Aggregation Based Feature Invention and Relational Concept Classes

(Claudia Perlich & Foster Provost)

Relational Learning

• Expressive

• Background Knowledge can be incorporated easily

• Aggregation
Predictive Relational Learning

- $M: (t, RDB) \rightarrow y$

$$y = \varphi(t, \psi(RDB)) + \varepsilon$$

- Complexity of relational concept
  1. Complexity of relationships
  2. Complexity of Aggregation Function
  3. Complexity of the function

**Figure 1: Transaction database**
Relational Concept Classes

• Propositional
  – Features can be concatenated
  – No aggregation
  – Example – One customer table and other
demographic table

• Independent Attributes
  – 1 to n relationship requires simple aggregation
  – Mapping from a bag of zero or more attributes to a
categorical or numeric value
  – Ex Sum, Average for numeric values
  – Ex Mode for categorical attributes

Relational Concept Classes - Contd

• Dependent Attributes within one table
  – Multi-dimensional Aggregation
  – Number of products bought on Dec 22\textsuperscript{nd} (conditioned on Date)

• Dependent Attributes across tables
  – More than one bag of objects of different type
  – Amount spent on items returned at a later date
  – Needs info from more than 1 table

• Global graph features
  – Transitive closure over a set possible joins
  – Customer Reputation
Methods for Relational Aggregation

- First Order Logic - ILP
- Simple Numeric Aggregation
  - Simple Aggregation operators – Mean, Min, Max, Mode
  - Cannot express above level 2
- Set Distances
  - Relational Distance metric & KNN
  - Calculates the minimum distance of all possible pairs of objects
  - Distance – Sum of squared distance (numeric values) or edit distance (categorical values)
  - Assumes attribute independence

Transformation Based Learning

![Diagram of Transformation Based Learning](image)
Value Distributions

- **Value Order**: List of (Value: Index) pairs
  - Ex (watch:1, book:2, CD:3, DVD:4)
- **Case Vector**
  - Ex \{book, CD, CD, book, DVD, book\} for case t
  - \(CV^t_{\text{Products.ProductType}} = (0,3,2,1)\)
- **Reference Vector** – based on a condition c
  - Has at position i the sum of values \(CV[i]\) for all cases t for which c was true
  - Ex Number of CDs
- **Variance Vector** – \((CV[i])^2 / (N_c - 1)\)
  where \(N_c\) – number of cases where c is true

**Aggregation = Density Estimation**

Target Dependent Individual Values

<table>
<thead>
<tr>
<th>RV Class +ve</th>
<th>RV Class -ve</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Book</strong></td>
<td><strong>Book</strong></td>
</tr>
<tr>
<td>.01</td>
<td>.21</td>
</tr>
<tr>
<td><strong>CD</strong></td>
<td><strong>CD</strong></td>
</tr>
<tr>
<td>.31</td>
<td>.36</td>
</tr>
<tr>
<td><strong>DVD</strong></td>
<td><strong>DVD</strong></td>
</tr>
<tr>
<td>.35</td>
<td>.28</td>
</tr>
<tr>
<td><strong>VCR</strong></td>
<td><strong>VCR</strong></td>
</tr>
<tr>
<td>.33</td>
<td>.15</td>
</tr>
</tbody>
</table>

- Most common (MC) - CD
- Most common positive (MOP): DVD
- Most common Negative (MON): CD
- Most Discriminative (MOD): Book
Feature Complexity

1. No Relational Features
2. Unconditional Features MC, Count
3. Class Conditional Features – MOP, MON
4. Discriminative Class Conditional Features – MOD, MOM

Vector Distances

\[
\begin{align*}
EDD &= ED(RV^0, CV) - ED(RV^1, CV) \\
EUD &= EU(RV^0, CV) - EU(RV^1, CV) \\
COSD &= COS(RV^0, CV) - COS(RV^1, CV) \\
MAD &= MA(RV^0, CV) - MA(RV^1, CV)
\end{align*}
\]

<table>
<thead>
<tr>
<th>Reference Vector</th>
<th>Euclidean</th>
<th>Edit</th>
<th>Cosine</th>
<th>Mahalanobis</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>EU</td>
<td>ED</td>
<td>COS</td>
<td>MA</td>
</tr>
<tr>
<td>Positive</td>
<td>EUP</td>
<td>EDP</td>
<td>COSP</td>
<td>MAP</td>
</tr>
<tr>
<td>Negative</td>
<td>EUN</td>
<td>EDN</td>
<td>COSN</td>
<td>MAN</td>
</tr>
<tr>
<td>Positive vs.</td>
<td>EUD</td>
<td>EDD</td>
<td>COSD</td>
<td>MAD</td>
</tr>
<tr>
<td>Negative</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Domain: Initial Public Offerings

- IPO(Date, Size, Price, Ticker, Exchange, SIC, Runup)
- HEAD(Ticker, Bank)
- UNDER(Ticker, Bank)
- IND(SIC, Ind2)
- IND2(Ind2, Ind)

- **Goal:** To predict whether the offer was made on the NASDAQ exchange

Implementation details

- Four approaches were tested
  - ILP
  - Logic Based feature construction
  - Selection of specific individual values
  - Target dependent vector aggregation

- Two features were constructed
  - One for (n:1) joins
  - Other for autocorrelation
Details (Contd)

- Exploration – To find related objects
  - Uses BFS
  - Stopping criterion – maximum number of chains
- Feature Selection – Weighted Sampling to select a subset of 10 features
- Model Estimation – Uses C4.5 to learn a tree
  - No change in results if logistic regression was used
- Logic Based Feature construction – Uses ILP to learn FOL clauses and append the binary features
- ILP – Only class labels

Aggregation approaches

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>No Feature Construction</td>
</tr>
<tr>
<td>MOC</td>
<td>Unconditional features – Counts in IPO table</td>
</tr>
<tr>
<td>VD</td>
<td></td>
</tr>
<tr>
<td>MVD</td>
<td></td>
</tr>
<tr>
<td>MOP</td>
<td>Class Conditional Features – Most positive and Negative categoricals and vector distances</td>
</tr>
<tr>
<td>MON</td>
<td></td>
</tr>
<tr>
<td>VDPN</td>
<td></td>
</tr>
<tr>
<td>MOD</td>
<td>Discriminative Features – Most common categoricals and vector distances</td>
</tr>
<tr>
<td>MOM</td>
<td></td>
</tr>
<tr>
<td>MVDD</td>
<td></td>
</tr>
</tbody>
</table>
As complexity increases, performance increases
As training size increases, performance increases
Conclusions

- Expressive power of models combined with aggregation
- Distance metric
- Complex aggregations can reduce explorations
- Focusses only up to level 2 of the hierarchy