Attribute-oriented analysis

1 Generality and specificity

1.1 Representing tuples as sets

- Let $X, Y$ be tuples, i.e. $X = \langle x_1, x_2, ..., x_n \rangle, Y = \langle y_1, y_2, ..., y_n \rangle$.
- Assume that the attributes are $A_1, A_2, ..., A_n$.
- Then we can represent tuples as sets of attribute-value pairs: $X = \{A_1 = x_1, A_2 = x_2, ..., A_n = x_n\}$, $Y = \{A_1 = y_1, A_2 = y_2, ..., A_n = y_n\}$.

1.2 Generality ordering with different attribute types

- **Nominal attributes.** $X$ is more general than $Y$ (or $X$ covers, subsumes $Y$), if $X \subseteq Y$. Conversely, $Y$ is more specific than $X$ (or $Y$ is covered, subsumed by $X$).
- **Structured attributes** (attributes forming a concept hierarchy). $X$ is more general than $Y$ (or $X$ covers, subsumes $Y$), if $y_i$ is a successor of $x_i$ in the concept hierarchy of $A_i$, for $i = 1, ..., n$.
- Converting nominal attributes into structured. Assume $A$ is a nominal attribute with values $v_1, v_2, ..., v_n$. Then we can create a two-level concept hierarchy with leaves $v_1, v_2, ..., v_n$ and a root label that allows all possible values for $A$ ($v_1, v_2, ..., v_n$), e.g. $ALL$ (as used in the data cube).
2 Attribute generalization

• Nominal attributes: Dropping condition. Removing an attribute-value pair from $X$, thus obtaining a subset of $X$. Similar to dicing (selecting a subset of values) in the data cube.

• Structured attributes: Climbing up concept hierarchy. Replacing a value in an attribute value pair with a more general one. Similar to roll-up in the data cube.
3 Example

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>weak</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>strong</td>
<td>no</td>
</tr>
<tr>
<td>3</td>
<td>overcast</td>
<td>hot</td>
<td>high</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>rain</td>
<td>mild</td>
<td>high</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>5</td>
<td>rain</td>
<td>cool</td>
<td>normal</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>6</td>
<td>rain</td>
<td>cool</td>
<td>normal</td>
<td>strong</td>
<td>no</td>
</tr>
<tr>
<td>7</td>
<td>overcast</td>
<td>cool</td>
<td>normal</td>
<td>strong</td>
<td>yes</td>
</tr>
<tr>
<td>8</td>
<td>sunny</td>
<td>mild</td>
<td>high</td>
<td>weak</td>
<td>no</td>
</tr>
<tr>
<td>9</td>
<td>sunny</td>
<td>cool</td>
<td>normal</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>10</td>
<td>rain</td>
<td>mild</td>
<td>normal</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>11</td>
<td>sunny</td>
<td>mild</td>
<td>normal</td>
<td>strong</td>
<td>yes</td>
</tr>
<tr>
<td>12</td>
<td>overcast</td>
<td>mild</td>
<td>high</td>
<td>strong</td>
<td>yes</td>
</tr>
<tr>
<td>13</td>
<td>overcast</td>
<td>hot</td>
<td>normal</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>14</td>
<td>rain</td>
<td>mild</td>
<td>high</td>
<td>strong</td>
<td>no</td>
</tr>
</tbody>
</table>

3.1 Set representation

\(X_1 = \{x_1 = \text{sunny}, x_2 = \text{hot}, x_3 = \text{high}, x_4 = \text{weak}, y = \text{no}\}\)

\(X_2 = \{x_1 = \text{sunny}, x_2 = \text{hot}, x_3 = \text{high}, x_4 = \text{strong}, y = \text{no}\}\)

\(X_3 = \{x_1 = \text{overcast}, x_2 = \text{hot}, x_3 = \text{high}, x_4 = \text{weak}, y = \text{yes}\}\)

3.2 Generalization

\(Y_1 = \{x_2 = \text{hot}, x_3 = \text{high}, x_4 = \text{weak}\}\) (\(X_1\) with first and last attributes dropped).

\(Y_1\) is more general than (covers) both \(X_1\) and \(X_3\), because \(Y_1 \subseteq X_1\) and \(Y_1 \subseteq X_3\).

We may create a classification rule IF \(Y_1\) THEN \(y = \text{no}\), that has coverage 2 (two tuples covered by \(Y_1\)) and accuracy 1/2. Note that the notion of coverage here is different from the support for the association rules.

The most general tuple is \(\top = \{\}\) (covers all 14 tuples). By adding attribute-value pairs we may specialize it. For example, \(\{x_1 = \text{overcast}\}\) covers 4 tuples (\(X_3, X_7, X_{12}, X_{13}\)). What is the accuracy of IF \(\{x_1 = \text{overcast}\}\) THEN \(y = \text{yes}\)?
4 Attribute relevance

4.1 Attribute selection

Searching the lattice of subsets of the set of attributes (similar to searching the lattice of cuboids).

4.2 Selection criterion

Find a subset of attributes that is most likely to describe/predict the class best.

- Filtering: scheme-independent attribute selection.
  - Minimal set of attributes that separate all tuples (class-independent).
    Problem: ID attribute (no possibility to generalize).
  - Minimal set of attributes that preserve the class distribution: instance-based methods and entropy-based methods.

- Scheme-specific methods.

4.3 Instance-based attribute selection

- Similarity measure (distance). For example:
  - Euclidean distance for numeric attributes: $D(X, Y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2}$
  - Number of differences for nominal attributes: $D(X, Y) = \sum_1^n d(x_i, y_i)$, where $d(x_i, y_i) = 0$ if $x_i = y_i$ and 1 otherwise.
  - Normalization required for mixed (numeric and nominal).

- Similarity-based attribute selection:
– For each tuple find the nearest neighbors (the closest tuples according to the distance measure) of the same and different classes – ”near hits” and ”near misses”.

– If a near hit has a different value for a certain attribute then that attribute appears to be irrelevant and its weight should be decreased.

– For near misses, the attributes with different values are relevant and their weights should be increased.

– Algorithm: Start with equal weights for all attributes and do the weight adjustment, as explained above. This allows ordering attributes by relevance and selecting the best subset of attributes.

• Example (weather data – Section 3, this chapter):

  – The nearest neighbors of $X_1$ in its class ”no” (near hits) are $X_2$ and $X_8$ (ignoring the class $y$ we have: $D(X_1, X_2) = 1$, $D(X_1, X_6) = 4$, $D(X_1, X_8) = 1$, $D(X_1, X_{14}) = 3$).

  – Attribute $x_4$ (wind) has different values in $X_1$ and $X_2$, so we decrease its relevance.

  – Attribute $x_2$ (temperature) has different values in $X_1$ and $X_8$, so we decrease its relevance too.

  – The nearest neighbor of $X_1$ in the opposite class ”yes” (near miss) is $X_3$ ($D(X_1, X_3) = 1$).

  – Attribute $x_1$ (outlook) has different values in $X_1$ and $X_3$, so we increase its relevance.
4.4 Entropy-based attribute selection

- Let $S$ be a set of tuples from $m$ classes – $C_1, C_2, ..., C_m$. Then the
  number of tuples in $S$ is $|S| = |S_1| + |S_2| + ... + |S_m|$, where $S_i$ is
  the set of tuples from class $C_i$.

- The entropy of the class distribution in $S$ (or the average information
  needed to classify an arbitrary tuple) is
  
  $$I(S) = -P(C_1) \times \log_2 P(C_1) - P(C_2) \times \log_2 P(C_2) - ... - P(C_n) \times \log_2 P(C_n),$$

  where $P(C_i) = \frac{|S_i|}{|S|}$.

- Assume that attribute $A$ splits $S$ into $k$ subsets – $A_1, A_2, ..., A_k$
  (each $A_i$ having the same value for $A$).

- Then (similarly to the info function used for entropy-based discretization in
  Chapter 3), the information in the split, based on the values of $A$ is
  
  $$I(A) = \frac{|A_1|}{|S|} \times I(A_1) + \frac{|A_2|}{|S|} \times I(A_2) + ... + \frac{|A_k|}{|S|} \times I(A_k))$$

- Then, the information gain is
  
  $$\text{gain}(A) = I(S) - I(A)$$

- The most relevant attribute (the one with the highest discriminant power) is
  the attribute with maximal information gain.

- What about the tuple ID attribute? $I(A) = ?$, Is it relevant?

- Example (weather data – Section 3, this chapter):
  
  $$I(S) = -P(\text{yes}) \times \log_2 P(\text{yes}) - P(\text{no}) \times \log_2 P(\text{no}) = \frac{9}{14} \times \log_2 \frac{9}{14} - \frac{5}{14} \times \log_2 \frac{5}{14}.$$
\[ A = \text{outlook}, \ A_1 = \{1, 2, 8, 9, 11\} \text{ (sunny)}, \ A_2 = \{3, 7, 12, 13\} \text{ (overcast)}, \ A_3 = \{4, 5, 6, 10, 14\} \text{ (rainy)}. \]

\[ I(\text{outlook}) = \frac{5}{14} \times I(A_1) + \frac{4}{14} \times I(A_2) + \frac{5}{14} \times I(A_3) \]

\[ I(A_1) = I(\{\text{no, no, no, yes, yes}\}) = -\frac{3}{5} \times \log_2 \frac{3}{5} - \frac{2}{5} \times \log_2 \frac{2}{5} \]

\[ I(A_2) = I(\{\text{yes, yes, yes, yes}\}) = 0 \]

\[ I(A_3) = I(\{\text{yes, yes, no, yes, no}\}) = -\frac{2}{5} \times \log_2 \frac{3}{5} - \frac{3}{5} \times \log_2 \frac{2}{5} \]

### 4.5 Class characterization and comparison

- Let \( X \) be a generalized tuple (rule) from class \( C_i \) in a data set \( S \) with \( n \) classes – \( C_1, C_2, ..., C_n \). Assume \( X \) covers \( M_i \) tuples from class \( C_i \) and a total of \( K_i \) tuples from \( S \).

- \( T(X) = \frac{M_i}{|C_i|} \)

- \( D(X) = \frac{M_i}{K_i} \)

- \( T(X) \) is a measure of the characterization power of \( X \). If \( T(X) < 1 \) (\( X \) does not cover all tuples in \( C_i \)), we need more generalized tuples to describe \( C_i \) (the new tuples are added to \( X \) as disjuncts). If \( T(X) \) is too small then we need to many disjuncts (overspecialization).

- \( D(X) \) is a measure of the discriminant power of \( X \). If \( D(X) = 1 \), \( X \) is a good rule (100% accurate). If \( D(X) < 1 \) (\( X \) covers tuples from contrasting classes) then \( X \) has to be specialized (we have overgeneralization).

- Example (weather data – Section 3, this chapter): \( X = \{\text{Day} = 3\}, \ T(X) = ?, \ D(X) = ? \)
4.6 Statistical measures

• Measuring central tendency
  
  – *Arithmetic mean* (average) of all values of an attribute:

\[
\mu = \frac{1}{n} \sum_{i=1}^{n} x_i
\]

  – *Median*: the middle value in an ordered sequence.

• Measuring *dispersion*: variance (\(\sigma\)) and standard deviation (\(\sigma^2\))

\[
\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2
\]