15-826: Multimedia Databases and Data Mining

Data Mining - AI reminders
C. Faloutsos

Outline

Goal: ‘Find similar / interesting things’
• Intro to DB
• Indexing - similarity search
• Data Mining

Data Mining - Detailed outline

• Statistics

• AI - decision trees
  – Problem
  – Approach
  – Conclusions

• DB

Decision Trees

• Problem: Classification - i.e.,
  • given a training set (N tuples, with M attributes, plus a label attribute)
  • find rules, to predict the label for newcomers

Pictorially:

Decision trees

<table>
<thead>
<tr>
<th>Age</th>
<th>Chol-level</th>
<th>Gender</th>
<th>…</th>
<th>CLASS-ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>150</td>
<td>M</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• issues:
  – missing values
  – noise
  – ‘rare’ events
Decision trees

• types of attributes
  – numerical (= continuous) - eg: ‘salary’
  – ordinal (= integer) - eg.: ‘# of children’
  – nominal (= categorical) - eg.: ‘car-type’

Decision trees

• Pictorially, we have

<table>
<thead>
<tr>
<th>num. attr#1 (eg., ‘age’)</th>
<th>num. attr#2 (eg., chol-level)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

• and we want to label ‘?’

<table>
<thead>
<tr>
<th>num. attr#1 (eg., ‘age’)</th>
<th>num. attr#2 (eg., chol-level)</th>
</tr>
</thead>
<tbody>
<tr>
<td>age&lt;50</td>
<td>chol. &lt;40</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Data Mining - Detailed outline

• Statistics
  • AI - decision trees
    – Problem
    – Approach
    – Conclusions
  • DB
Decision trees

• Typically, two steps:
  – tree building
  – tree pruning (for over-training/over-fitting)

Tree building

• How?

  num. attr#1 (eg., ‘age’)
  num. attr#2 (eg., chol-level)

• Q1: how to introduce splits along attribute A
  • A: Partition, recursively - pseudocode:
    Partition (Dataset S)
    if all points in S have same label
    then return
    evaluate splits along each attribute A
    pick best split, to divide S into S1 and S2
    Partition(S1); Partition(S2)

• Q2: how to evaluate a split?
  • for num. attributes:
    • binary split, or
    • multiple split
  • for categorical attributes:
    • compute all subsets (expensive!), or
    • use a greedy algo
Tree building

- Q1: how to introduce splits along attribute $A_i$
- Q2: how to evaluate a split?
- $A$: by how close to uniform each subset is, i.e., we need a measure of uniformity:

\[
\text{entropy: } H(p, p^+) + (1-p)\text{'gini' index: } 1-p
\]

(How about multiple labels?)

Tree building

Intuition:
- entropy: #bits to encode the class label
- gini: classification error, if we randomly guess '+' with prob. $p_+$

Thus, we choose the split that reduces entropy/classification-error the most: Eg.:

```
num. attr#1 (eg., 'age')
```

```
num. attr#2 (eg., chol-level)
```

Any other measure?
Tree building

- Before split: we need
  \((n_1 + n_2) \cdot H(p_1, p_2) = (7+6) \cdot H(7/13, 6/13)\)
  bits total, to encode all the class labels
- After the split we need:
  0 bits for the first half and
  \((2+6) \cdot H(2/8, 6/8)\) bits for the second half

Data Mining - Detailed outline

- Statistics
- AI - decision trees
  - Problem
  - Approach
  - Tree building
    - Tree pruning
  - Conclusions
- DB

Tree pruning

- What for?

- Q: How to do it?

- A1: use a ‘training’ and a ‘testing’ set -
  prune nodes that improve classification in the ‘testing’ set. (Drawbacks?)

- A2: or, rely on MDL (= Minimum Description Language) - in detail:
Tree pruning

- envision the problem as compression (of what?)
- and try to min. the # bits to compress
  - (a) the class labels AND
  - (b) the representation of the decision tree

(MDL)

- a brilliant idea - e.g.: best \( n \)-degree polynomial to compress these points:
- the one that minimizes (sum of errors + \( n \))

Conclusions

- Classification through trees
- Building phase - splitting policies
- Pruning phase (to avoid over-fitting)
- Observation: classification is subtly related to compression

Reference