Learning Probabilistic Relational Models

Getoor, Friedman, Koller, Pfeffer

Probabilistic Relational Models

Course.Instructor is foreign key for Professor relation
Registration.Course is foreign key for Course
Registration.Student is foreign key for Student
Corresponding Database

<table>
<thead>
<tr>
<th>Professor</th>
<th>Popularity</th>
<th>Teaching-Ability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gump</td>
<td>high</td>
<td>medium</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course</th>
<th>Professor</th>
<th>Difficulty</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phil101</td>
<td>Gump</td>
<td>low</td>
<td>high</td>
</tr>
<tr>
<td>Com301</td>
<td>Gump</td>
<td>high</td>
<td>medium</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Registration</th>
<th>Course</th>
<th>Student</th>
<th>Grade</th>
<th>Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reg123</td>
<td>Com301</td>
<td>Gomer Pyle</td>
<td>C</td>
<td>1</td>
</tr>
<tr>
<td>Reg333</td>
<td>Phil101</td>
<td>Jane Doe</td>
<td>A</td>
<td>3</td>
</tr>
<tr>
<td>Reg135</td>
<td>Com301</td>
<td>Jane Doe</td>
<td>A</td>
<td>2</td>
</tr>
</tbody>
</table>

Relational Schema

- Set of classes \( X = \{X_1, \ldots, X_n\} \) (equivalent to relational tables)
- Each class has
  - descriptive attributes \( A(X) \)
    - \( A(\text{Student}) = \{\text{Intelligence, GPA}\} \)
    - JaneDoe.Intelligence
  - reference slots (foreign keys that point to other relations): \( R(X) \)
    - \( R(\text{Registration}) = (\text{Student, Course}) \)
    - Reg333.Student = JaneDoe
    - Reg333.Course = Phil101
  - inverse reference slots:
    - JaneDoe.RegisteredIn = {Reg333, Reg334}
      - Constructed automatically
Slot Chains
(path expressions)

• Student.registered-in.Course.Instructor
  = bag of instructors of the courses that the student is registered in
  – bag is a set with multiple occurrences allowed
  – JaneDoe.registered-in.Course.Instructor = {Gump,Gump}

• Aggregations: Mean, Average, Mode
  – AVG(Student.registered-in.Grade)
    • Average grade of student
  – MODE(Student.registered-in.Course.Instructor)
    • Professor from whom student has taken the most courses

PRM Schema = Relational Schema + Probabilistic Parents

• Each attribute has a set of path expressions describing the parents of that attribute
  – parents(Student.gpa) = {AVG(Student.registered-in.Grade)}
  – parents(Registration.satisfaction) =
    {Registration.Course.Professor.TeachingAbility, Registration.Grade}
  – parents(Registration.grade) = {Registration.Student.Intelligence, Registration.Course.Difficulty}
  – parents(Professor.Popularity) = {Professor.TeachingAbility}
  – parents(Course.rating) =
    {AVG(Course.Registrations.Satisfaction)}
Visualizing the PRM Schema

Probabilistic Relational Model

1. Relational Schema
2. Specification of the parents of each descriptive attribute (in terms of path expressions)
3. Conditional Probability Distribution for each attribute in each class
   - Conditional probability table:
     \[ P(\text{attribute} \mid \text{parents(attribute)}) \]
   - Parametric model:
     \[ P(\text{attribute} \mid \text{parents(attribute)}) = F(\text{attribute}, \text{parents(attribute)}; \theta) \text{ for some parameters } \theta. \]
Instantiating the PRM on a database

Redrawn to show DAG
Aggregations

• We must introduce deterministic intermediate nodes to represent the aggregated value

Example Inferences
useful for tenure and letters of reference

• Observe Registration.Grade (and Student.GPA), Registration.Satisfaction (and Course.Rating), and Professor.Popularity

• Infer Student.Intelligence and Professor.TeachingAbility

• \(P(\text{Gump.TeachingAbility}, \text{Pyle.Intelligence}, \text{Doe.Intelligence} | \ldots)\)
Example Inference (3)

- Example: We might observe that Pyle has a GPA of 4.0. This could be explained either by Pyle.Intelligence or by Course.Difficulty for all of the courses that he took.
- The grades of other students in the same classes that Pyle took can tell us Course.Difficulty, which in turn can help us explain away the 4.0 GPA (e.g., because Pyle took only easy courses).
- This is a form of relational inference! We could not figure it out only from looking at Pyle’s courses and grades.
Example Inference (4)

\[ P(P.I | \ldots) = \sum_{C301.D} \sum_{D.I} \sum_{P101.D} P(R123.G | P.I, C301.D) \cdot P(R135.G | D.I, C301.D) \cdot P(R333.G | D.I, P101.D) \cdot P(P.I) \cdot P(C301.D) \cdot P(D.I) \cdot P(P101.D) \]

Example Inference (5)

\[ P(P.I | \ldots) = \sum_{C301.D} \sum_{D.I} \sum_{P101.D} P(P.I, C301.D) \cdot P(P.I, C301.D) \cdot P(D.I) \cdot \sum_{P101.D} P[D.I, P101.D] \cdot P(D.I) \cdot P(D.I) \cdot P(P101.D) \]
Can we be sure that the instantiated PRM gives a DAG?

- Case 1: Check at the skeleton level

The graph is a DAG at the skeleton level.
Case 2: Skeleton graph contains cycles, but instantiated graph does not

Blood type depends on chromosomes inherited from parents

parents(Person.M-chromosome) =
{Person.Mother.M-chromosome, Person.Mother.P-chromosome}

Case 2: Skeleton graph contains cycles, but instantiated graph does not

parents(Person.M-chromosome) =
{Person.Mother.M-chromosome, Person.Mother.P-chromosome}
PRM Semantics: PRM Skeleton

- Take database: keep Reference attributes, but replace all Descriptive attributes by random variables
- PRM defines the joint distribution of these random variables

PRM Skeleton:

??? denotes random variable

<table>
<thead>
<tr>
<th>Professor</th>
<th>Popularity</th>
<th>Teaching-Ability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gump</td>
<td>???</td>
<td>???</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Student</th>
<th>Intelligence</th>
<th>GPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gomer Pyle</td>
<td>???</td>
<td>???</td>
</tr>
<tr>
<td>Jane Doe</td>
<td>???</td>
<td>???</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course</th>
<th>Professor</th>
<th>Difficulty</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phil101</td>
<td>Gump</td>
<td>???</td>
<td>???</td>
</tr>
<tr>
<td>Com301</td>
<td>Gump</td>
<td>???</td>
<td>???</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Registration</th>
<th>Course</th>
<th>Student</th>
<th>Grade</th>
<th>Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reg123</td>
<td>Com301</td>
<td>Gomer Pyle</td>
<td>???</td>
<td>???</td>
</tr>
<tr>
<td>Reg333</td>
<td>Phil101</td>
<td>Jane Doe</td>
<td>???</td>
<td>???</td>
</tr>
<tr>
<td>Reg135</td>
<td>Com301</td>
<td>Jane Doe</td>
<td>???</td>
<td>???</td>
</tr>
</tbody>
</table>
PRM Semantics (2)

- The PRM does not provide a probabilistic model over the reference attributes (i.e., over the “link structure”) of the database
- The PRM does not provide a model of all possible databases involving these relations. It does not model, for example, the number and nature of the courses that a student takes or the number of classes that a professor teaches.

Learning

- Known Skeleton, Fully Observed
  - Constrain corresponding CPT’s to have the same parameters

\[
P(\text{Reg. Grade} = A | \text{Course.Diff} = \text{high}, \text{Student.Int} = \text{low}) = \frac{N(\text{Reg. Grade} = A, \text{Course.Diff} = \text{High}, \text{Student.Int} = \text{low})}{N(\text{Course.Diff} = \text{high}, \text{Student.Int} = \text{low})}
\]
Learning the Structure

• Case 1: We know how individual objects are connected and we just need to learn the parents of each attribute.
• Case 2: We need to learn how objects are connected as well as learning the parents of each attribute. This is the subject of our next paper.

Case 1: Learning the parents of each attribute

• Search in the space of path expressions and aggregators
  – infinite space!
  – impose some complexity limits?
Application: Tuberculosis

Application: Banking