Probabilistic Graphical Models

10-708

Models with Higher-Level Structures: logic + probabilities

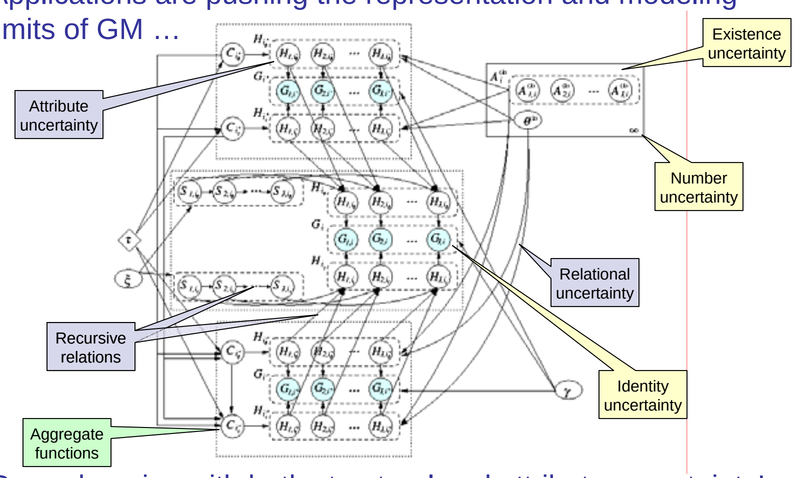
Eric Xing

Lecture 21, Nov 28, 2005

Reading: Getoor et al 2001, Milch et al. 2005

Limitations of GM

- Applications are pushing the representation and modeling limits of GM …



- Open domains with both structural and attribute uncertainty!

Propositional Logic

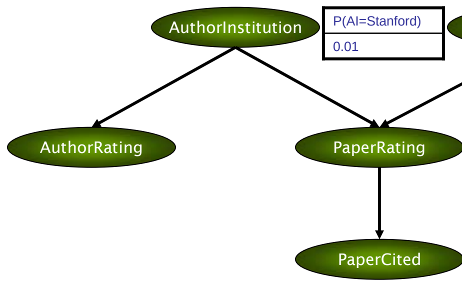
- Ontological commitment: the world consists of propositions, or facts, or atomic events, which are either true or false
  - *e.g., Paper_X_HighPaperRating*
- Set of 2^n possible worlds – one for each truth assignment to the *n* propositions
- Propositional logic allows us to compactly represent restrictions on possible worlds:
  - If *Auther_A_HighPublicationRating* then *Paper_X_HighPaperRating*
- Means that we have eliminated the possible worlds where *Auther_A_HighPublicationRating* is true but *Paper_X_HighPaperRating* is false.

Propositional Uncertainty

- To model uncertainty we would like to represent a probability distribution over all possible worlds.
- To represent the full joint distribution we would need 2^n-1 parameters (infeasible)
- Insight: the value of most propositions isn't affected by the value of most other propositions!
- More formally, some propositions are conditionally independent of each other given the value of other propositions

Bayesian Networks

- A BN uses a directed acyclic graph to encode these independence assumptions



P(AI=Stanford)
0.01

P(JR=high)
0.3

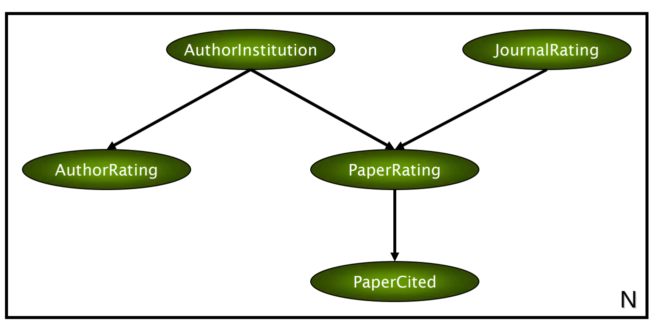
AI	P(AR=high | AI)
Stanf.	0.1
other	0.001

AI	JR	P(PR=high | AI, JR)
Stanf.	high	0.6
Stanf.	low	0.1
other	high	0.2
other	low	0.01

PR	P(PC=true | PR)
high	0.5
low	0.01

- This model encodes the assumption that each variable is independent of its non-descendents given its parents
  - The full joint over these five binary variables would need 2^5-1=31 parameters, but this factored representation only needs 10!

Plates and beyond



- Graphical model applies to any paper → already "universally quantified"
  - a *Plate* stands for N IID *replicates* of the enclosed model (Buntine 1994)
- Can we reason across objects?
  - e.g., the rating of a paper authored by F. Crick given the ratings of some papers authored by J. Watson

Shortcomings of Bayes Net

- **BNs lack the concept of an object**
  - Cannot represent general rules about the relations between multiple similar objects
  - For example, if we wanted to represent the probabilities over multiple papers, authors, and journals:
    - We would need an explicit random variable for each paper/author/journal
    - The distributions would be separate, so knowledge about one wouldn't impart any knowledge about the others
- **BNs assume domain closure, unique name, and relational invariance**
  - Can not represent open possible world with unknown number of objects
  - Can not accommodate objects possibly with multiple names
  - Can not succinctly represent uncertainty in data association
- **…**

Statistical Relational Learning

- In general, SRL combines logic and probabilities
- Historically, there are two general threads of research

  1. Frame-based Probabilistic Models
     - Probabilistic Relational Models (PRMs),
     - Probabilistic Entity Relation Models (PERs),
     - Object Oriented Bayesian Networks (OOBNs)

     This thread takes graphical models or hierarchical Bayesian models and adds in some form of relational/logical representation

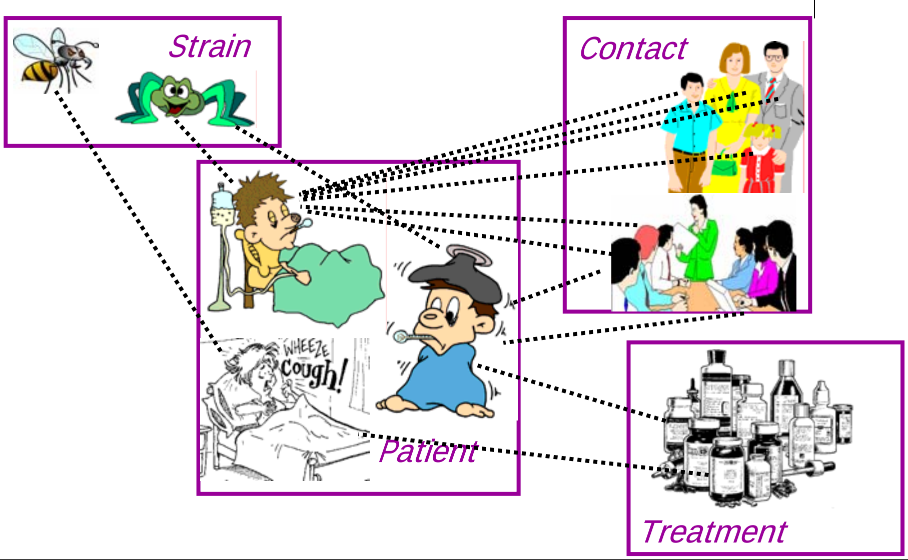
  2. First Order Probabilistic Logic (FOPL)
     - BLOGs
     - Relational Markov Logic (RML)

     This thread takes a logical representation (first-order logic, horn clauses, etc) and adds in some form of probabilities

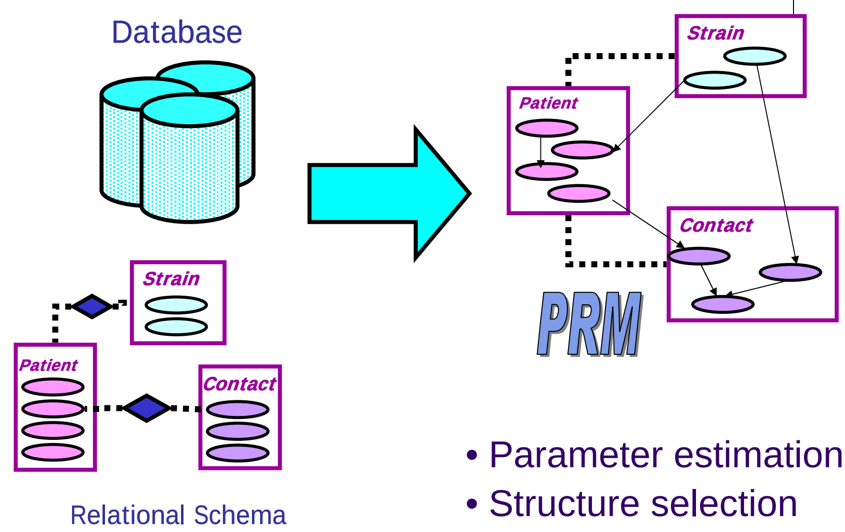
Probabilistic Relational Models (PRMs)

- Combine advantages of relational logic & Bayesian networks:
  - 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.

Motivation: Discovering Patterns in Structured Data

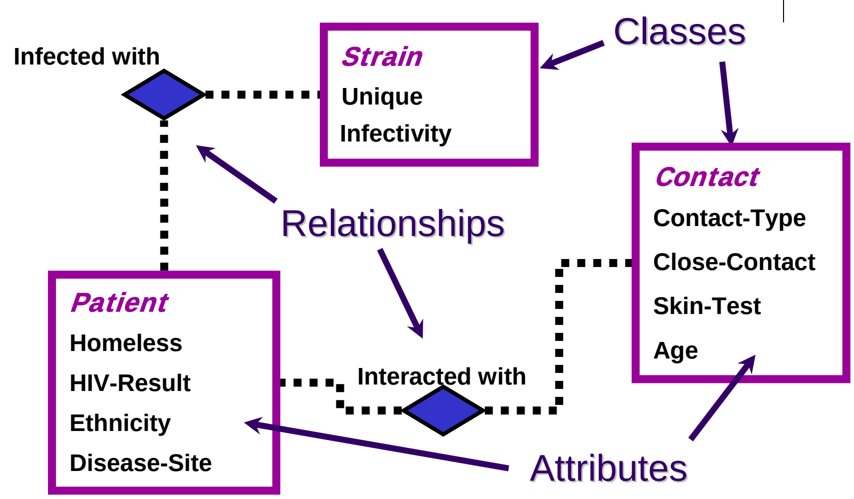


From relational database to PRM



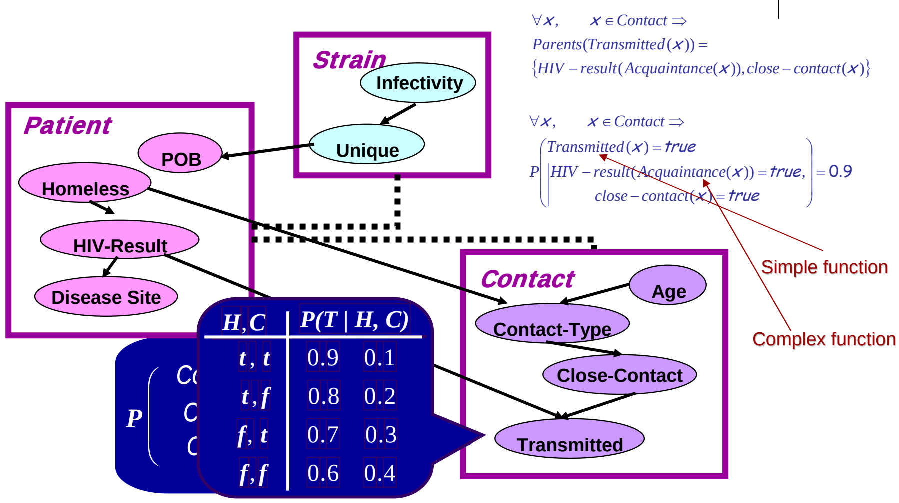
- Parameter estimation
- Structure selection

Relational Schema



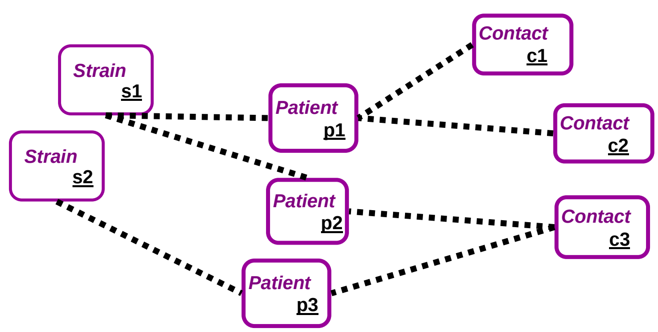
- Describes the types of objects and relations in the database

Probabilistic Relational Model



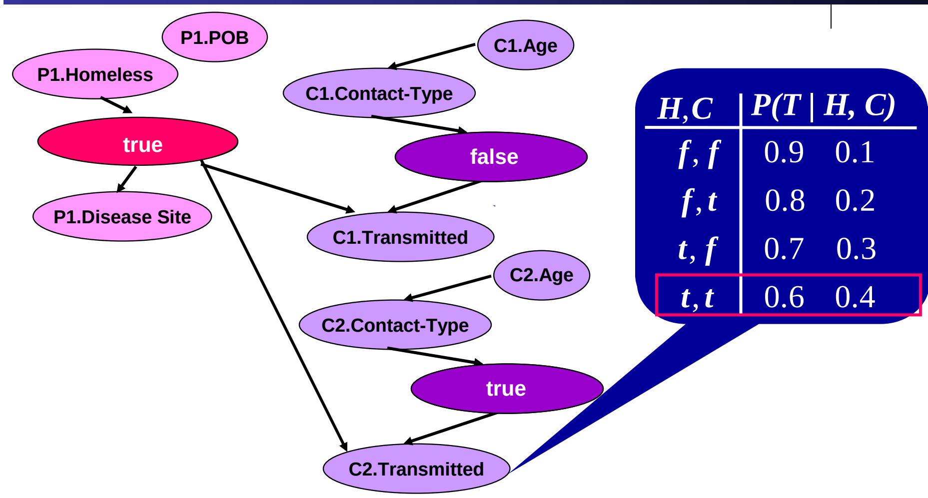
- Complex functions specifies complex relations among objects

Relational Skeleton



- Fixed relational skeleton σ
  - set of objects in each class
  - relations between them
- Uncertainty over assignment of values to attributes (AU)
- PRM defines distribution over instantiations of attributes

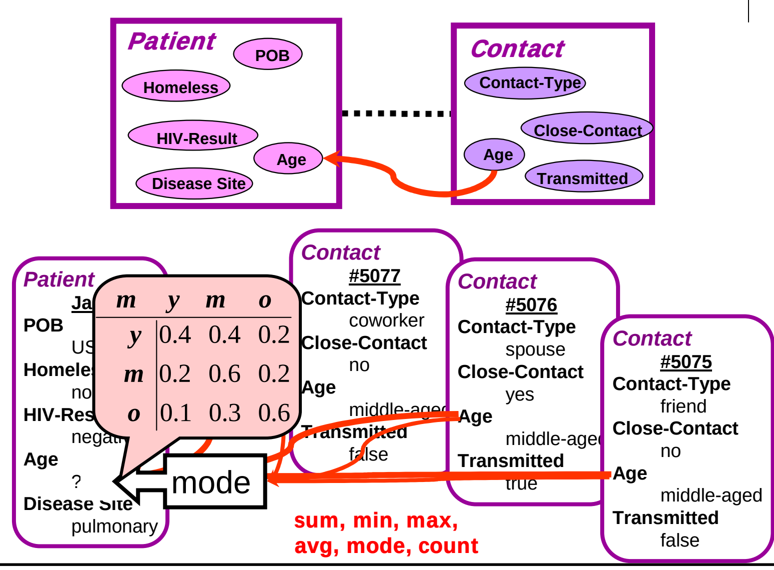
A Portion of the BN



H,C	$P(T \mid H, C)$	
f, f	0.9	0.1
f, t	0.8	0.2
t, f	0.7	0.3
t, t	0.6	0.4

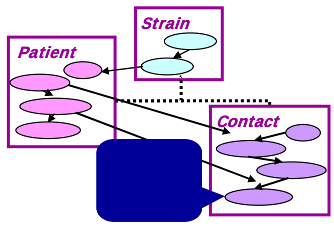
- A PRM w/ AU and fixed, valid relations is equivalent to an unrolled BN

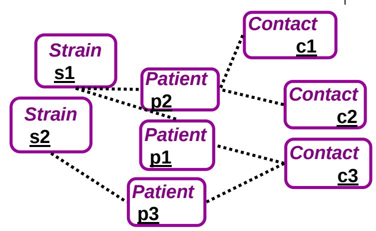
PRM: Aggregate Dependencies



	y	m	o
y	0.4	0.4	0.2
m	0.2	0.6	0.2
o	0.1	0.3	0.6

Semantics of PRM with AU





PRM + relational skeleton σ

= probability distribution over completions I:

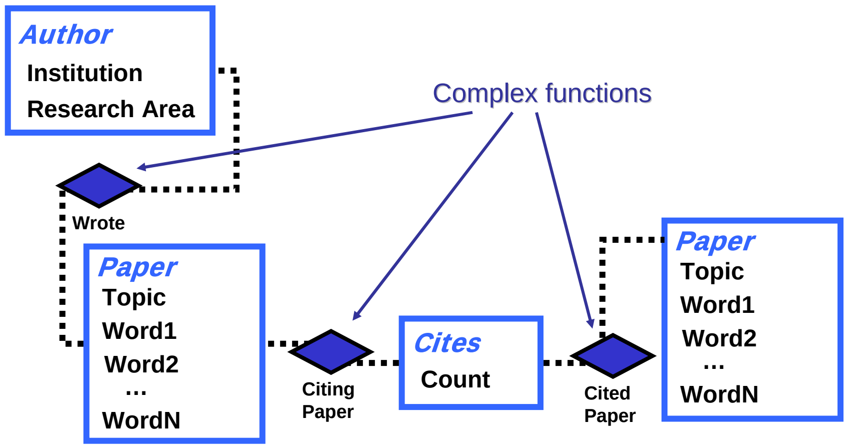
$$P(I \mid \sigma, S, \Theta) = \prod_{x \in \sigma} \prod_{x.A} P(x.A \mid parents_{S,\sigma}(x.A))$$

Objects Attributes

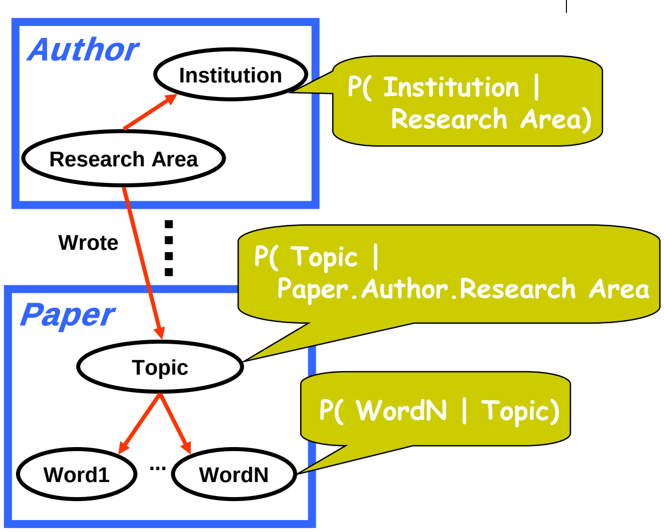
Structural Uncertainty

- Motivation: relational structure provides useful information for density estimation and prediction
- PRM w/ AU applicable only in domains where we have full knowledge of the relational structure
- Construct probabilistic models of relational structure that capture structural uncertainty
  - Applicable in cases where we do not have full knowledge of relational structure
  - Incorporating uncertainty over relational structure into probabilistic model can improve predictive accuracy
- Two new mechanisms:
  - Reference uncertainty (RU)
  - Existence uncertainty (EU)

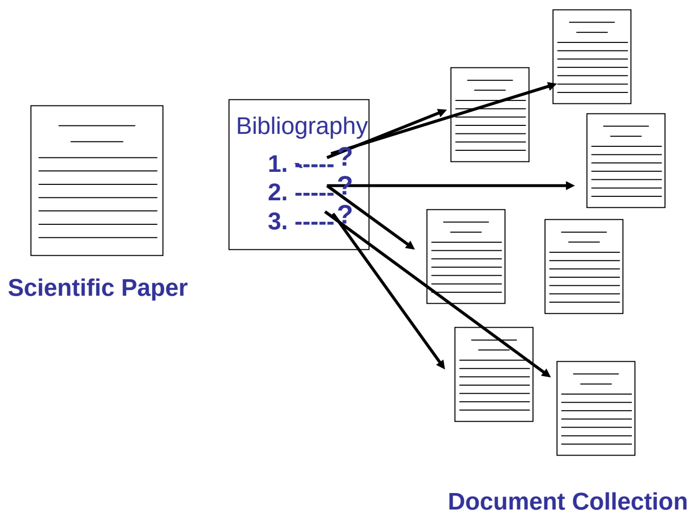
Citation Relational Schema



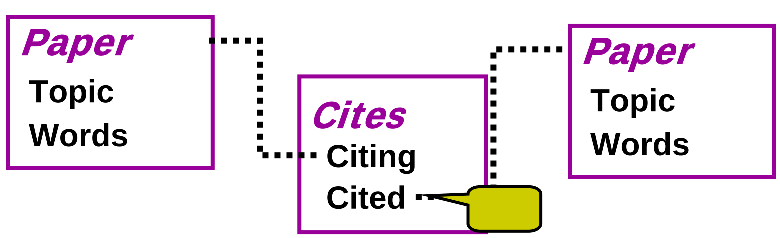
Attribute Uncertainty



Reference Uncertainty

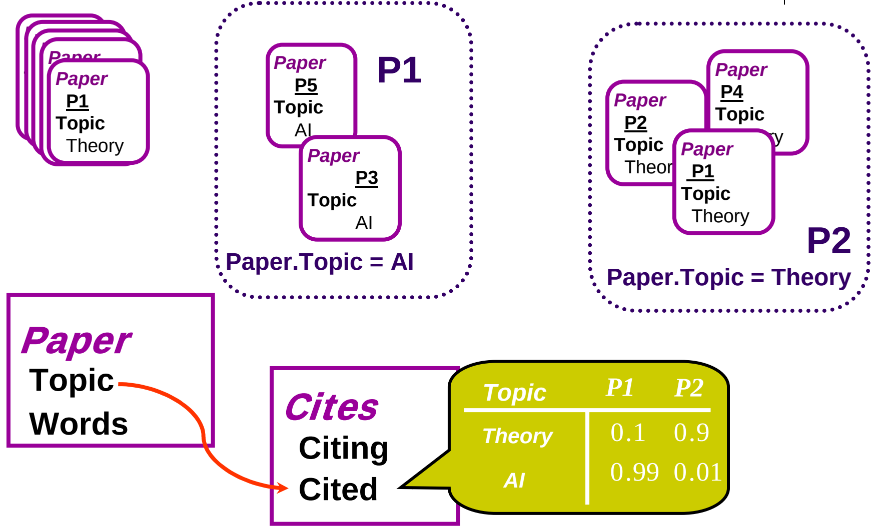


PRM w/ Reference Uncertainty

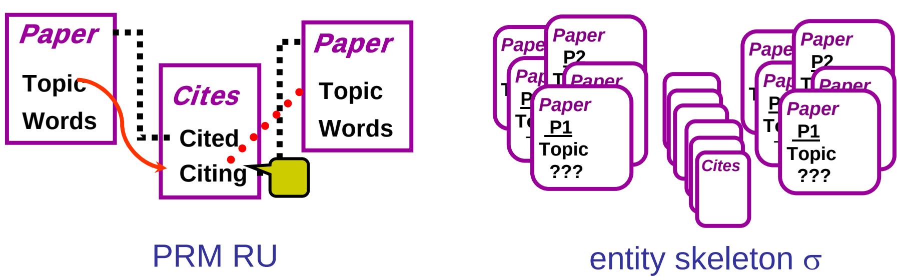


- *Dependency model for foreign keys* (i.e., complex functions)
- Define semantics for uncertainty over foreign-key values
- Naïve Approach: multinomial over primary key
  - noncompact
  - limits ability to generalize

Modeling Reference Uncertainty



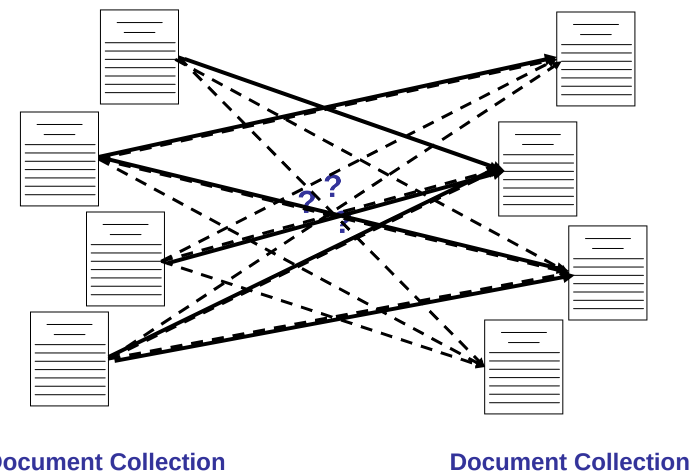
Semantics of PRMs w/ RU



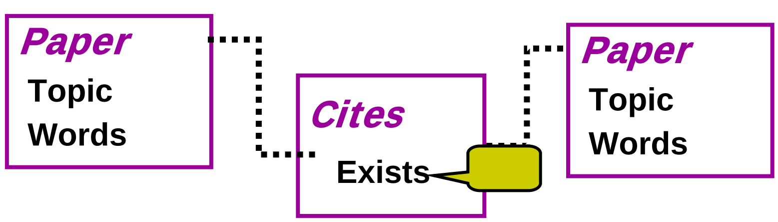
PRM-RU + entity skeleton σ

⇒ probability distribution over full instantiations I

Existence Uncertainty

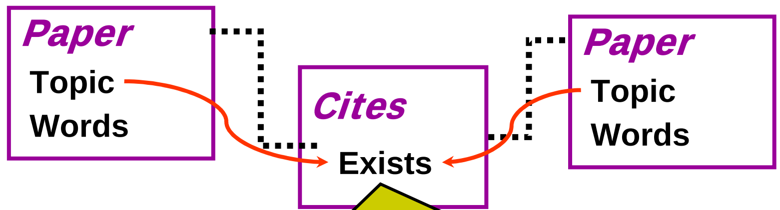


PRM w/ Exists Uncertainty



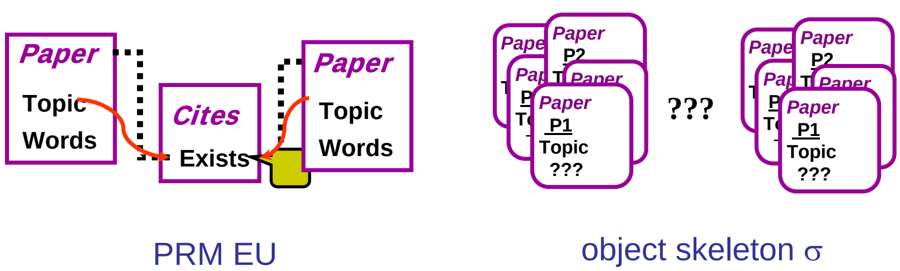
Dependency model for existence of relationship

Exists Uncertainty Example



Citer.Topic	Cited.Topic	*False*	*True*
Theory	*Theory*	0.995	0005
Theory	*AI*	0.999	0001
AI	*Theory*	0.997	0003
AI	*AI*	0.993	0008

Semantics of PRMs w/ EU



PRM-EU + *object skeleton σ*

⇒ probability distribution over full instantiations I

More extensions

- In PRM, all instances of the same class must use the same dependency mode, it cannot distinguish:
  - documentaries and sitcoms
- PRM cannot have dependencies that are "cyclic"
  - ranking for Frasier depends on ranking for Friends
- PRMs w/ Class Hierarchies
  - Refine a "heterogenous" class into more coherent subclasses
  - Refine probabilistic model along class hierarchy
    - Can specialize/inherit CPDs
    - Construct new dependencies that were originally "acyclic"
    - *Provides bridge from class-based to instance-based model*
- Undirected relational models

Inference in Unrolled BN

- Prediction requires inference in "unrolled" network
  - Infeasible for large networks
  - Use approximate inference for E-step
- Loopy belief propagation (Pearl, 88; McEliece, 98)
  - Scales linearly with size of network
  - Guaranteed to converge only for polytrees
  - Empirically, often converges in general nets (Murphy,99)
- Local message passing
  - Belief messages transferred between related instances
  - Induces a natural "influence" propagation behavior
    - Instances give information about related instances
- MCMC (Russell group)
  - Instantiate structures and models by sampling

Learning PRMs

- Training set consists of a fully specified instance: a set of objects, the relations between them, and the values of all attributes
  - In other words, a database!
- As in BNs, we split into two problems:
  - Given a dependency structure *S*, estimate the the conditional probability distribution at each node (*parameter estimation*)
  - Select the best dependency structure (*structure learning*)
    - legal models (e.g., acyclic)
    - scoring models (e.g., Bayesian …)
    - searching model space (e.g., hill climbing or heuristic search with special operators)

General Relational Models

- The most general relational model: the world consists of objects and relations over them
- **First order logic** is perhaps the most basic relational setting:
  - Syntax
    - Constants and quantified variables (representing objects)
    - Predicates (representing relations), stated in terms of constants and variables, composed with logical connectives
    - Functions specifies relations hold among objects/observations
  - Semantics:
    - Set of possible worlds, one for each possible extent of each relation

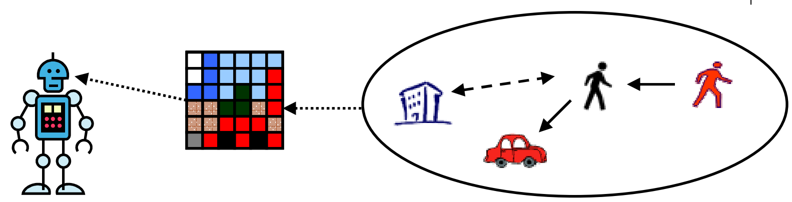
Limitations of PRMs

- PRMs as currently defined cannot represent uncertainty in general FOL
  - The basic model cannot represent uncertainty about whether or not a relation exists between a given tuple of objects
- Even when we add "structural uncertainty" as proposed PRMs are too specialized
  - The probability of a relation between objects would conditioned on the values of some of their attributes, not on their participation in other relations

BLOG Approach

- BLOG model defines probability distribution over model structures of a typed first-order language
  [Gaifman 1964; Halpern 1990]
- Unique distribution, not just constraints on the distribution

Basic Task



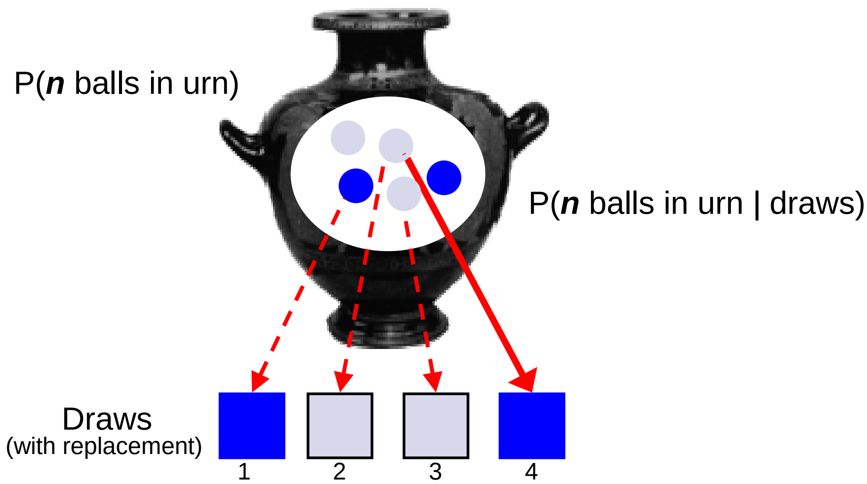
- Given observations, make inferences about underlying objects
- Difficulties:
  - Don't know list of objects in advance
  - Don't know when same object observed twice

(identity uncertainty / data association / record linkage)

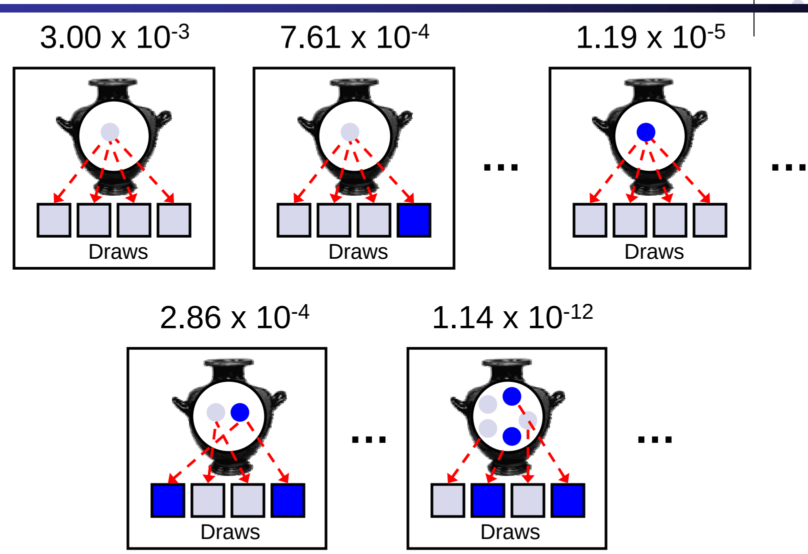
Handling Unknown Objects

- Standard practice: special-purpose algorithms to resolve identity uncertainty
  - E.g., in PRM, we can remunerate all possible identity of an object and model their associations as "uncertain relations"
  - This is very cumbersome and inflexible
- Goal: Resolve identity uncertainty by inference in probabilistic model
- Bayesian LOGic (BLOG): representation language for models with
  - Unknown set of objects
  - Unknown map from observations to objects

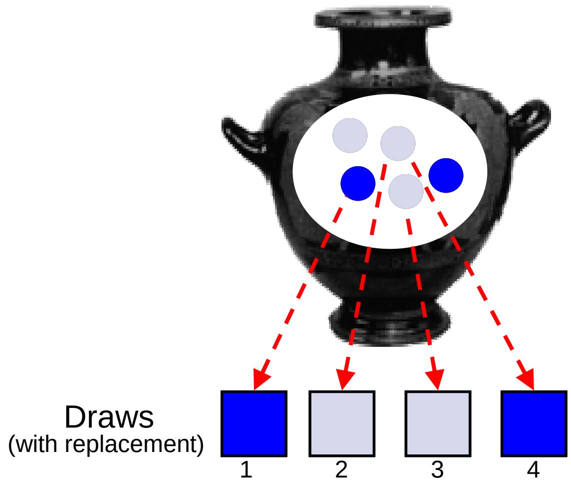
Simple Example: Balls in an Urn



Possible Worlds



Generative Process for Possible Worlds



BLOG Model for Urn and Balls

```
type Color;  type Ball;  type Draw;

random Color TrueColor(Ball);
random Ball BallDrawn(Draw);
random Color ObsColor(Draw);

guaranteed Color Blue, Green;
guaranteed Draw Draw1, Draw2, Draw3, Draw4;

#Ball ~ Poisson[6]();

TrueColor(b) ~ TabularCPD[[0.5, 0.5]]();

BallDrawn(d) ~ UniformChoice({Ball b});

ObsColor(d)
    if (BallDrawn(d) != null) then
        ~ NoisyCopy(TrueColor(BallDrawn(d)));
```

BLOG Model for Urn and Balls

```
type Color;  type Ball;  type Draw;

random Color TrueColor(Ball);
random Ball BallDrawn(Draw);
random Color ObsColor(Draw);

guaranteed Color Blue, Green;
guaranteed Draw Draw1, Draw2, Draw3, Draw4;

#Ball ~ Poisson[6]();

TrueColor(b) ~ TabularCPD[[0.5, 0.5]]();

BallDrawn(d) ~ UniformChoice({Ball b});

ObsColor(d)
    if (BallDrawn(d) != null) then
        ~ NoisyCopy(TrueColor(BallDrawn(d)));
```

header

number statement

dependency statements

BLOG Model for Urn and Balls

```
type Color;  type Ball;  type Draw;

random Color TrueColor(Ball);
random Ball BallDrawn(Draw);
random Color ObsColor(Draw);

guaranteed Color Blue, Green;

TrueColor(b) ~ TabularCPD[[0.5, 0.5]]();

BallDrawn(d) ~ UniformChoice({Ball b});

ObsColor(d)
    if (BallDrawn(d) != null) then
        ~ NoisyCopy(TrueColor(BallDrawn(d)));
```

Identity uncertainty: BallDrawn(Draw1) $\stackrel{?}{=}$ BallDrawn(Draw2)

BLOG Model for Urn and Balls

```
type Color;  type Ball;  type Draw;

random Color TrueColor(Ball);
random Ball BallDrawn(Draw);
random Color ObsColor(Draw);

guaranteed Color Blue, Green;
guaranteed Draw Draw1, Draw2, Draw3, Draw4;

#Ball ~ Poisson[6]();

TrueColor(b) ~ TabularCPD[[0.5, 0.5]]();

BallDrawn(d) ~ UniformChoice({Ball b});

ObsColor(d)
    if (BallDrawn(d) != null) then
        ~ NoisyCopy(TrueColor(BallDrawn(d)));
```

Arbitrary conditional probability distributions

CPD arguments

BLOG Model for Urn and Balls

```
type Color;  type Ball;  type Draw;

random Color TrueColor(Ball);
random Ball BallDrawn(Draw);
random Color ObsColor(Draw);

guaranteed Color Blue, Green;
guaranteed Draw Draw1, Draw2, Draw3, Draw4;

#Ball ~ Poisson[6]();

TrueColor(b) ~ TabularCPD[[0.5, 0.5]]();

BallDrawn(d) ~ UniformChoice({Ball b});

ObsColor(d)
    if (BallDrawn(d) != null) then
        ~ NoisyCopy(TrueColor(BallDrawn(d)));
```

Context-specific dependence

BLOG Model for Urn and Balls

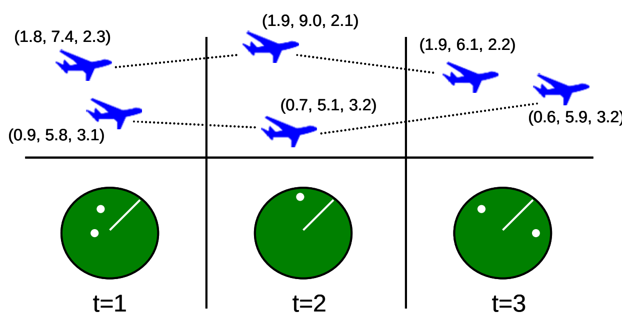
```
type Color;  type Ball;  type Draw;

random Color TrueColor(Ball);
random Ball BallDrawn(Draw);
random Color ObsColor(Draw);

guaranteed Color Blue, Green;
guaranteed Draw Draw1, Draw2, Draw3, Draw4;

#Ball ~ Poisson[6]();

TrueColor(b) ~ TabularCPD[[0.5, 0.5]]();

BallDrawn(d) ~ UniformChoice({Ball b});

ObsColor(d)
    if (BallDrawn(d) != null) then
        ~ NoisyCopy(TrueColor(BallDrawn(d)));
```

Declarative Semantics

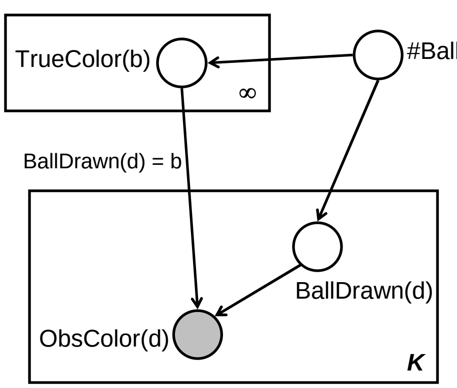
- What is the set of possible worlds?
- What is the probability distribution over worlds?

What Exactly Are the Objects?

- Objects are tuples that encode generation history
- Aircraft: (Aircraft, 1), (Aircraft, 2), …
- Blip from (Aircraft, 2) at time 8:
  (Blip, (Source, (Aircraft, 2)), (Time, 8), 1)



Graphical Representation of BLOG Model



- Like a BN, but:
  - Edges are only active in certain contexts
  - Ignoring contexts, ObsColor(d) has infinitely many parents
  - In other models, graph may be cyclic if you ignore contexts

Basic Random Variables (RVs)

- For each number statement and tuple of generating objects, have RV for number of objects generated
- For each function symbol and tuple of arguments, have RV for function value
- Lemma: Full instantiation of these RVs uniquely identifies a possible world

Probability Distribution

- BLOG model specifies:
  - Conditional distributions for basic RVs
  - Factorization properties for certain finite instantiations of basic RVs
- Theorem: Under certain conditions (analogous to BN acyclicity), every BLOG model defines unique distribution over possible worlds

Inference

- Does infinite set of basic RVs prevent inference?
- No: Sampling algorithm only needs to instantiate finite set of relevant variables
- Algorithms:
  - Rejection sampling [Milch *et al.*, IJCAI 2005]
  - Guided likelihood weighting [Milch *et al.*, AI/Stats 2005]
- Theorem: For large class of BLOG models, sampling algorithms converge to correct probability for any query, using finite time per sampling step

Summary: Distributions over First-Order Structures

- Idea goes back to Gaifman [1964]
- Halpern [1990] defines language for stating constraints on such distributions
  - But not specifying a distribution uniquely
- Logic programming approaches [Poole 1993; Sato & Kameya 2001; Kersting & De Raedt 2001] define unique distributions, but assume unique names and domain closure
- PRMs [Koller & Pfeffer 1998] have special constructs for number uncertainty, existence uncertainty
- BLOG: Unified syntax for distributions over worlds with:
  - Varying sets of objects
  - Varying mappings from observations to objects

  See also MEBN (Multi- Entity Bayesian Networks) [Laskey and da Costa, UAI 2005]