Intelligent Systems (AI–2)

Computer Science cpsc422, Lecture 33

Nov, 29, 2017

Slide source: from David Page (MIT) (which were from From Lise Getoor, Nir Friedman, Daphne Koller, and Avi Pfeffer) and from Lise Getoor
Lecture Overview

• Recap Motivation and Representation for Probabilistic Relational Models (PRMs)
  • Full Relational Schema and its Instances
  • Relational Skeleton and its Completion Instances
• Probabilistic Model of PRMs
  • Dependency Structure
  • Parameters
How PRMs extend BNs?

1. PRMs conceptually extend BNs to allow the specification of a probability model for classes of objects rather than a fixed set of simple attributes.

2. PRMs also allow properties of an entity to depend probabilistically on properties of other related entities.
Mapping PRMs from Relational Models

• The representation of PRMs is a direct mapping from that of relational databases

• A relational model consists of a set of classes $X_1, \ldots, X_n$ and a set of relations $R_1, \ldots, R_m$, where each relation $R_i$ is typed
University Domain Example - Full Relational Schema

- Primary keys are indicated by a blue rectangle.
- Indicating many-to-many relationship.
- Dashed lines indicate the types of objects referenced.
- Underlined attributes are reference slots of the class.

Professor
- Name
- Popularity
- Teaching-Ability

Student
- Name
- Intelligence
- Ranking

Course
- Name
- Instructor
- Rating
- Difficulty

Registration
- RegID
- Course
- Student
- Grade
- Satisfaction

CPSC 422, Lecture 33
University Domain Example - An Instance of the Schema

One professor is the instructor for both courses.

Jane Doe is registered for only one course, Phil101, while the other student is registered for both courses.
University Domain Example - Another Instance of the Schema

There are two professors instructing a course.

There are three students in the Phil201 course.
University Domain Example - fixed vs. probabilistic attributes

Fixed attributes are shown in regular font.

Probabilistic attributes are shown in italic regular font.
PRM Semantics: Skeleton Structure

- A *skeleton structure* $\sigma$ of a relational schema is a *partial specification of an instance of the schema*. It specifies
  - set of objects for each class,
  - values of the fixed attributes of these objects,
  - relations that hold between the objects

- The values of probabilistic attributes are left unspecified

- A *completion* $I$ of the skeleton structure $\sigma$ extends the skeleton by also specifying the values of the probabilistic attributes
University Domain Example - Relational Skeleton

Professor
Name
Prof. Vincent
Popularity
???
Teaching-Ability
???

Student
Name
John Doe
Intelligence
???
Ranking
???

Course
Name
Phil201
Difficulty
???
Rating
???

Registration
RegID
#5723
Grade
???
Satisfaction
???

Registration
RegID
#5639
Grade
A
Satisfaction
3
University Domain Example - The Completion Instance I

PRMs also allow multiple possible instances and values
Lecture Overview

• Recap Motivation and Representation for Probabilistic Relational Models (PRMs)
  • Full Relational Schema and its Instances
  • Relational Skeleton and its Completion Instances

• Probabilistic Model of PRMs
  • Dependency Structure
  • Parameters
PRMs: Probabilistic Model

- The probabilistic model consists of two components:
  - the qualitative dependency structure, $S$
  - the parameters associated with it, $\Theta_S$

- The dependency structure is defined by associating with each attribute $X.A$ a set of parents $Pa(X.A)$; parents are attributes that are “direct influences” on $X.A$. This dependency holds for any object of class $X$
Dependencies within a class

The prob. attribute $X.A$ can depend on another probabilistic attribute $B$ of $X$. This induces a corresponding dependency for individual objects.
Dependencies across classes

- The attribute $X.A$ can also depend on attributes of related objects $X.\tau.B$, where $\tau$ is a slot chain
Possible PRM Dependency Structure for the University Domain

Edges from one class to another are routed through slot-chains.

Edges correspond to probabilistic dependency for objects in that class.

CPSC 422, Lecture 33
Let’s derive the Corresponding “grounded” Dependency Structure for this Skeleton

- Professor Name: Prof. Gump
  - Popularity: ???
  - Teaching-Ability: ???

- Professor Name: Prof. Vincent
  - Popularity: ???
  - Teaching-Ability: ???

- Course Name: CS101
  - Difficulty: ???
  - Rating: ??

- Course Name: Phil101
  - Difficulty: ???
  - Rating: ??

- Registration RegID: #3
  - Grade: ??
  - Satisfaction: ??

- Registration RegID: #5
  - Grade: ??
  - Satisfaction: ??

- Registration RegID: #6
  - Grade: ??
  - Satisfaction: ??

- Student Name: Jane Doe
  - Intelligence: high
  - Ranking: average

- Student Name: Sue Chu
  - Intelligence: ???
  - Ranking: ???

- Professor Name: Prof. Vincent
  - Popularity: ???
  - Teaching-Ability: ???

- Professor Name: Prof. Gump
  - Popularity: ???
  - Teaching-Ability: ???

CPSC 422, Lecture 33
Parameters of PRMs

- A PRM contains a conditional probability distribution (CPD) $P(X.A|Pa(X.A))$ for each attribute $X.A$ of each class e.g.,

$$P(\text{Registration.Grade} \mid \text{Course.Difficulty, Student.Intelligence})$$

- More precisely, let $U$ be the set of parents of $X.A$. For each tuple of values $u \in V(U)$, the CPD specifies a distribution $P(X.A|u)$ over $V(X.A)$.

\[
\begin{array}{c|ccc}
\text{Course.Difficulty} & \text{A} & \text{B} & \text{C} \\
\hline
\text{high} & 0.5 & 0.4 & 0.1 \\
\text{low} & 0.1 & 0.5 & 0.4 \\
\end{array}
\]

\[
\begin{array}{c|ccc}
\text{Student.Intelligence} & \text{A} & \text{B} & \text{C} \\
\hline
\text{high} & 0.8 & 0.1 & 0.1 \\
\text{low} & 0.3 & 0.6 & 0.1 \\
\end{array}
\]

- The parameters in all of these CPDs comprise $\Theta_S$.
Now, what are the parameters $\theta_S$?
Problem with some parameters $\theta_S$

- **Course**: Rating, Difficulty
- **Professor**: Teaching-Ability, Popularity
- **Student**: Intelligence, Ranking
- **Registration**: Satisfaction, Grade
- **Professor** connects to **Course**
- **Student** connects to **Registration**

Questions:

A. too many parents
B. variable # of parents
C. too few parents
D. another problem
When the slot chain $\tau$ (e.g., Course, Instructor) is not guaranteed to be single-valued, we must specify the probabilistic dependence of

- $x.A$ (Registration, Satisfaction)
- on the set $\{y.B: y \in x.\tau\}$

The Teaching-Ability of the profs who are instructors of the Course.
How to specify cond. Prob. When # of parents can vary?

• The notion of aggregation from database theory gives us the tool to address this issue; i.e., \( x.A \) will depend probabilistically on some aggregate property of this set.
Aggregation in PRMs

Examples of aggregation are:

• the **mode** of the set (most frequently occurring value);
• **mean** value of the set (if values are numerical);
• **median**, **maximum**, or **minimum** (if values are ordered);
• **cardinality** of the set; etc.
The student’s ranking depends on the average of his grades

A course satisfaction depends on the teaching abilities of its instructors

A student may take multiple courses

A course rating depends on the average satisfaction of students in the course

The same course can be taught by multiple profs
CPDs in PRMs

Course
- Rating
- Difficulty

Professor
- Teaching-Ability
- Popularity

Registration
- Satisfaction
- Grade

Student
- Intelligence
- Ranking

CPSC 422, Lecture 33
JPD in PRMs

- Given a **skeleton structure** $\sigma$ for our schema, we can apply these **local conditional probabilities** to define a **JPD** (joint probability distribution) **over all completions** of the skeleton.

- Note that the objects and relations between objects in a skeleton are always specified by $\sigma$, hence we are disallowing uncertainty over the relational structure of the model.
Parameter Sharing / CPTs
reuse, where else?

- Temporal Models
- Because of the stationary assumption!
To define a coherent probabilistic model as a Bayesian network, we **must ensure that our probabilistic dependencies are...**

- A. Polynomial
- **B. Acyclic**
- C. Cyclic
- D. Recursive
Class Dependency Graph for the University Domain

- Course.Difficulty
- Student.Intelligence
- Professor.Teaching-Ability
- Registration.Grade
- Professor.Popularity
- Registration.Satisfaction
- Student.Ranking
- Course.Rating
Ensuring Acyclic Dependencies

• In general, however, a cycle in the class dependency graph does not imply that all skeletons induce cyclic dependencies.

• A model may appear to be cyclic at the class level, however, this cyclicity is always resolved at the level of individual objects.

• The ability to guarantee that the cyclicity is resolved relies on some prior knowledge about the domain. The user can specify that certain slots are guaranteed acyclic.
Relational Schema for the Genetics Domain

Person

- Name
- Blood-Type
- P-Chromosome
- M-Chromosome

Blood Test

- BT-ID
- Patient
- Results
- Contaminated

1

M
Dependency Graph for Genetics Domain

Person.M-chromosome

Person.P-chromosome

Person.BloodType

BloodTest.Contaminated

BloodTest.Result
PRM for the Genetics Domain

(Father)

Person

BloodType

P-chromosome

M-chromosome

(Mother)

Person

BloodType

P-chromosome

M-chromosome

Person

BloodType

P-chromosome

M-chromosome

BloodTest

Contaminated

Result
Dependency Graph for Genetics Domain

- **Person.M-chromosome**
- **Person.P-chromosome**
- **Person.BloodType**
- **BloodTest.Contaminated**
- **BloodTest.Result**

Dashed edges correspond to “guaranteed acyclic” dependencies.
Learning Goals for today’s class

You can:

• Build the grounded Bnet, given a Relational Skeleton, a dependency structure, and the corresponding parameters

• Define and apply guaranteed acyclicity
422 big picture: Where are we?

Deterministic

- First Order Logics
- Ontologies
- Full Resolution
- SAT

Stochastic

- Belief Nets
  - Approx. : Gibbs
- Markov Chains and HMMs
  - Forward, Viterbi
  - Approx. : Particle Filtering
- Undirected Graphical Models
  - Markov Networks
  - Conditional Random Fields
- Markov Decision Processes and Partially Observable MDP
  - Value Iteration
  - Approx. Inference

StarAI (statistical relational AI)
Hybrid: Det + Sto

Prob CFG
Prob Relational Models
Markov Logics

Applications of AI

Representation
Reasoning
Technique
Last class on Fri

- Beyond 322/422 (ML + grad courses)
- Watson
- Final Exam

Fill out on-line Teaching Evaluation