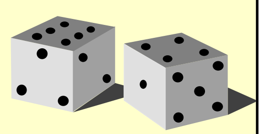
Probabilistic Relational Models

Daphne Koller
Stanford University
koller@cs.Stanford.edu

Joint work with:
Lise Getoor, Stanford
Nir Friedman, Hebrew U.
Avi Pfeffer, Harvard
Ben Taskar, Stanford

Probabilistic models

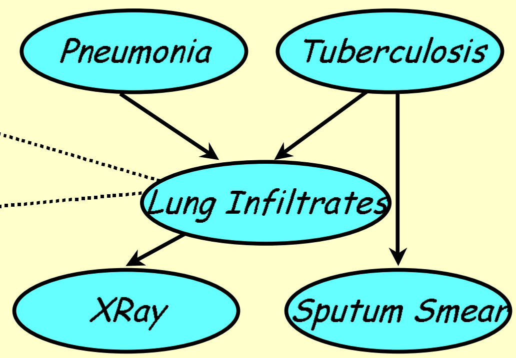


- Uncertainty model:
  - space of “possible worlds”;
  - probability distribution over this space.
- Worlds: often defined via a set of state variables
  - medical diagnosis: diseases, symptoms, findings, …

  each world: an assignment of values to variables.

Bayesian networks

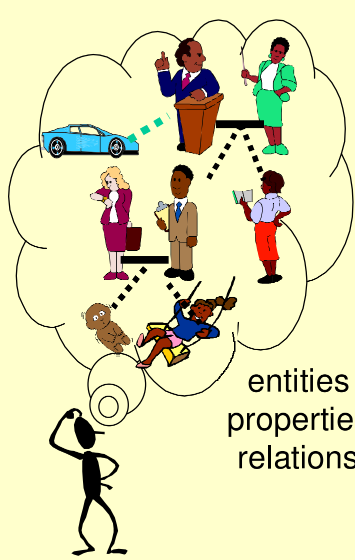
P	T	$P(I \mid P, T)$	
p	t	0.8	0.2
p	$\bar{t}$	0.6	0.4
$\bar{p}$	t	0.2	0.8
$\bar{p}$	$\bar{t}$	0.01	0.99



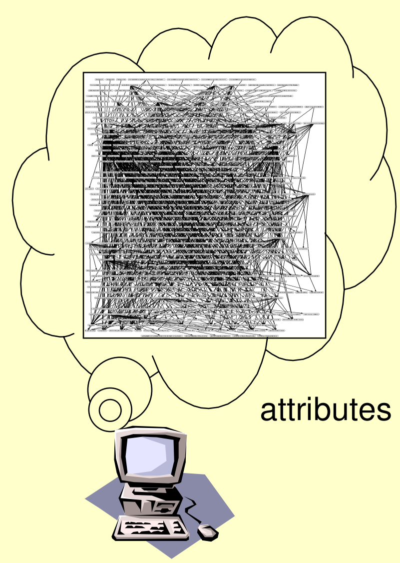
nodes = variables
edges = direct causal influence

Network structure encodes independence assumptions:
XRay conditionally independent of *Pneumonia* given *Infiltrates*

Basic units of knowledge



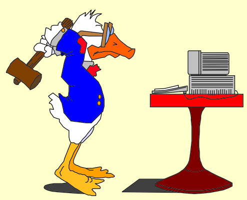
entities
properties
relations



attributes

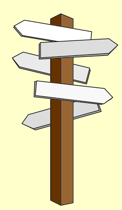
So what?

- Set of entities and relations between them is determined at BN design time
  - structure must be known in advance
  - hard to adapt to changes
- BNs for complex domains are large & unstructured
  $\Rightarrow$ very hard to build
- No ability to generalize
  - across “similar” individuals
  - across related situations



BNs are not suitable for representing complex, structured, flexible domains.

Outline

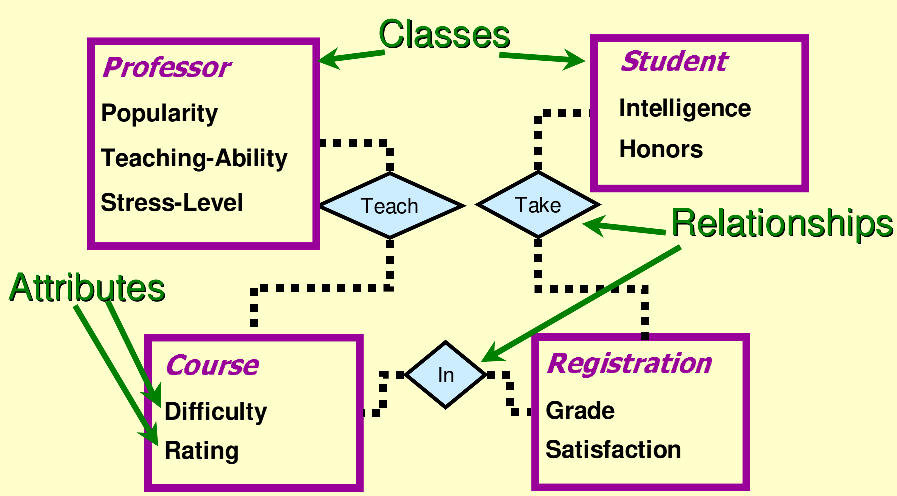


- ➢ Probabilistic relational models: Syntax & semantics
- PRM inference: Exploiting structure
- PRM learning: Learning from structured data
- Extensions
- Conclusions & research directions

Probabilistic Relational Models

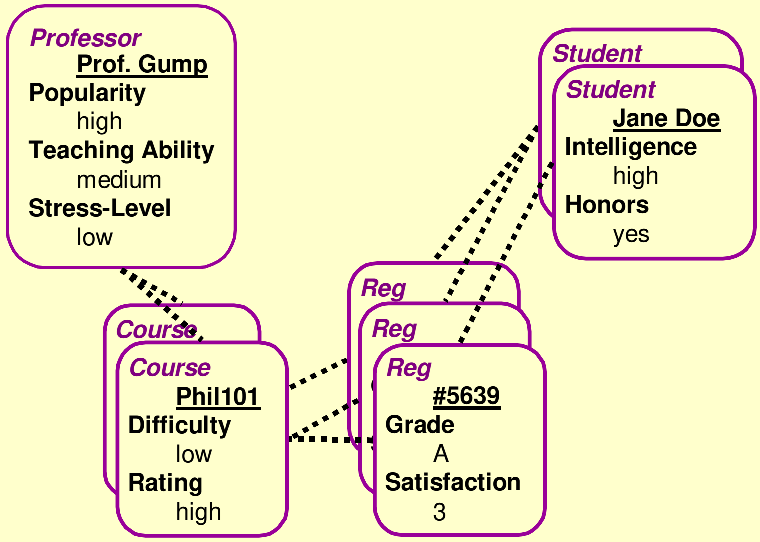
- Combine advantages of relational logic & BNs:
  - natural domain modeling: objects, properties, relations;
  - generalization over a variety of situations;
  - compact, natural probability models.
- Integrate uncertainty with relational model:
  - properties of domain entities can depend on properties of related entities;
  - uncertainty over relational structure of domain.

Relational Schema

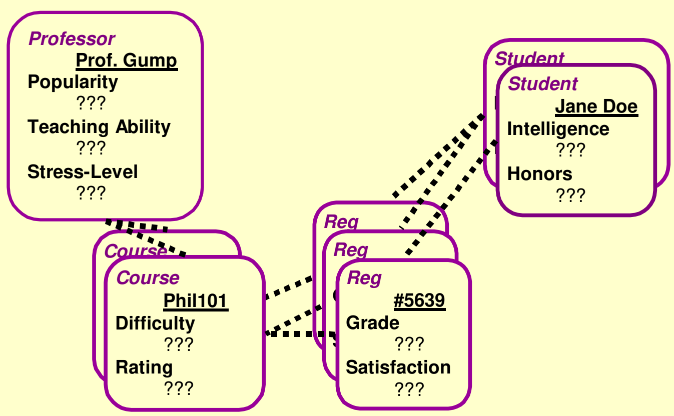


- Describes types of objects & relations in domain

Example Instance I



Basic PRMs: Attribute Uncertainty



Fixed skeleton σ
- set of objects in each class
- relations between them

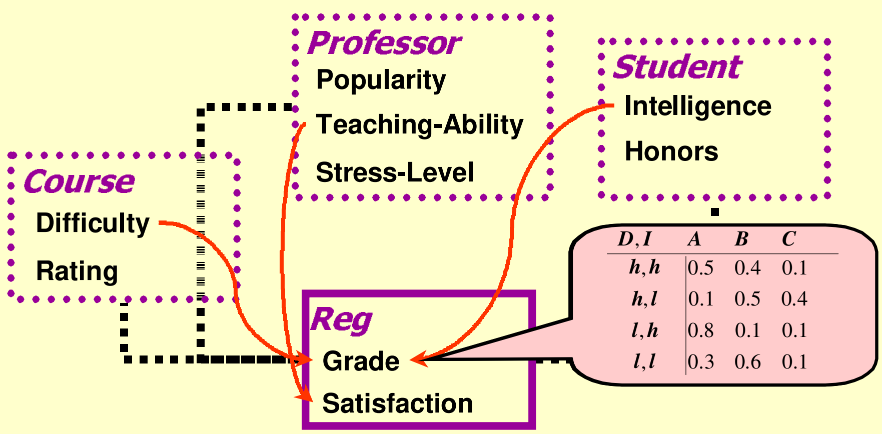
Uncertainty
- over assignments of values to attributes

Probabilistic relational models

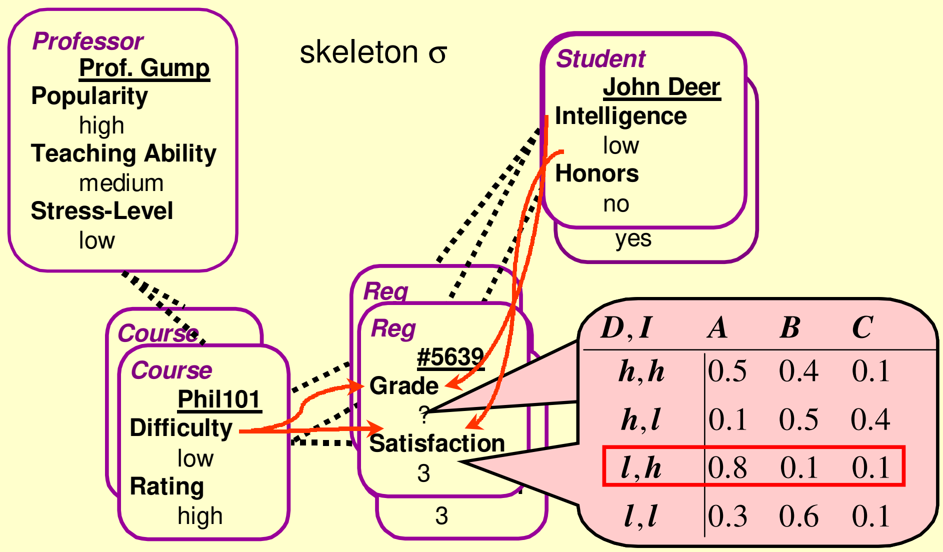
- A PRM specifies a probabilistic model for each class
- For each attribute, it defines:
  - dependencies on other attributes, in same object or related objects
  - a conditional probability distribution

Probabilistic relational models

- A PRM specifies a probabilistic model for each class
- For each attribute, it defines:
  - dependencies on other attributes, in same object or related objects
  - a conditional probability distribution

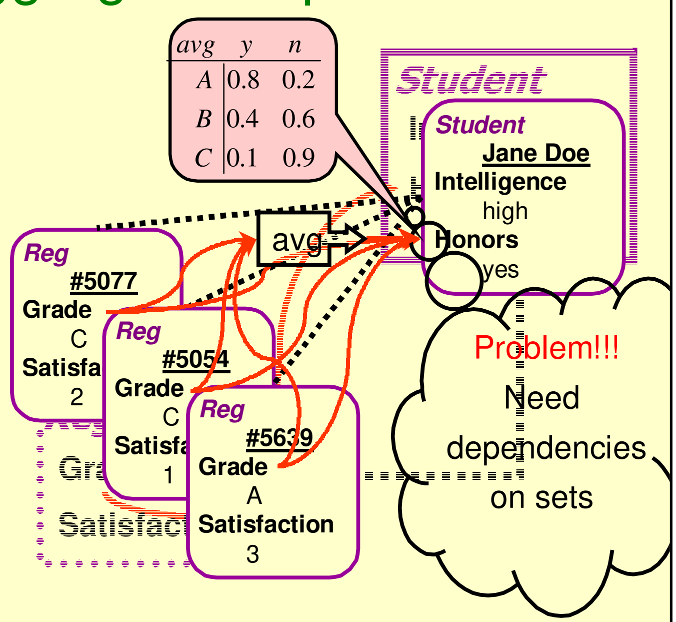


PRM instantiation



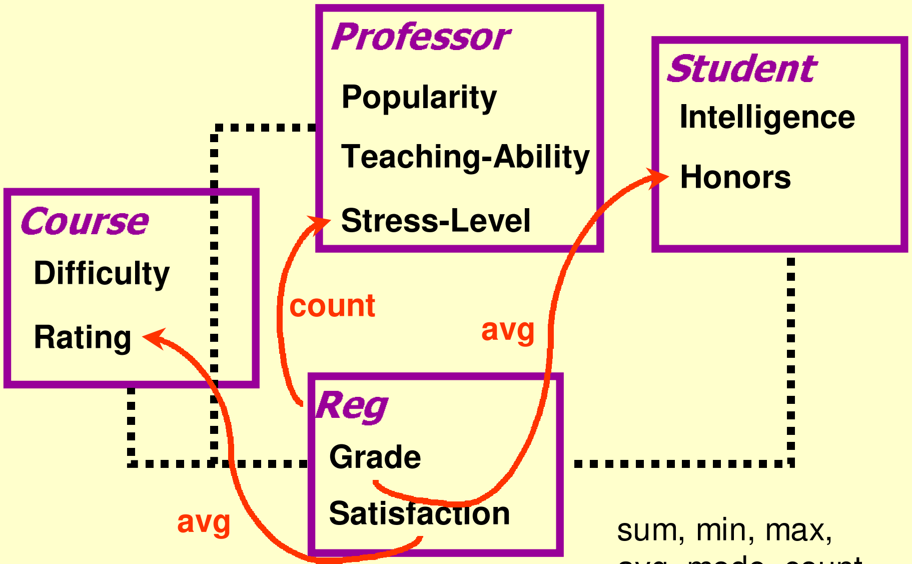
D, I	A	B	C
h, h	0.5	0.4	0.1
h, l	0.1	0.5	0.4
l, h	0.8	0.1	0.1
l, l	0.3	0.6	0.1

PRM: Aggregate Dependencies



avg	y	n
A	0.8	0.2
B	0.4	0.6
C	0.1	0.9

PRM: Aggregate Dependencies



sum, min, max,
avg, mode, count

Basic PRMs: Summary

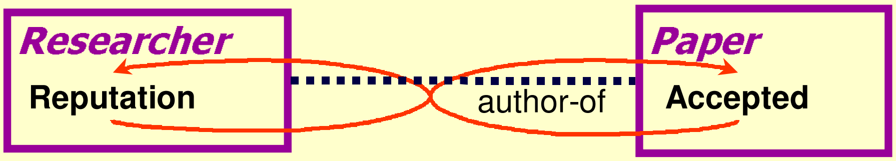
- A PRM specifies
  - A probabilistic dependency structure S
    - A set of parents for each class attribute $X.A$
  - A set of *local probability models* Θ
- Given a skeleton structure σ, a PRM specifies a probability distribution over instances $\mathcal{I}$:
  - Over attribute values of all objects in σ

$$P(\mathcal{I} \mid \sigma, S, \Theta) = \prod_{x \in \sigma} \prod_{x.A} P(x.A \mid parents_{S,\sigma}(x.A))$$

Objects Attributes

attributes of objects linked to x:
links specified in σ (different for every x)
class-level dependencies specified in S

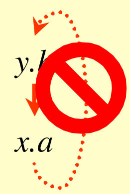
Coherence



- Dependency ordering over attributes:

  $y.B \prec_\sigma x.A$ if $X.A$ depends on $Y.B$

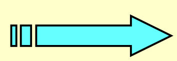
- PRM defines a coherent probability model over skeleton σ if $\prec_\sigma$ is acyclic



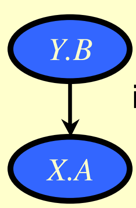
Guaranteeing Acyclicity

How do we guarantee that a PRM is acyclic for ***every*** skeleton?

PRM dependency structure S



dependency graph

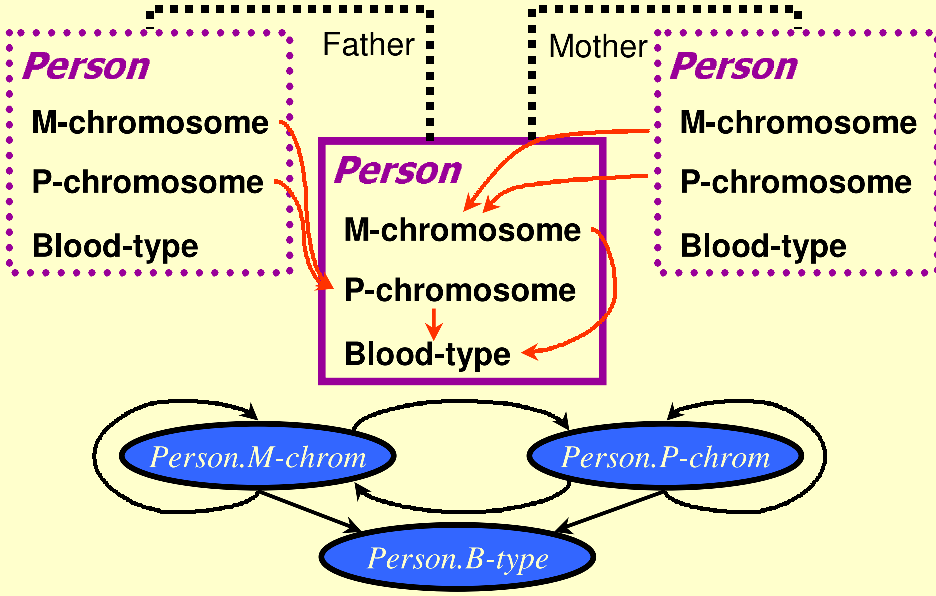


if $X.A$ depends directly on $Y.B$

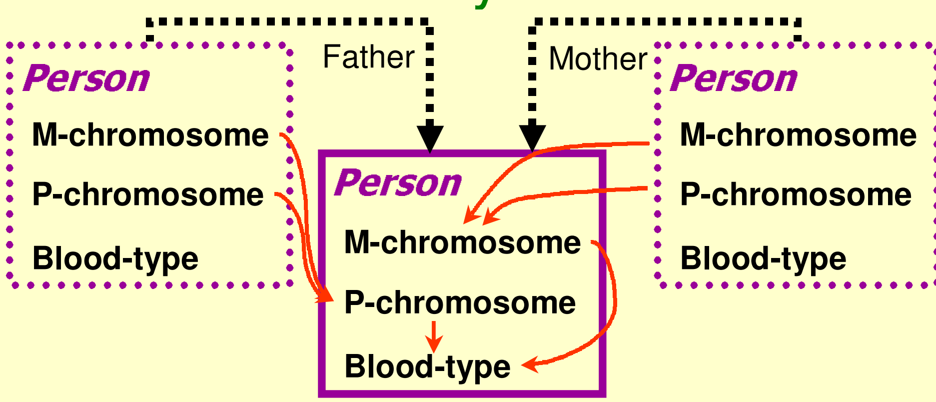
Attribute stratification:

dependency graph acyclic $\Rightarrow \prec_\sigma$ acyclic for any σ

Limitation of Stratification



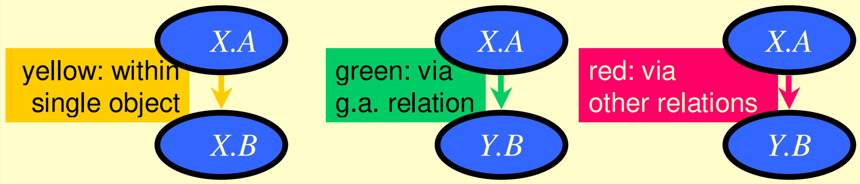
Guaranteed Acyclic Relations

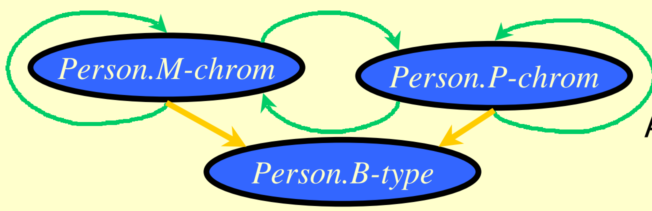


- Prior knowledge: the Father-of relation is acyclic
  - Dependence of *Person.A* on *Person.Father.B* cannot induce cycles

Guaranteeing Acyclicity

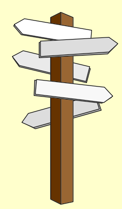
- With guaranteed acyclic relations, some cycles in the dependency graph are guaranteed to be safe.
- We color the edges in the dependency graph





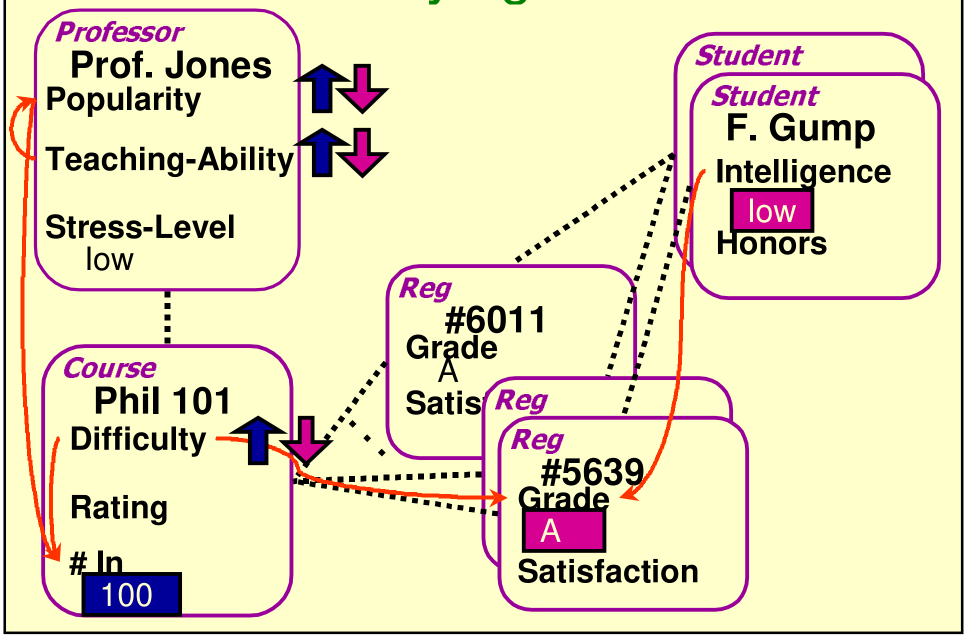
A cycle is safe if
- it has a green edge
- it has no red edge

Outline

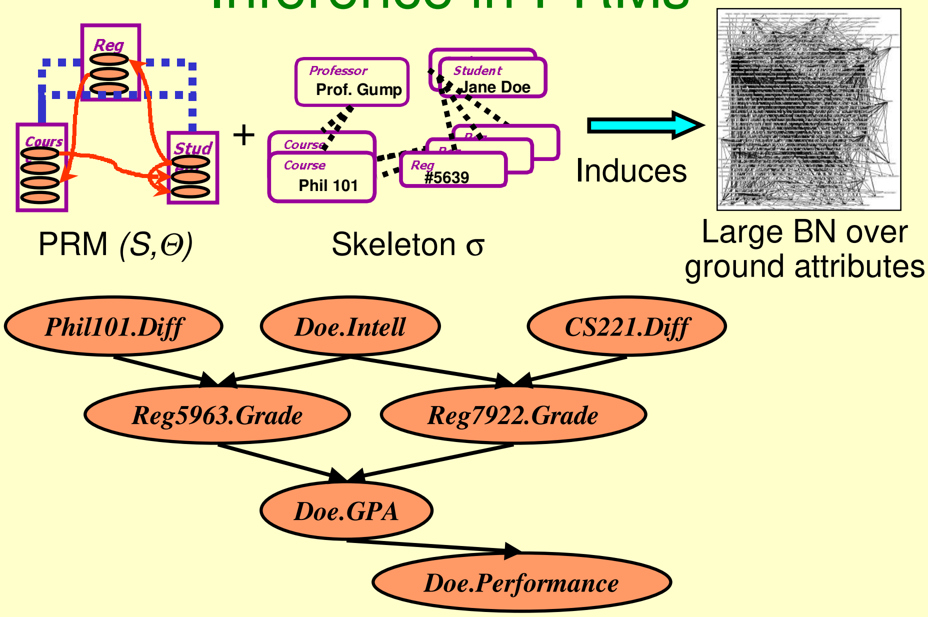


- Probabilistic relational models: Syntax & semantics
- ➢ PRM inference: Exploiting structure
- PRM learning: Learning from structured data
- Extensions
- Conclusions & research directions

Querying a PRM



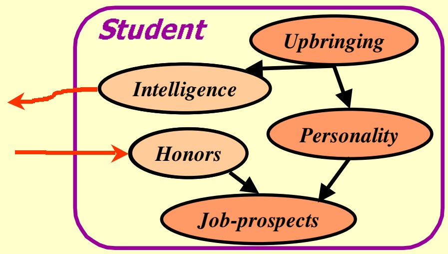
Inference in PRMs



Exploiting structure

Structure can be exploited to speed up inference.

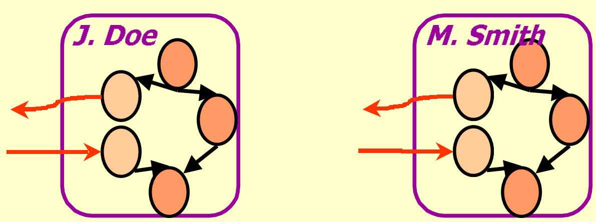
- **Encapsulation:** objects interact only in limited ways
  - ⇒ Inference can be encapsulated within objects, with "communication" limited to interfaces.



Exploiting structure

Structure can be exploited to speed up inference.

- **Encapsulation:** objects interact only in limited ways
  - ⇒ Inference can be encapsulated within objects, with "communication" limited to interfaces.
- **Reuse:** objects from same class have same model*
  - ⇒ if internal evidence is the same, can reuse inference.



* a form of *lifting*

Exploiting structure

Structure can be exploited to speed up inference.

- **Encapsulation:** objects interact only in limited ways
  - ⇒ Inference can be encapsulated within objects, with "communication" limited to interfaces.
- **Reuse:** objects from same class have same model
  - ⇒ if internal evidence is the same, can reuse inference.
- **Combinatorics:** multiple generic objects often play identical role in the model
  - ⇒ can compute their total contribution in closed form.

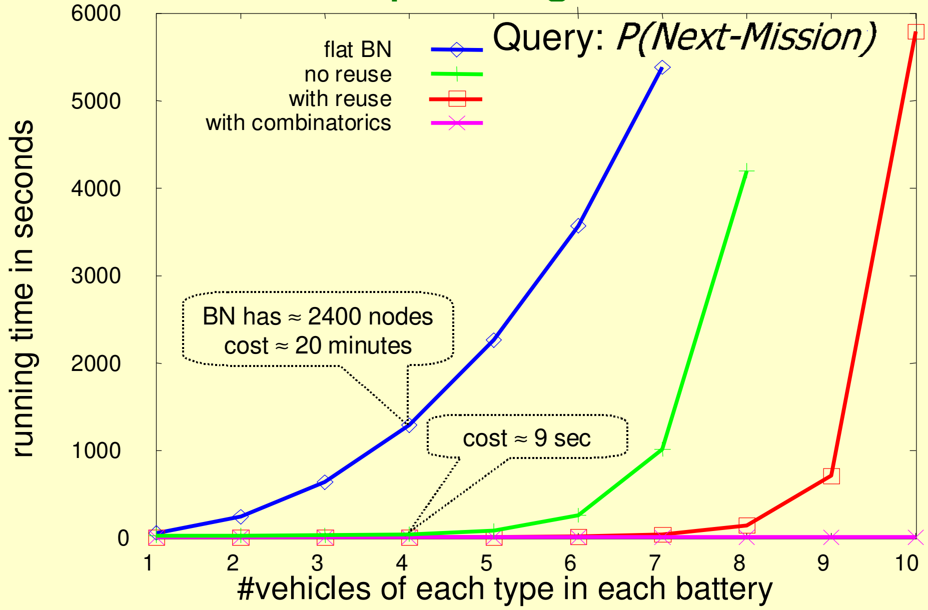
Real-world case study

Battlefield situation assessment for missile units

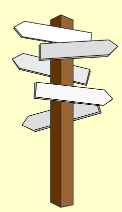
- several locations
- many units
- each has detailed model

- Example object classes:
  - *Battalion*
  - *Battery*
  - *Vehicle*
  - *Location*
  - *Weather*.
- Example relations:
  - *At-Location*
  - *Has-Weather*
  - *Sub-battery/In-battalion*
  - *Sub-vehicle/In-battery*

Effect of Exploiting Structure



Outline

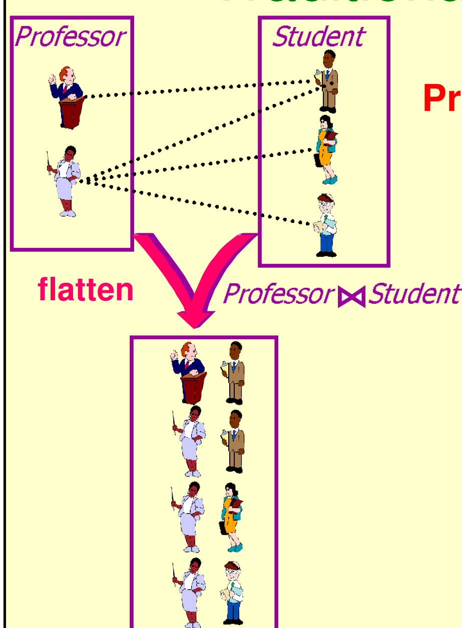


- Probabilistic relational models: Syntax & semantics
- PRM inference: Exploiting structure
- ➢ PRM learning: Learning from structured data
- Extensions
- Conclusions & research directions

Learning from Relational Data

- Data sources
  - structured: relational and object-oriented databases
  - semi-structured: XML, World Wide Web
- Goal: Learn statistical model from relational data
  - Supports discovery of statistical patterns, classification and anomaly detection
- Traditional approaches
  - work well with flat representations
  - assume fixed length attribute-value vectors
  - assume independent identically distributed (IID) samples

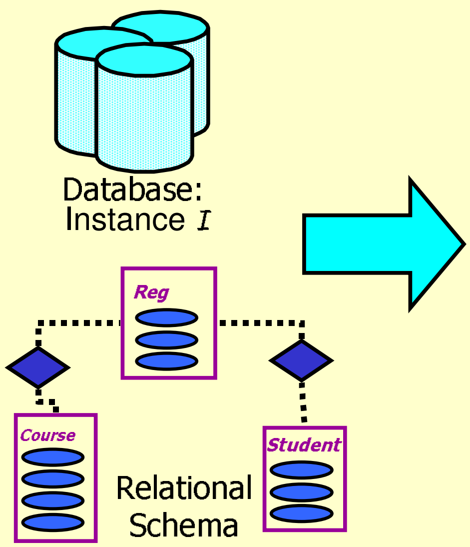
Traditional Approach

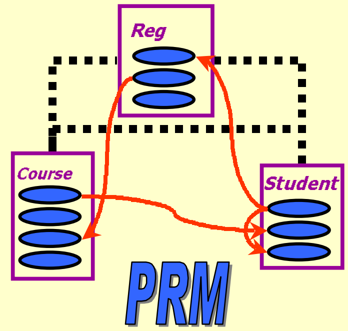


Problems:

- expensive for large databases
- introduces statistical skew
- must fix attributes in advance
  - ⇒ can represent only a limited set of structures
- loses relational structure
  - ⇒ incapable of detecting graph substructures

Learning PRMs





- Parameter estimation
- Structure selection

Learning PRMs

- Basic approach:
  - Define scoring function
  - Do local search over legal structures

$$Score(S:\mathcal{I}) = \log[\overbrace{P(\mathcal{I} \mid S)}^{\text{marginal likelihood}}\overbrace{P(S)}^{\text{prior}}]$$

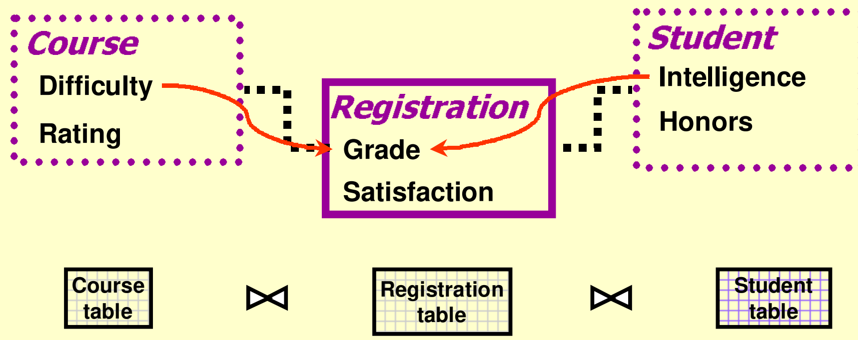
- Properties of Bayesian score:
  - Automatically trades off fit to data with model complexity
  - Has simple closed form
  - Can be computed based only on *sufficient statistics*:

$$Count\big(x.A,\ parents(x.A)\big)$$

$$Count(Reg.Grade{=}A, Course.Diff{=}lo, Student.Intel{=}hi)$$

Computing Sufficient Statistics

- Use database technology to perform computation efficiently
  - indices
  - materialized views

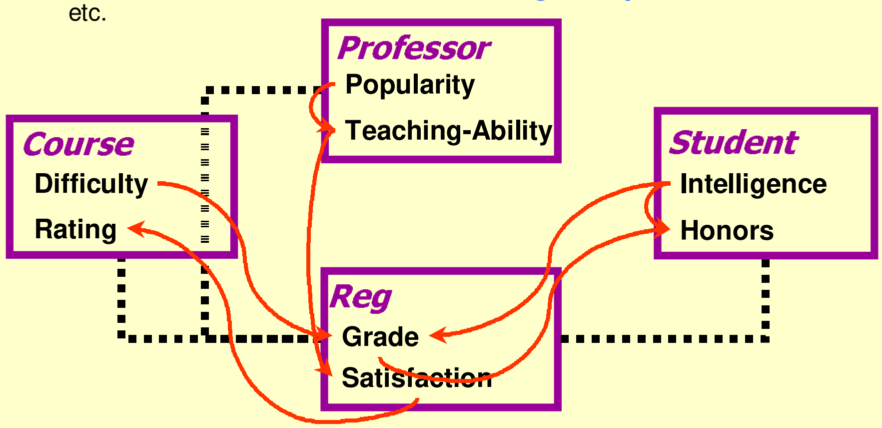


Exploiting Locality: Phased Search

Search for dependencies:

first between attributes within object
then from directly linked objects
then from objects two links away
etc.

At each phase, only consider dependencies that are informative even given previous ones



TB Cases in SF

Patient (≈2300)
- Ethnicity
- Homeless
- Age @ diagnosis
- HIV result
- Disease-site
- X-ray
- •
- •
- •

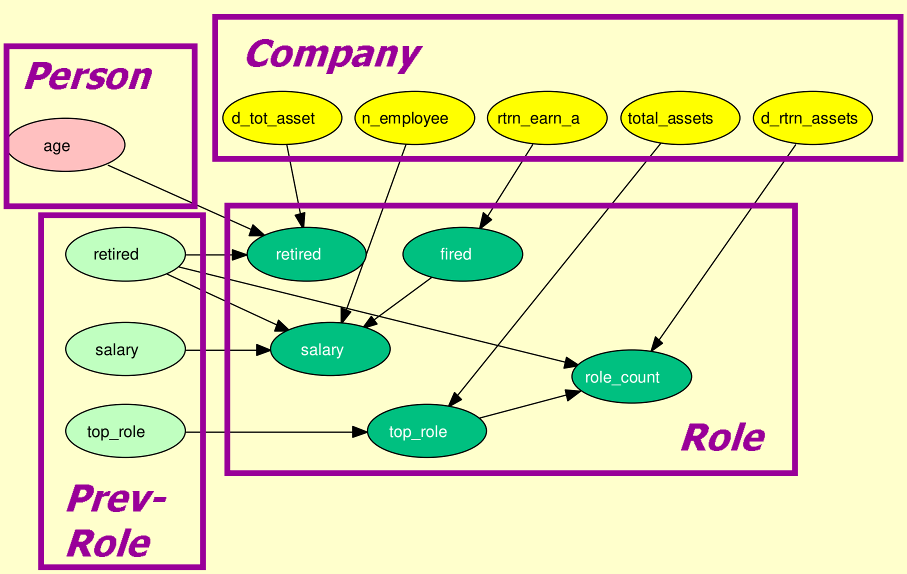
Contact (≈20000)
- Contact-type
- Age
- Care
- Infected
- •
- •
- •

Subcase (≈200)
- Contact-type
- Close-contact
- Transmitted
- •
- •
- •

Strain (≈1000)
- Num-infected
- Infectivity

contact1.hkb

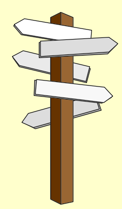
Bank Data (Companies & Officers)



Applications of PRM learning

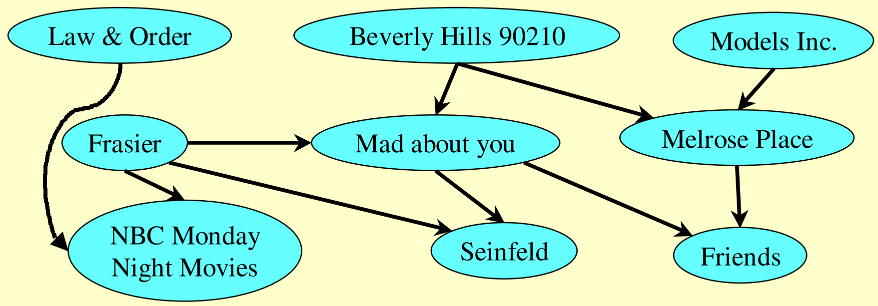
- Data exploration:
  - discovering significant patterns in the data
- Data summarization:
  - compact summary of large relational database
- Inference:
  - reasoning about important unobserved attributes
- Clustering:
  - discovering clusters of entities that are similar
- Anomaly detection:
  - finding “unusual” elements in data

Outline



- Probabilistic relational models: Syntax & Semantics
- PRM inference: Exploiting structure
- PRM learning: Learning from structured data
- ➢ Extensions
  - ➢ From instances to classes
  - ➢ Models of relational uncertainty
  - ➢ Query size estimation
- Conclusions & research directions

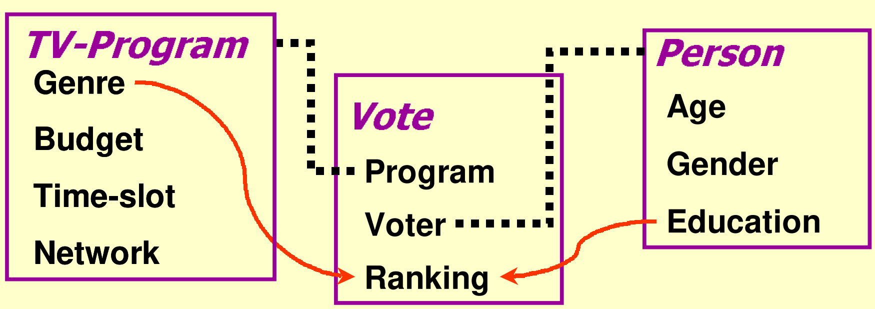
BN for Collaborative filtering



Breese, et al. UAI-98

- Each program has own dependency model, cannot generalize over programs
  - people who like ensemble comedies
- Cannot model dependencies between people
  - John watches what his roommate watches

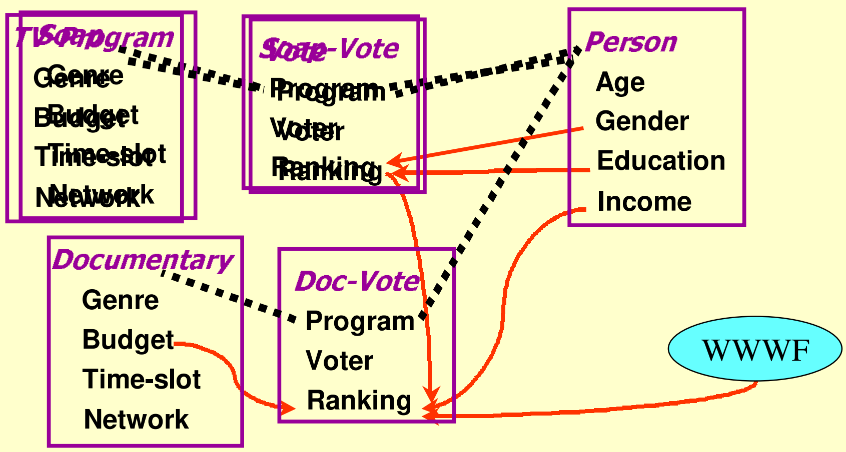
PRMs for Collaborative Filtering



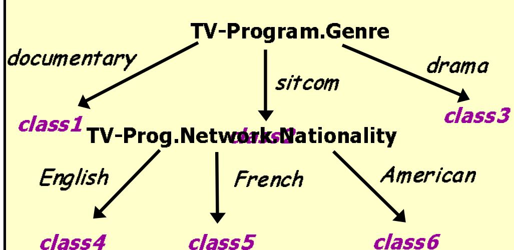
- All instances in class must use same dependency model:
  - cannot distinguish documentaries and sitcoms
- PRM cannot have dependencies that are “cyclic”
  - ranking for Frasier depends on ranking for Friends

Desired Model

Allows *both* class and instance dependencies



PRMs with Class Hierarchy



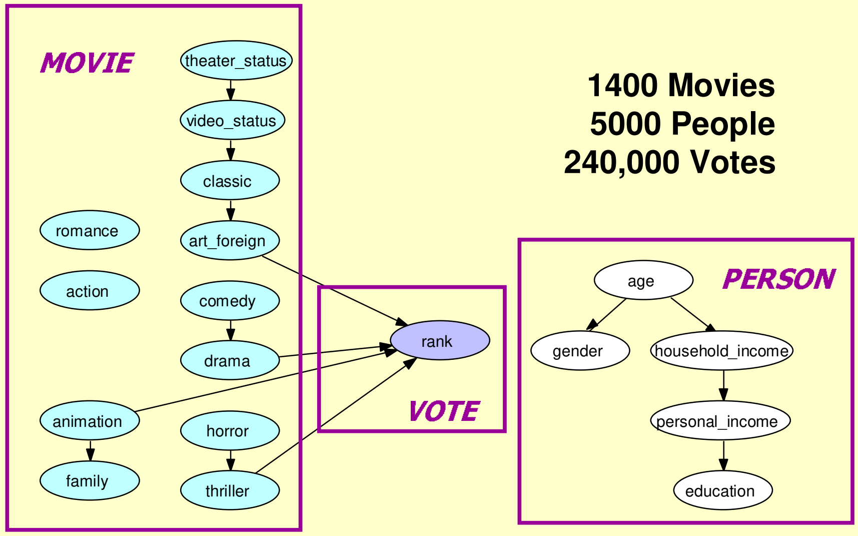
- Class hierarchy = tree
- Defined via attributes of entity and related entities
- Can be learned from data

Allow us to:

- Refine "heterogenous" class into coherent subclasses
- Specialize/inherit probabilistic model along class hierarchy
- Add dependencies that were originally "cyclic"

Provides bridge from class-based model to instance-based model

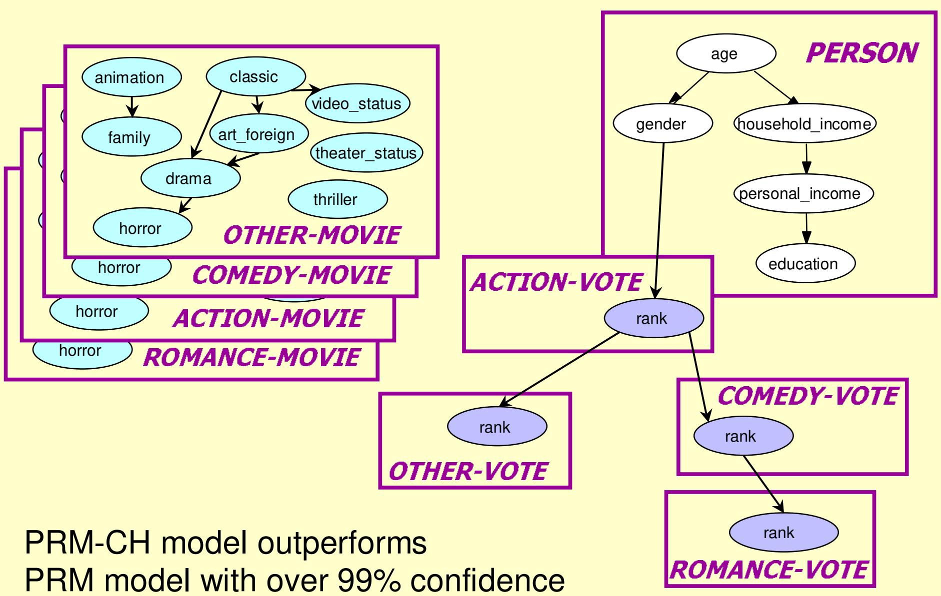
EachMovie^{+} PRM



1400 Movies
5000 People
240,000 Votes

http://www.research.digital.com/SRC/EachMovie

EachMovie^{+} PRM-CH



PRM-CH model outperforms
PRM model with over 99% confidence

Structural Uncertainty

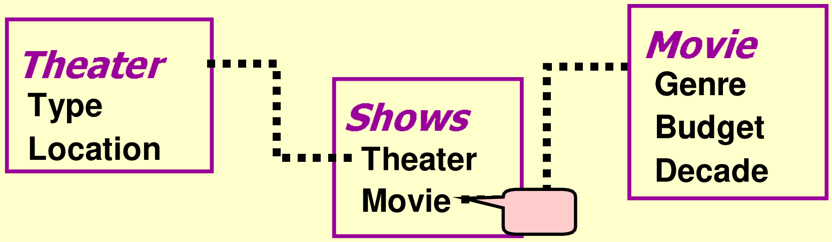
Uncertainty over the relationships that hold between entities in a domain

- Has actor *appeared* in movie?
- Has person *seen* movie?
- Which company *acquired* company A?
- With which strain is patient *infected*?

PRMs w/ Structural Uncertainty

- Introduce probabilistic model over links (relations) in the database
- Advantages:
  - Applicable in cases where we do not have full knowledge of link structure
  - Allows us to predict the existence of links from other attributes
  - Even when relational structure is known, can improve predictive ability for attributes
- Many possible approaches, very rich area

PRM w/ Reference Uncertainty

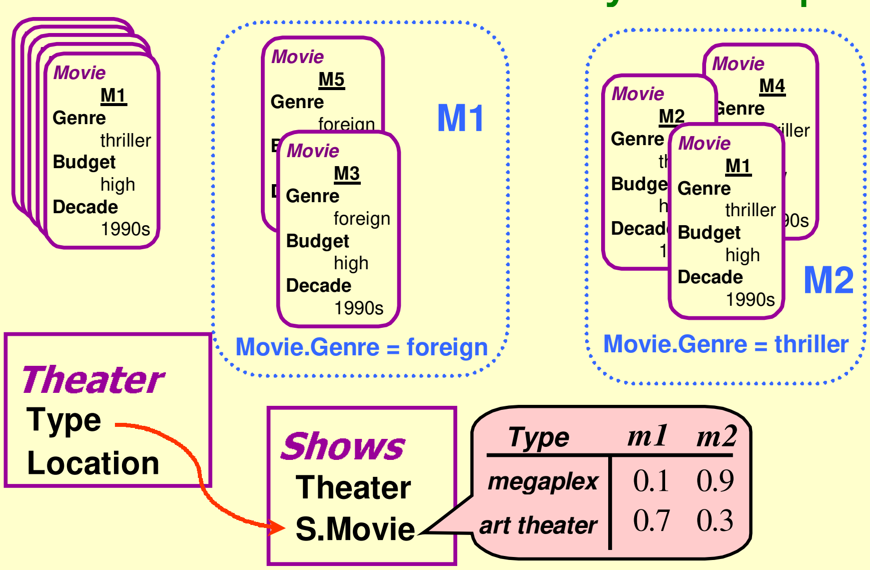


Dependency model for foreign keys

Naïve Approach: multinomial over primary key

- noncompact
- limits ability to generalize

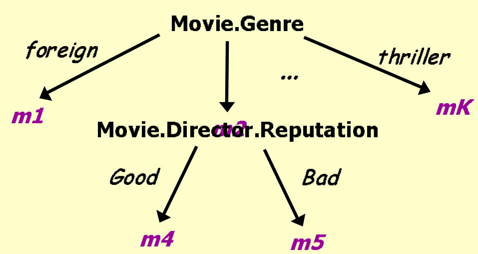
Reference Uncertainty Example



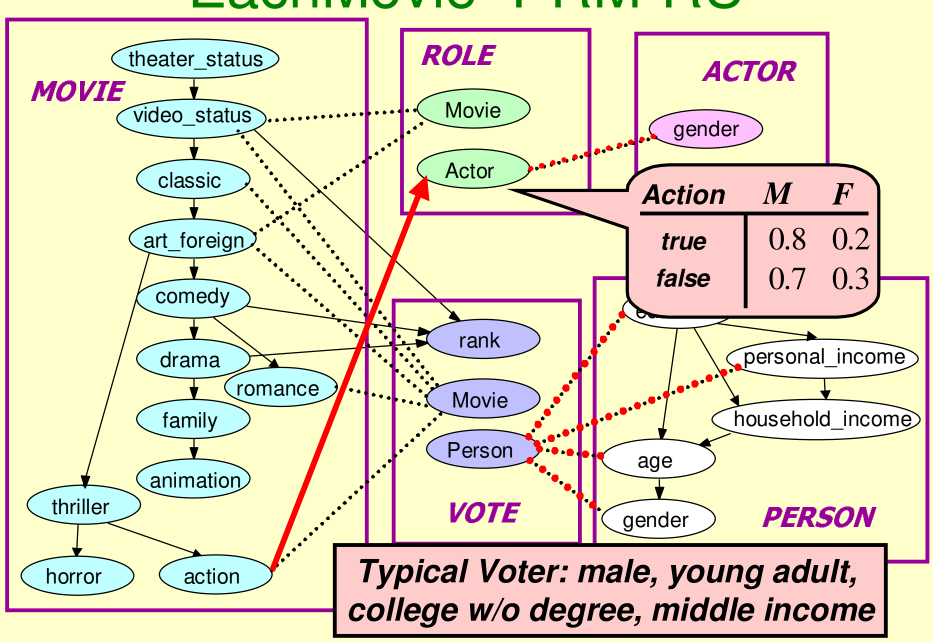
Learning PRM-RU

Construct decision tree for partition defined over attributes *observed in training set*

- New operators: Abstract/Refine
  - Refine splits partition on class attribute
  - or related class attribute



EachMovie+ PRM-RU



Query Size Estimation

Task: estimate result size of DB query

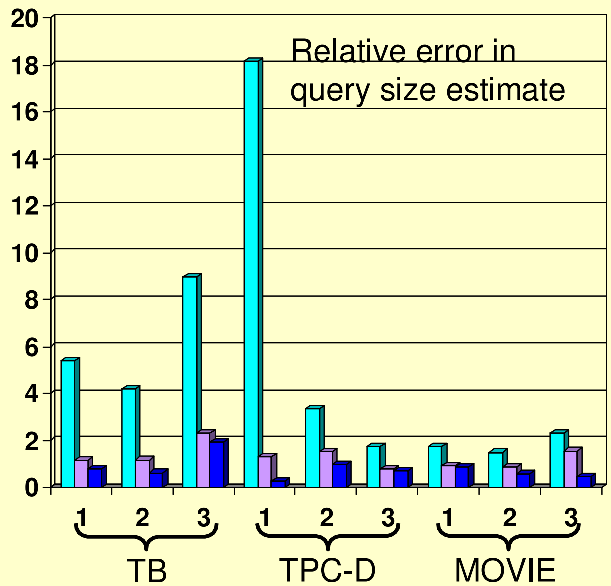
$$\sigma_{\substack{C.Diff = lo \\ R.Grade = C \\ S.Int = hi}} \left(\text{Student table} \bowtie \text{Reg table} \bowtie \text{Course table} \right)$$

Uses:

- Quick approximate query answering
- Query optimization
  - Cost of query plan depends on size of intermediate results

- PRMs can be used for select selectivity estimation
- PRM-RUs also handle join selectivity estimation

RU for query-size estimation



Task: estimate result size of select-join DB queries

Learning time < 10 min
Estimation time < 0.01 sec