Overview: Clustering

- Supervised vs. unsupervised learning
- Three algorithms
  1. Agglomerative
  2. K-means
  3. COBWEB
- Issues

Supervised vs. unsupervised learning: example

- Do we really need labels?
Supervised vs. unsupervised learning

- Everything so far has been **supervised**
  - Labeled training data
  - In general, "tutor" provides labels and/or feedback
- Clustering is **unsupervised**
  - Unlabeled training data
  - In general, given some info, goal is to learn "something"
- Pluses and minuses
  - Labels can bias the supervised algorithm - data may not actually support the concept
  - Unsupervised is less biased but may return spurious results or miss the concept you wanted

---

Slide CS478 Clustering 3

Clustering

- Definition of **clustering**:
  - Grouping items so that those in the same cluster are more similar to each other than to items in other clusters
- Finding optimal solution is NP-hard
- Number of ways to partition $n$ items into $k$ groups:

$$S_n^{(k)} = \frac{1}{k!} \sum_{i=0}^{k} (-1)^{k-i} \binom{k}{i} i^n$$

- e.g., for 25 items and 5 groups:

$$S_{25}^{(5)} = 2, 436, 684, 974, 110, 751$$

---

Slide CS478 Clustering 4
Clustering Algorithm

- Focus on approximations (usually greedy)
- Many, many variations
- Four main categories:

<table>
<thead>
<tr>
<th></th>
<th>Batch</th>
<th>Incremental</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partitioning</td>
<td>k-means</td>
<td>COP-COBWEB</td>
</tr>
<tr>
<td>Hierarchical</td>
<td>agglomerative</td>
<td>COBWEB</td>
</tr>
</tbody>
</table>

Slide CS478 Clustering 5

Agglomerative algorithm (Ward 63)

- Batch, hierarchical
- Goal: hierarchy with varying levels of generality
- Basic algorithm
  1. Place each instance $D_i$ in its own cluster $C_i$, forming a partition $P_1$ of the input $D$ such that $|P_1| = n$. Let $j = 1$.
  2. While $|P_j| > 1$, find the two closest clusters $C_q, C_r \in P_j$.
     Let $P_{j+1} = P_j \setminus C_q \setminus C_r \cup \{C_q \cup C_r\}$. Increment $j$.

Slide CS478 Clustering 6
Agglomerative algorithm: Example

Slide CS478 Clustering 7

Agglomerative algorithm: Variations

- Different ways to compute the distance between clusters
- Usually based on distance between instances $d(x, y)$
  - Single linkage: $d(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y)$
  - Complete linkage: $d(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y)$
  - Sum of squares:
    $$d(C_i, C_j) = \sum_{x \in \{C_i \cup C_j\}} d(x, centroid(C_i \cup C_j))^2$$

Slide CS478 Clustering 8
**K-means algorithm** (Jancey 66, Lance 67, MacQueen 67)

- Batch, partitioning
- Goal: $k$ disjoint groups that cover the data set
- Basic algorithm
  1. Select $k$ initial cluster centroids, $c_1 \ldots c_k$.
  2. Assign each instance $D_i$ to its nearest centroid $c_j$.
  3. For each cluster, recalculate its centroid based on which instances it contains.
  4. Alternate between (2) and (3) until convergence.

---

**K-means algorithm: Example**

Slide CS478 Clustering 9

---

Slide CS478 Clustering 10
**K-means algorithm: Variations**

- Different ways to generate the initial $k$ centroids
  - Pick first $k$ items in $D$
  - Pick $k$ items randomly from $D$
  - Use point densities to pick top $k$ widely-separated dense regions
- Can use EM, e.g. Autoclass (Bayesian) Cheeseman 88
- Iteratively swap pairs of instances

---

**COBWEB (Fisher 87)**

- Incremental, hierarchical, "conceptual clustering"
- Goal: hierarchical concept that can predict attribute values
- Basic algorithm: For each $D_i \in D$, call COBWEB($D_i$, Root)
  1. If Root is a leaf, add $D_i$ to the leaf and return it.
  2. Otherwise, find the best host child $c$ of Root and do the best of the following:
     (a) **Add**: call COBWEB($D_i$, $c$).
     (b) Create a **New** child containing $D_i$.
     (c) **Merge** two best children to get $c'$ and call COBWEB($D_i$, $c'$).
     (d) **Split** $c$ and call COBWEB($D_i$, Root).

---

**Slide CS478 Clustering 11**

---

**Slide CS478 Clustering 12**
**COBWEB: Example**

<table>
<thead>
<tr>
<th></th>
<th>Number of legs</th>
<th>Body covering</th>
</tr>
</thead>
<tbody>
<tr>
<td>fish</td>
<td>0 scales</td>
<td></td>
</tr>
<tr>
<td>lizard</td>
<td>4 scales</td>
<td></td>
</tr>
<tr>
<td>mouse</td>
<td>4 fur</td>
<td></td>
</tr>
<tr>
<td>rabbit</td>
<td>4 fur</td>
<td></td>
</tr>
<tr>
<td>snake</td>
<td>0 scales</td>
<td></td>
</tr>
<tr>
<td>gator</td>
<td>4 scales</td>
<td></td>
</tr>
</tbody>
</table>

Add fish:

\[
P(C0) = 1.0 \\
P(0 \text{ legs} \mid C0) = 1.0 \\
P(\text{scales} \mid C0) = 1.0 \\
\ldots
\]

**COBWEB: Example**

Add lizard:

\[
P(C0) = 0.5 \\
P(0 \text{ legs} \mid C0) = 1.0 \\
P(\text{scales} \mid C0) = 1.0 \\
\ldots
\]

\[
P(C1) = 0.5 \\
P(4 \text{ legs} \mid C1) = 1.0 \\
P(\text{scales} \mid C1) = 1.0 \\
\ldots
\]

\[
CU = 0.0 \\
CU = 0.25
\]

- Category utility (CU) for c classes

\[
CU = \frac{\sum_{k=1}^{c} P(C_k) \left[ \sum_i \sum_j P(A_i = V_{ij} \mid C_k)^2 \right] - \sum_i \sum_j P(A_i = V_{ij})^2}{c}
\]
COBWEB: Example

Add mouse:

ADD to C0

P(C0) = 0.67
P(0 legs | C0) = 0.5
P(4 legs | C0) = 0.5
P(scales | C0) = 0.5
P(fur | C0) = 0.5
...

P(C1) = 0.33
P(4 legs | C1) = 1.0
P(scales | C1) = 0.5
P(fur | C1) = 0.5
...

CU = 0.11

ADD to C1

P(C0) = 0.33
P(0 legs | C0) = 1.0
P(4 legs | C0) = 1.0
P(scales | C0) = 1.0
...

P(C1) = 0.67
P(4 legs | C1) = 1.0
P(scales | C1) = 0.5
P(fur | C1) = 0.5
...

CU = 0.28

Summary of CU values
ADD to C0: 0.11
ADD to C1: 0.28
NEW: 0.30

Slide CS478 Clustering 15

COBWEB: Example

NEW

Add mouse:

fish

P(C0) = 0.33
P(0 legs | C0) = 1.0
P(scales | C0) = 1.0
...

P(C1) = 0.33
P(4 legs | C1) = 1.0
P(scales | C1) = 1.0
...

lizard

P(C2) = 0.33
P(4 legs | C2) = 1.0
P(fur | C2) = 1.0
...

mouse

CU = 0.30

Summary of CU values
ADD to C0: 0.11
ADD to C1: 0.28
NEW: 0.30

Slide CS478 Clustering 16
## COBWEB: Example

### Add rabbit:

<table>
<thead>
<tr>
<th>Summary of CU values</th>
<th>ADD to C0: 0.13</th>
<th>ADD to C1: 0.21</th>
<th>ADD to C2: 0.29</th>
<th>NEW: 0.22</th>
<th>MERGE C1,C2: 0.27</th>
</tr>
</thead>
</table>

### Add snake:

<table>
<thead>
<tr>
<th>Summary of CU values</th>
<th>ADD to C0: 0.32</th>
<th>ADD to C1: 0.25</th>
<th>ADD to C2: 0.14</th>
<th>NEW: 0.24</th>
<th>MERGE C0,C1: 0.35</th>
</tr>
</thead>
</table>

### Slide CS478 Clustering 17

### Before snake:

- **P(C0) = 0.25**
  - P(0 legs | C0) = 1.0
  - P(scales | C0) = 1.0
- **P(C1) = 0.25**
  - P(4 legs | C1) = 1.0
  - P(scales | C1) = 1.0
- **P(C2) = 0.50**
  - P(4 legs | C2) = 1.0
  - P(fur | C2) = 1.0

### After snake:

- **P(C0) = 0.60**
  - P(0 legs | C0) = 1.0
  - P(scales | C0) = 1.0
- **P(C1) = 0.40**
  - P(0 legs | C1) = 1.0
  - P(scales | C1) = 1.0
- **P(C2) = 0.33**
  - P(4 legs | C2) = 1.0
  - P(fur | C2) = 1.0

### CU = 0.35

### Slide CS478 Clustering 18
**Issues**

- Evaluation is difficult for unsupervised learning!
  - Using labels only evaluates how well the algorithm does on finding that specific concept.
  - In any real application, labels will not be known.

- For partitioning, how do you choose the right $k$ (number of clusters)?
  - Possibly many distinct meaningful answers.
  - For example, cluster a deck of cards into $x$ groups.