+1)}) (x_{n} - \mu_{k}^{(t+1)})^{T}}{\sum_{n} \tau_{n}^{k(t)}}$$
Compare: K-means

- The EM algorithm for mixtures of Gaussians is like a "soft version" of the K-means algorithm.
- In the K-means “E-step” we do hard assignment:

\[
Z_n^{(t)} = \arg \max_k (x_n - \mu_k^{(t)})^T \Sigma_k^{-1(t)} (x_n - \mu_k^{(t)})
\]

- In the K-means “M-step” we update the means as the weighted sum of the data, but now the weights are 0 or 1:

\[
\mu_k^{(t+1)} = \frac{\sum_n \delta(Z_n^{(t)}, k) x_n}{\sum_n \delta(Z_n^{(t)}, k)}
\]
Theory underlying EM

- What are we doing?
- Recall that according to MLE, we intend to learn the model parameter that would have maximize the likelihood of the data.
- But we do not observe $z$, so computing
  \[
  \ell_c(\theta; D) = \log \sum_z p(x, z | \theta) = \log \sum_z p(z | \theta_z) p(x | z, \theta_x)
  \]
  is difficult!
- What shall we do?
Complete vs. Expected Complete Log Likelihoods

- The complete log likelihood:

\[ \ell(\theta; D) = \log \prod_n p(z_n, x_n) = \log \prod_n p(z_n | \pi) p(x_n | z_n, \mu, \sigma) \]

\[ = \sum \log \prod_k \pi_k^{z_n^k} + \sum \log \prod_k N(x_n; \mu_k, \sigma_k^{z_n^k}) \]

\[ = \sum \sum z_n^k \log \pi_k - \sum \sum z_n^k \frac{1}{2\sigma^2} (x_n - \mu_k)^2 + C \]

- The expected complete log likelihood

\[ \langle \ell_c(\theta; x, z) \rangle = \sum_n \langle \log p(z_n | \pi) \rangle_{p(z|x)} + \sum_n \langle \log p(x_n | z_n, \mu, \Sigma) \rangle_{p(z|x)} \]

\[ = \sum \sum \langle z_n^k \rangle \log \pi_k - \frac{1}{2} \sum_n \sum_k \langle z_n^k \rangle ((x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k) + \log |\Sigma_k| + C \]

- EM optimizes the expected complete log likelihood
Expected Complete Log Likelihood Lower-bounds Complete Log Likelihood

- For any distribution \( q(z) \), define \textit{expected complete log likelihood}:

\[
\langle \ell_c(\theta; x, z) \rangle_q \overset{\text{def}}{=} \sum_z q(z \mid x, \theta) \log p(x, z \mid \theta)
\]

- A deterministic function of \( \theta \)
- Linear in \( \ell_c() \) --- inherit its factorizability
- Does maximizing this surrogate yield a maximizer of the likelihood?

- Jensen's inequality

\[
\ell(\theta; x) = \log p(x \mid \theta) = \log \sum_z p(x, z \mid \theta) = \log \sum_z q(z \mid x) \frac{p(x, z \mid \theta)}{q(z \mid x)} \geq \sum_z q(z \mid x) \log \frac{p(x, z \mid \theta)}{q(z \mid x)} \implies \ell(\theta; x) \geq \langle \ell_c(\theta; x, z) \rangle_q + H_q
\]
Closing notes

- Time complexity
- Convergence
- Seed choice
- Quality of cluster
- How many clusters
Time Complexity

- Computing distance between two objs is $O(m)$ where $m$ is the dimensionality of the vectors.

- Relabeling:
  - K-means: reassigning clusters -- $O(Kn)$ distance computations, or $O(Knm)$.
  - Mixture: Reassigning soft-labeling?

- Re-update clusters:
  - K-means: computing centroids-- each obj gets added once to some centroid: $O(nm)$.
  - Mixture: computing mean/variance?

- Assume these two steps are each done once for $l$ iterations: $O(lKnm)$. 

© Eric Xing @ CMU, 2013
Convergence

- Why should the K-means algorithm ever reach a fixed point?
  - A state in which clusters don’t change.

- K-means is a special case of a general procedure known as the Expectation Maximization (EM) algorithm.
  - Both are known to converge.
  - Number of iterations could be large.

- Goodness measure
  - Sum of squared distances from cluster centroid:
    \[ SD_{K_i} = \sum_{j=1}^{m_k} ||x_{ij} - \mu_i||^2 \]
    \[ SD_K = \sum_{i=1}^{k} SD_{K_i} \]

- Reassignment monotonically decreases SD since each vector is assigned to the closest centroid.
Seed Choice

- Results can vary based on random seed selection.

- Some seeds can result in poor convergence rate, or convergence to sub-optimal clusterings.
  - Select good seeds using a heuristic (e.g., doc least similar to any existing mean)
  - Try out multiple starting points (very important!!!)
  - Initialize with the results of another method.
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 object 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.
  
  \[
  Purity(w_i) = \frac{1}{n_i} \max_j \left( n_{ij} \right) \quad j \in C
  \]

- 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
How Many Clusters?

- Number of clusters $K$ is given
  - Partition $n$ docs into predetermined number of clusters

- Finding the “right” number of clusters is part of the problem
  - Given objs, partition into an “appropriate” number of subsets.
  - E.g., for query results - ideal value of $K$ not known up front - though UI may impose limits.

- Solve an optimization problem: penalize having lots of clusters
  - Application dependent, e.g., compressed summary of search results list.
  - Information theoretic approaches: model-based approach

- Tradeoff between having more clusters (better focus within each cluster) and having too many clusters

- Nonparametric Bayesian Inference