Learning and Structural Uncertainty in Relational Probability Models

Brian Milch

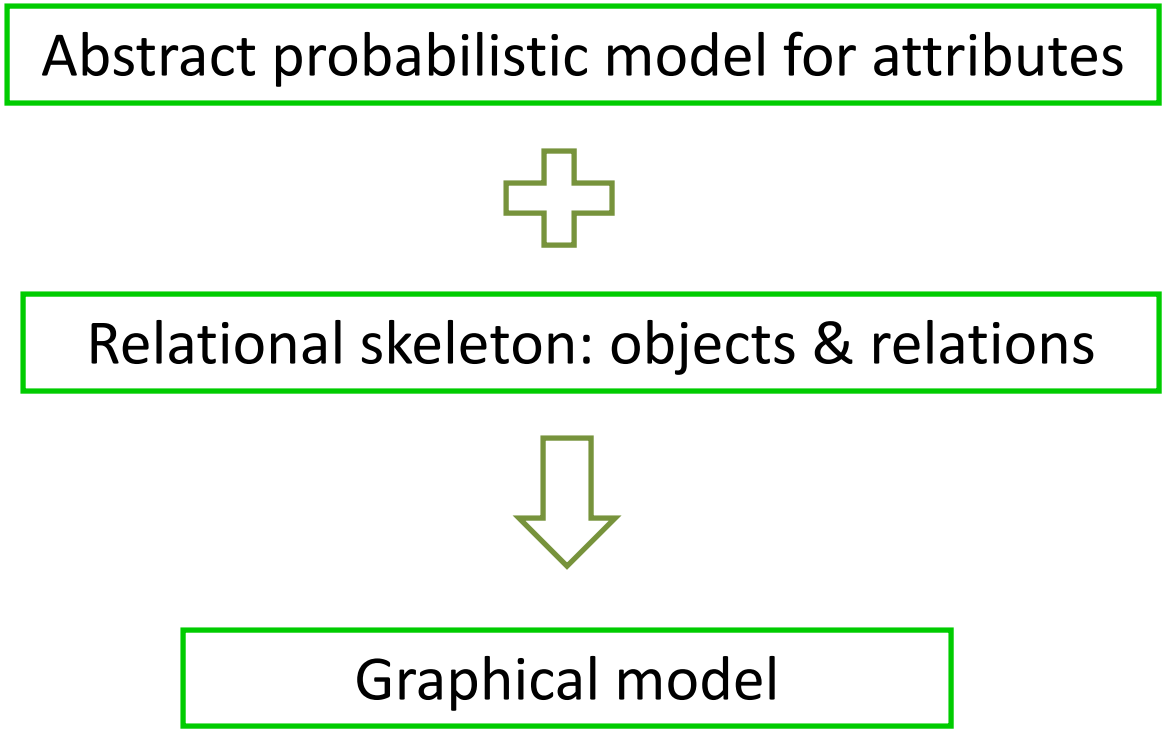
MIT 9.66

November 29, 2007

Outline

- Learning relational probability models
- Structural uncertainty
  - Uncertainty about relations
  - Uncertainty about object existence and identity
- Applications of BLOG

Review: Relational Probability Models



Reivew: Dependency Statements

```
                                  BNs    RL    Theory
Specialty(r) ~ TabularCPD[[0.5, 0.3, 0.2]];

                              BNs     RL     Theory
Topic(p) ~ TabularCPD[[0.90, 0.01, 0.09],     | BNs
                      [0.02, 0.85, 0.13],     | RL
                      [0.10, 0.10, 0.80]]     | Theory
                (Specialty(FirstAuthor(p)));

                                  the          Bayesian   reinforcement
WordAt(wp) ~ TabularCPD[[0.03,..., 0.02, 0.001,...],  | BNs
                        [0.03,..., 0.001, 0.02,...],  | RL
                        [0.03,..., 0.003, 0.003,...]] | Theory
                (Topic(Doc(wp)));
```

Learning

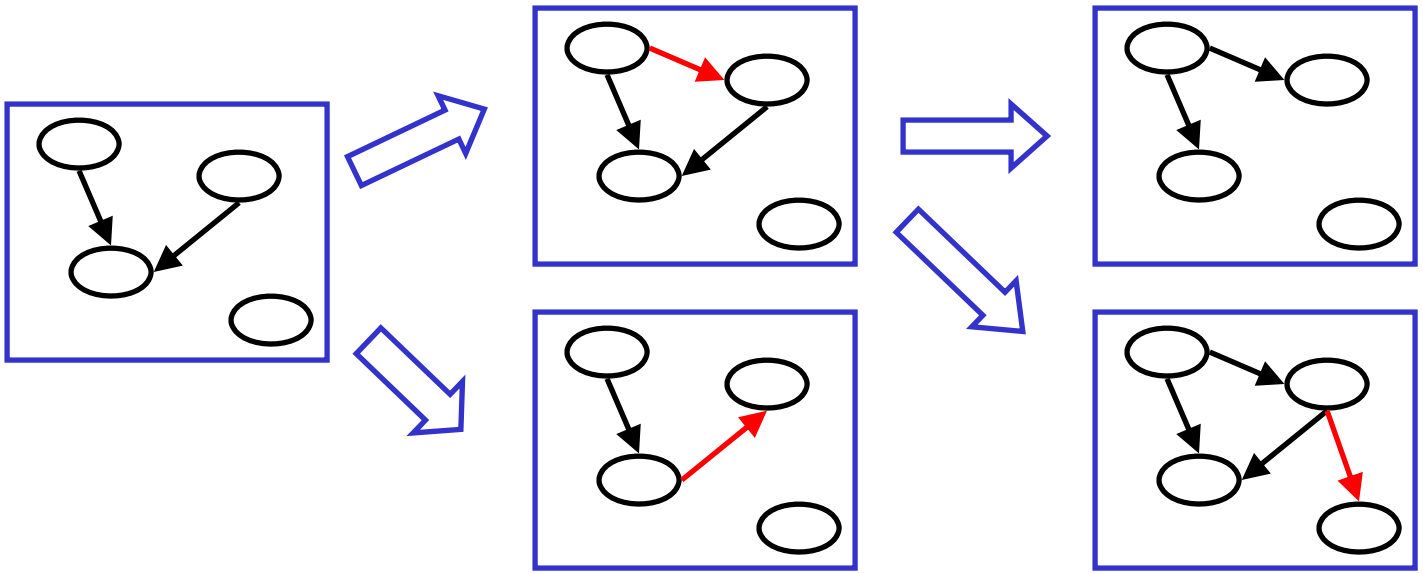
- Assume types, functions are given
- Straightforward task: given structure, learn parameters
  - Just like in BNs, but parameters are shared across variables for same function, e.g., Topic(Smith98a), Topic(Jones00), etc.
- Harder: learn abstract dependency structure

Structure Learning for BNs

- Find BN structure M that maximizes

$$p(M \mid \text{data}) \propto p(M) \int p(\text{data} \mid M, \theta)\, p(\theta \mid M)\, d\theta$$

- Greedy local search over structures
  - Operators: add, delete, reverse edges
  - Exclude cyclic structures



Logical Structure Learning

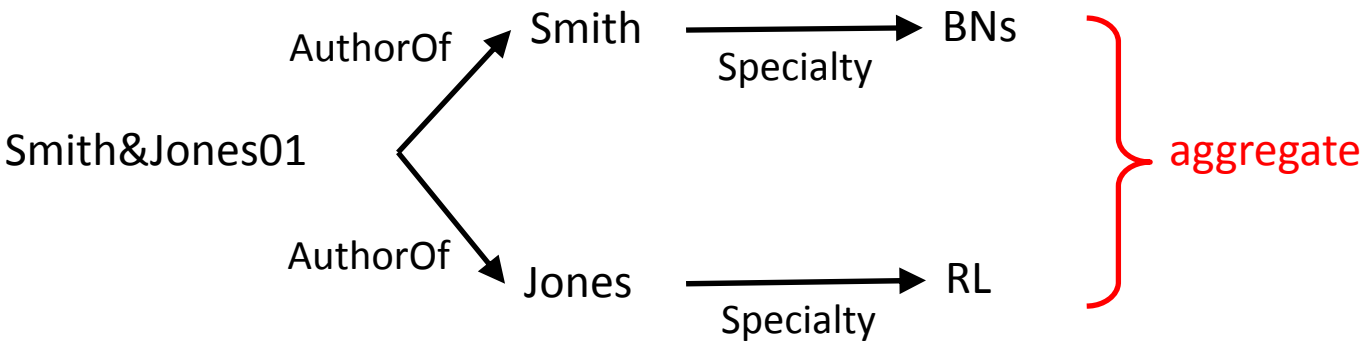
- In RPM, want logical specification of each node's parent set
- Deterministic analogue: inductive logic programming (ILP)
  [Dzeroski & Lavrac 2001; Flach and Lavrac 2002]
- Classic work on RPMs by Friedman, Getoor, Koller & Pfeffer [1999]
  - We'll call their models FGKP models
    (they call them "probabilistic relational models" (PRMs))

FGKP Models

- Each dependency statement has form:

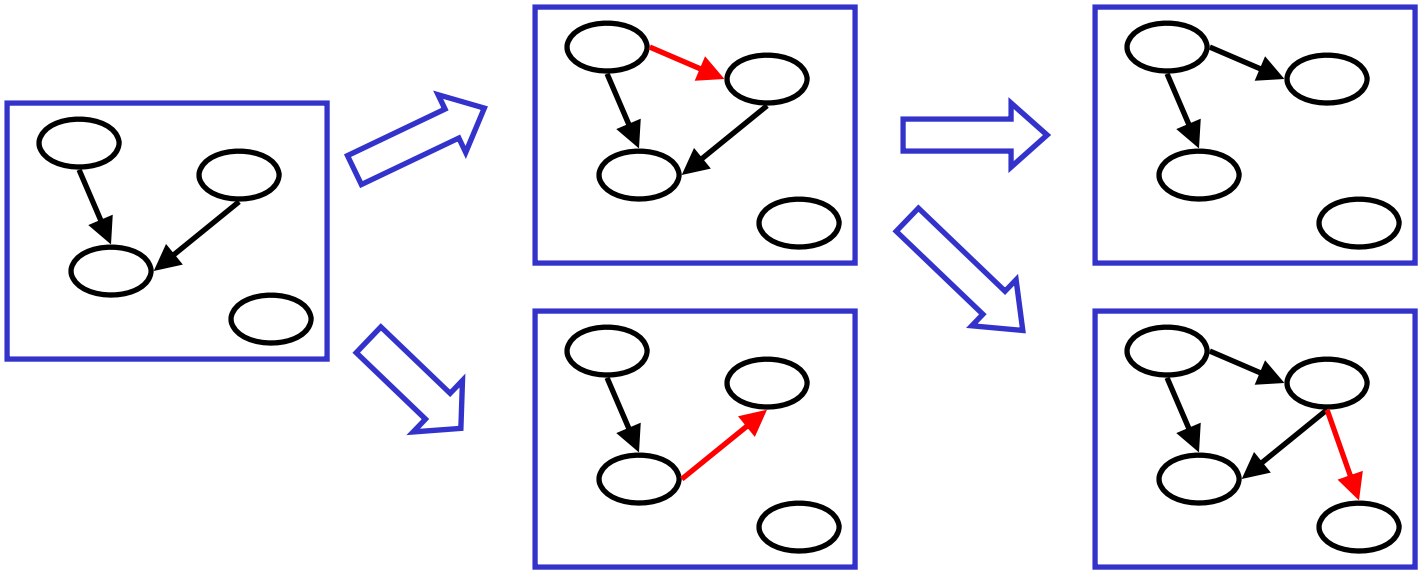
  `Func(x) ~ TabularCPD[...](`s_1`,..., `s_k`)`

  where $s_1,...,s_k$ are slot chains
- Slot chains
  - Basically logical terms: Specialty(FirstAuthor(p))
  - But can also treat predicates as "multi-valued functions": Specialty(AuthorOf(p))



Structure Learning for FGKP Models

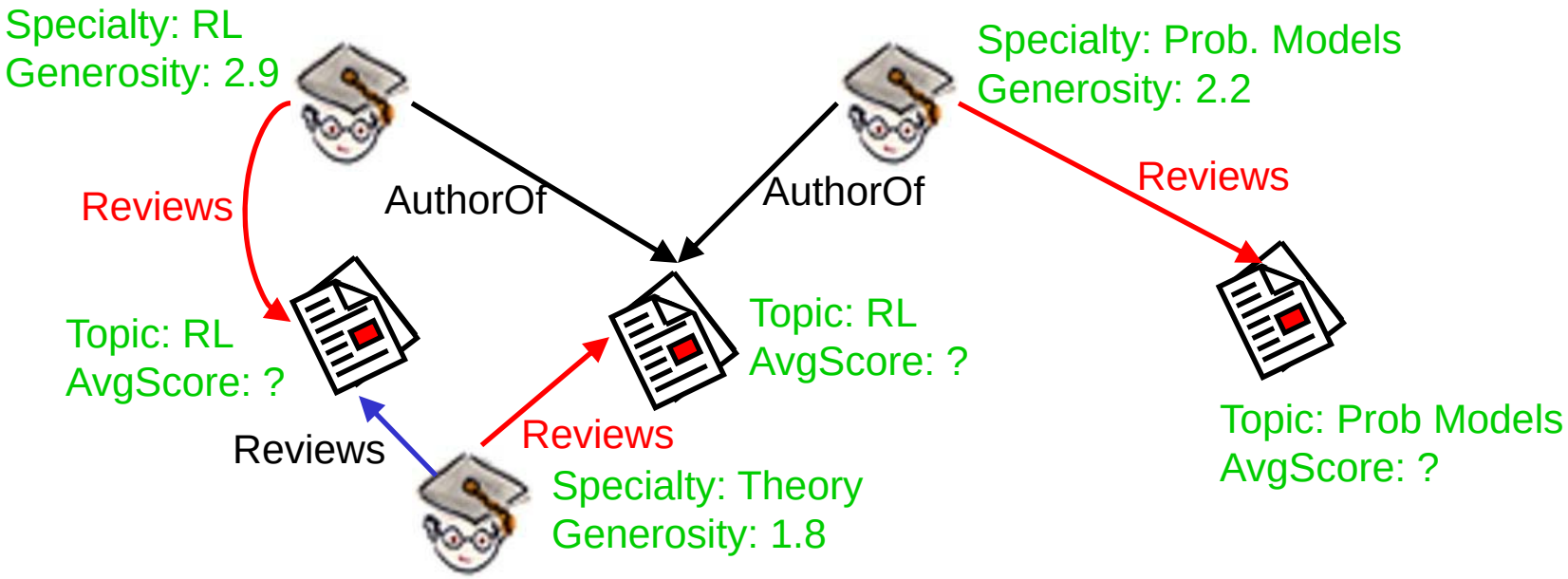
- Greedy search again
  - But add or remove whole slot chains
  - Start with chains of length 1, then 2, etc.
  - Check for acyclicity using symbol graph



Outline

- Learning relational probability models
- Structural uncertainty
  - Uncertainty about relations
  - Uncertainty about object existence and identity
- Applications of BLOG

Relational Uncertainty: Example



- Questions: Who will review my paper, and what will its average review score be?

Simplest Approach to Relational Uncertainty

- Add predicate Reviews(r, p)
- Can model this with existing syntax:

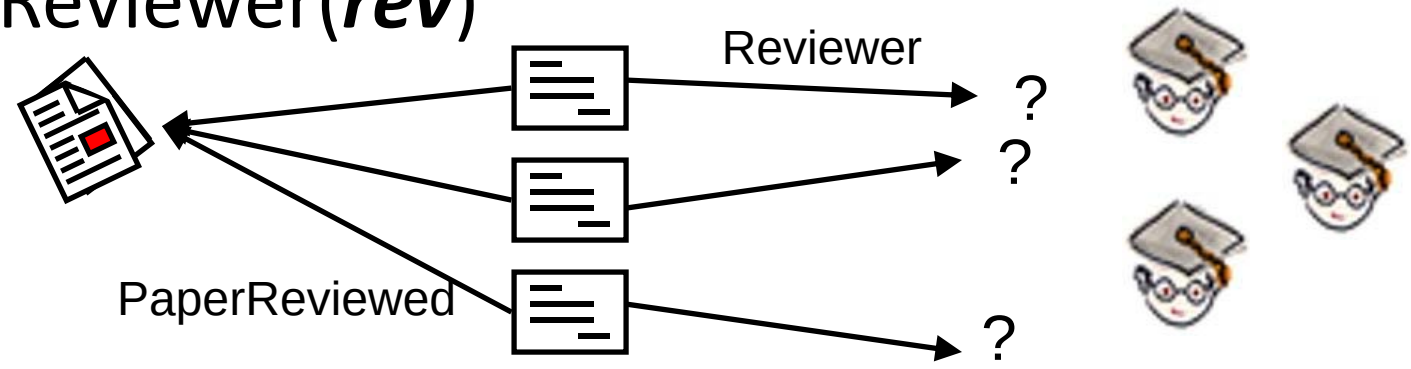
```
Reviews(r, p) ~ ReviewCPD(Specialty(r), Topic(p));
```

- Potential drawback:
  - Reviews(r, p) nodes are independent given specialties and topics
  - Expected number of reviews per paper grows with number of researchers in skeleton

[Getoor et al., JMLR 2002]

Another Approach: Reference Uncertainty

- Say each paper gets ***k*** reviews
  - Can add Review objects to skeleton
  - For each paper ***p***, include ***k*** review objects ***rev*** with PaperReviewed(***rev***) = ***p***
- Uncertain about values of function Reviewer(***rev***)



[Getoor et al., JMLR 2002]

Models for Reviewer(***rev***)

- Explicit distribution over researchers?
  - No: won't generalize across skeletons
- Selection models:
  - Uniform sampling from researchers with certain attribute values [Getoor *et al*., JMLR 2002]
  - Weighted sampling, with weights determined by attributes [Pasula *et al*., IJCAI 2001]

Choosing Reviewer Based on Specialty

```
ReviewerSpecialty(rev) ~ SpecSelectionCPD
                                (Topic(PaperReviewed(rev)));

Reviewer(rev) ~ Uniform({Researcher r :
                  Specialty(r) = ReviewerSpecialty(rev)});
```

Context-Specific Dependencies

```
RevScore(rev) ~ ScoreCPD(Generosity(Reviewer(rev)));
```


```
AvgScore(p) = Mean({RevScore(rev) for Review rev :
                                    PaperReviewed(Rev) = p});
```

- Consequence of relational uncertainty: dependencies become context-specific
  - RevScore(Rev1) depends on Generosity(R1) only when Reviewer(Rev1) = R1

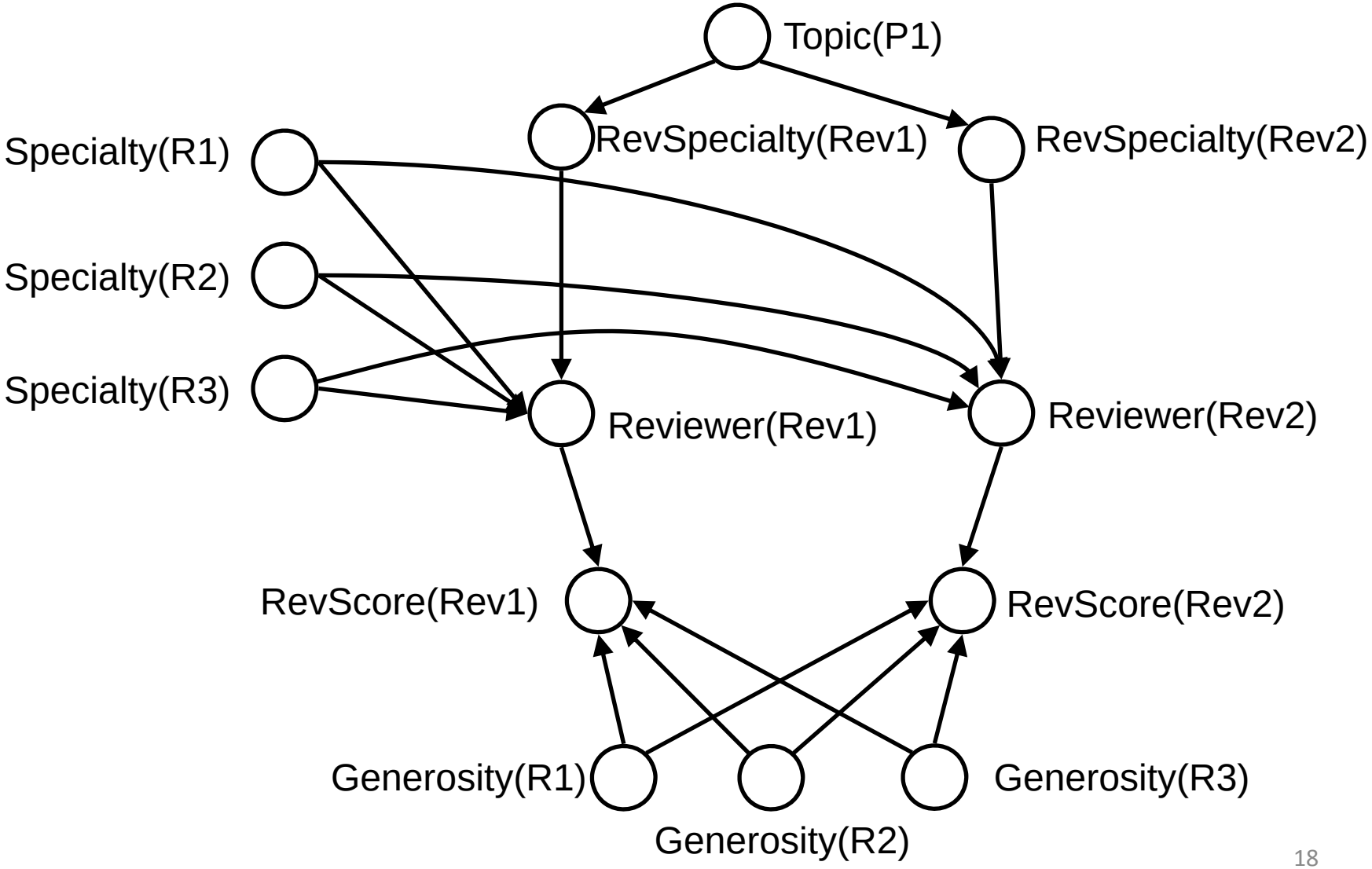
Semantics: Ground BN

- Can still define ground BN
- Parents of node ***X*** are all basic RVs whose values are potentially relevant in evaluating the right hand side of ***X***'s dependency statement
- Example: for RevScore(Rev1)...

```
RevScore(rev) ~ ScoreCPD(Generosity(Reviewer(rev)));
```

  - Reviewer(Rev1) is always relevant
  - Generosity(R) might be relevant for any researcher R

Ground BN


Topic(P1)
RevSpecialty(Rev1)
RevSpecialty(Rev2)
Specialty(R1)
Specialty(R2)
Specialty(R3)
Reviewer(Rev1)
Reviewer(Rev2)
RevScore(Rev1)
RevScore(Rev2)
Generosity(R1)
Generosity(R2)
Generosity(R3)

Inference

- Can still use ground BN, but it's often very highly connected
- Alternative: MCMC over possible worlds
  [Pasula & Russell, IJCAI 2001]
  - In each world, only certain dependencies are active

Metropolis-Hastings MCMC

- Metropolis-Hastings process: in world ω,
  - sample new world ω' from proposal distribution $\boldsymbol{q}(\omega' \mid \omega)$
  - accept proposal with probability

$$\max\left(1, \frac{p(\omega')q(\omega \mid \omega')}{p(\omega)q(\omega' \mid \omega)}\right)$$

    otherwise remain in ω
- Stationary distribution is $\boldsymbol{p}(\omega)$

Computing Acceptance Ratio Efficiently

- World probability is

$$p(\omega) = \prod_X P(X = x_\omega \mid \mathrm{pa}_\omega(X))$$

  where $\mathbf{pa}_\omega(\boldsymbol{X})$ is inst. of $\boldsymbol{X}$'s active parents in ω
- If proposal changes only $\boldsymbol{X}$, then all factors not containing $\boldsymbol{X}$ cancel in $\boldsymbol{p}(\omega)$ and $\boldsymbol{p}(\omega')$

Result: Time to compute acceptance ratio often doesn't depend on number of objects

[Pasula *et al.*, IJCAI 2001]

Learning Models for Relations

- Binary predicate approach:

```
Reviews(r, p) ~ ReviewCPD(Specialty(r), Topic(p));
```

  – Use existing search over slot chains

- Selecting based on attributes

```
ReviewerSpecialty(rev) ~ SpecSelectionCPD
                              (Topic(PaperReviewed(rev)));

Reviewer(rev) ~ Uniform({Researcher r :
                    Specialty(r) = ReviewerSpecialty(rev)});
```

  – Search over sets of attributes to look at

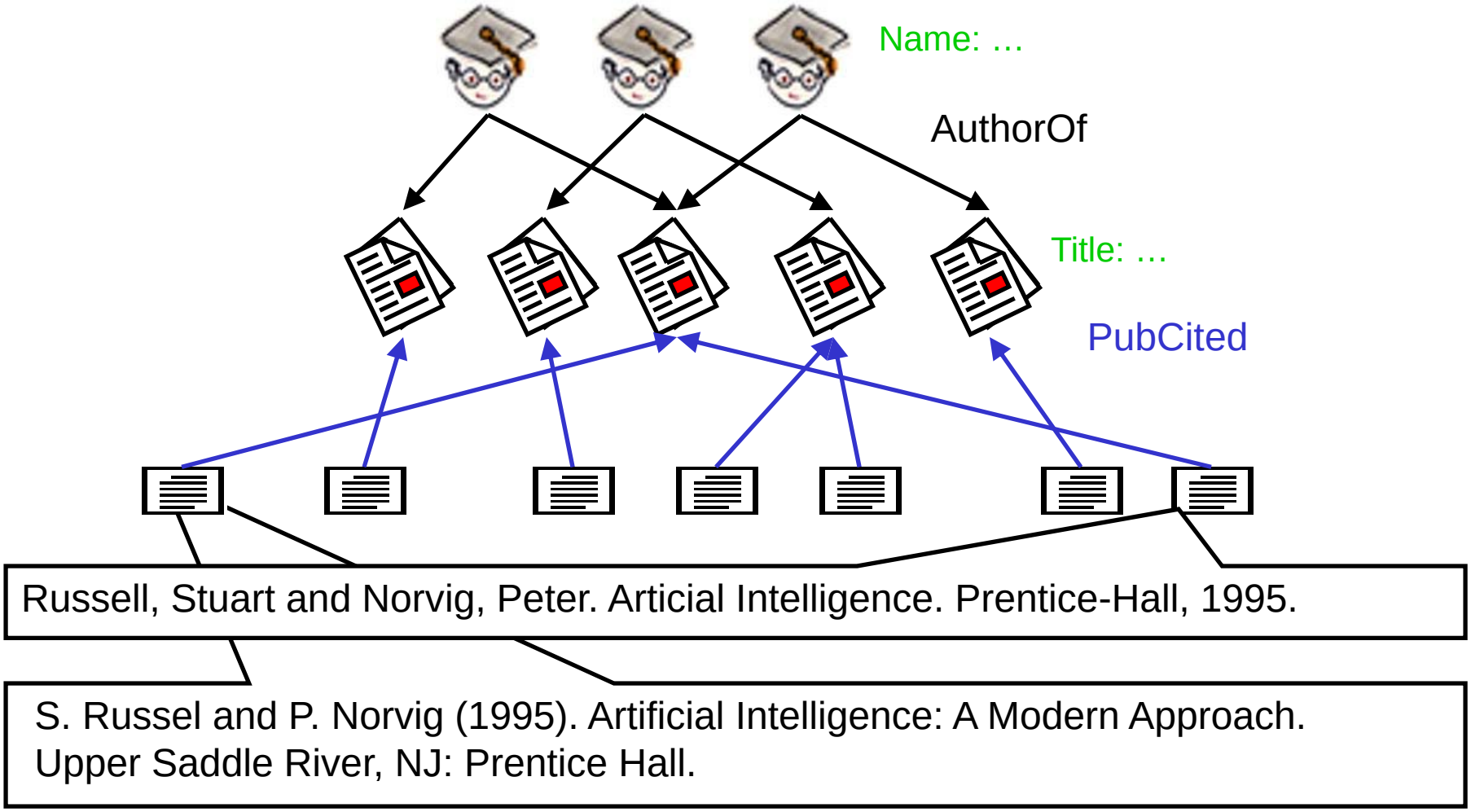
  – Search over parent slot chains for choosing attribute values

[Getoor *et al.*, JMLR 2002]

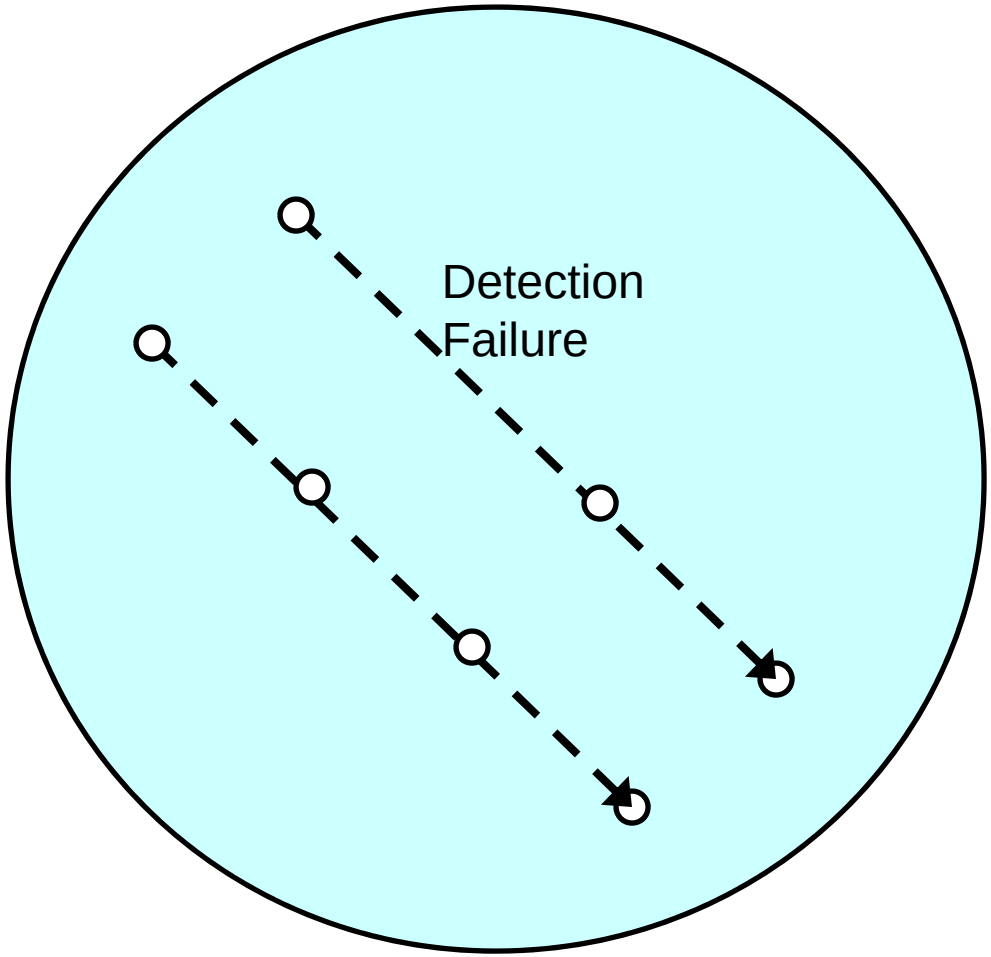
Outline

- Learning relational probability models
- Structural uncertainty
  - Uncertainty about relations
  - Uncertainty about object existence and identity
- Applications of BLOG

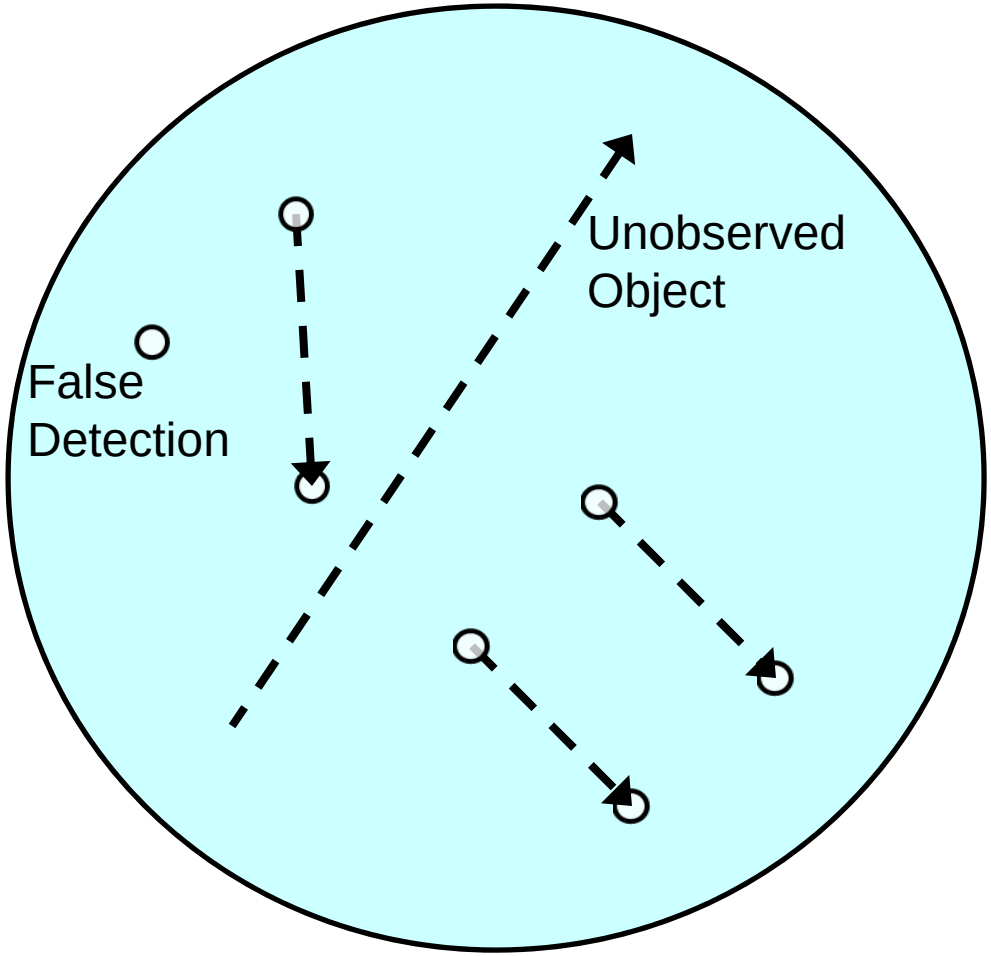
Example 1: Bibliographies



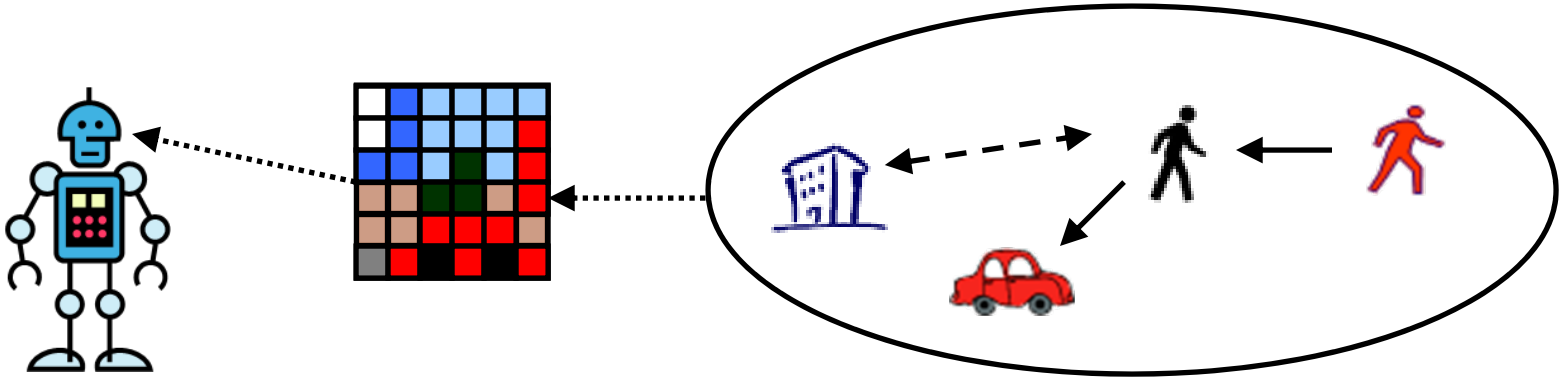
Example 2: Aircraft Tracking



Example 2: Aircraft Tracking



Handling Unknown Objects

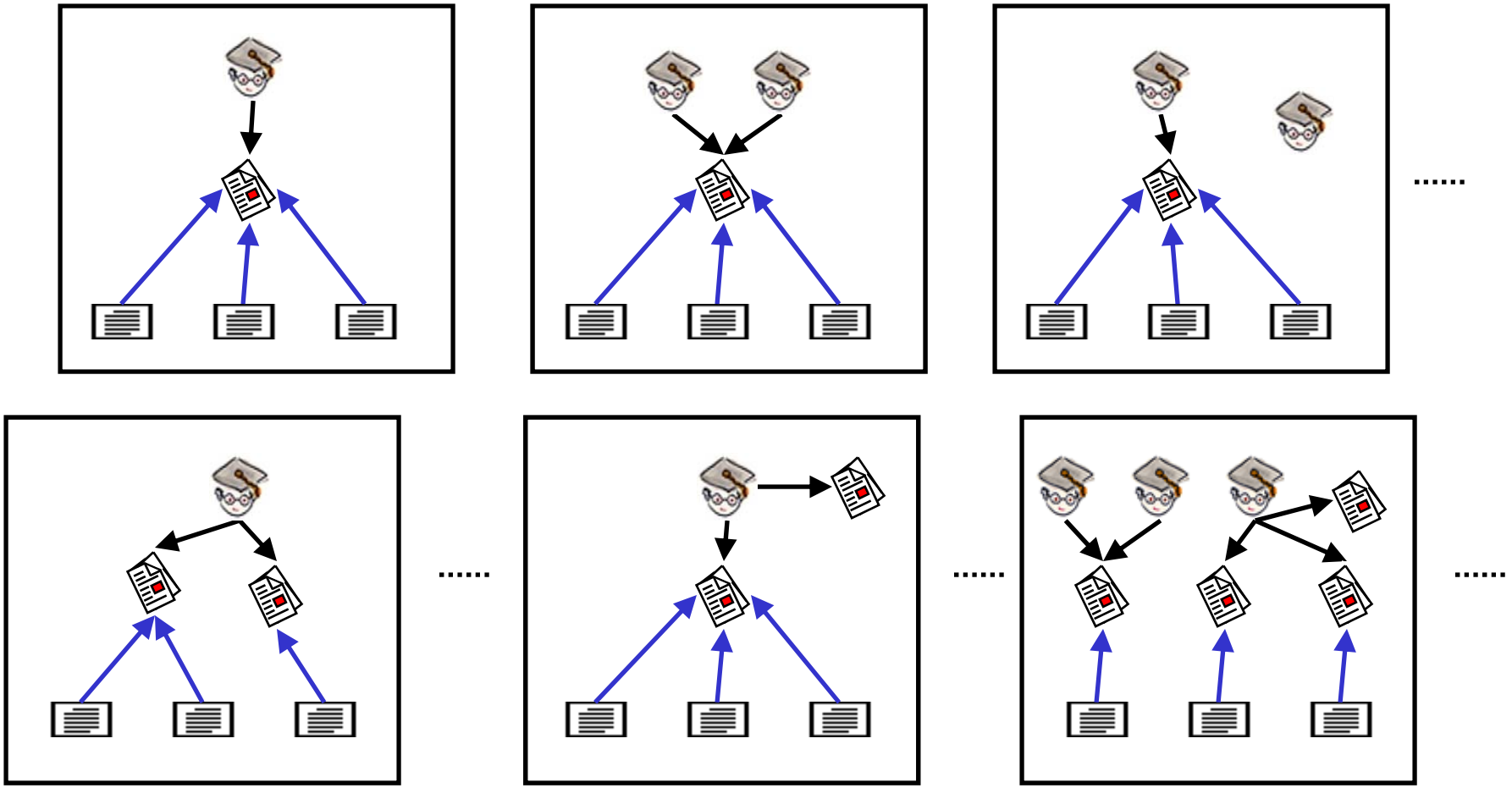


- Fundamental task: given observations, make inferences about initially unknown objects
- But most RPM languages assume set of objects is fixed and known (Herbrand models)
- Bayesian logic (BLOG) lifts this assumption

[Milch *et al.*, IJCAI 2005. See also MEBN: Laskey & da Costa, UAI 2005; Dynamical Grammars: Mjolsness & Yosiphon, AMAI to appear]

Possible Worlds

(not showing attribute values)



How can we define a distribution over such outcomes?

Generative Process

- Imagine process that constructs worlds using two kinds of steps
  - Add some objects to the world
  - Set the value of a function on a tuple of arguments

BLOG Model for Citations

```
#Paper ~ NumPapersPrior();

Title(p) ~ TitlePrior();

guaranteed Citation Cit1, Cit2, Cit3, Cit4, Cit5, Cit6, Cit7;

PubCited(c) ~ Uniform({Paper p});

Text(c) ~ NoisyCitationGrammar(Title(PubCited(c)));
```

number statement

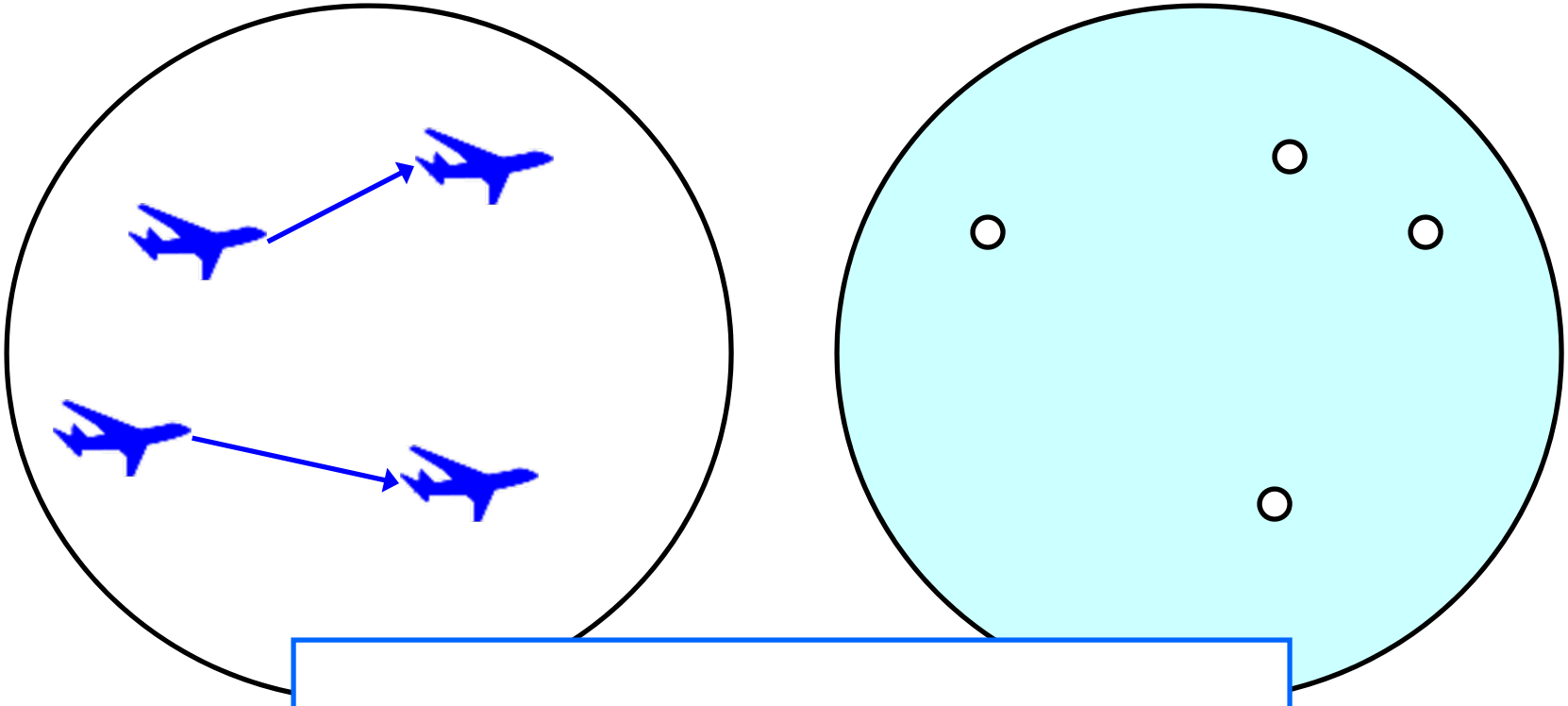
part of skeleton:
exhaustive list of distinct citations

familiar syntax for
reference uncertainty

Adding Authors

```
#Researcher ~ NumResearchersPrior();

Name(r) ~ NamePrior();

#Paper ~ NumPapersPrior();

FirstAuthor(p) ~ Uniform({Researcher r});

Title(p) ~ TitlePrior();

PubCited(c) ~ Uniform({Paper p});

Text(c) ~ NoisyCitationGrammar
        (Name(FirstAuthor(PubCited(c))), Title(PubCited(c)));
```

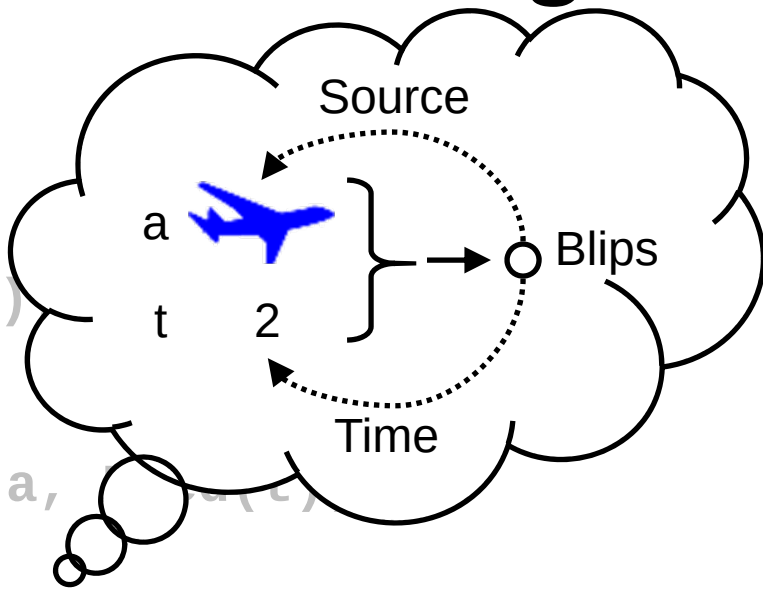
Generative Process for Aircraft Tracking

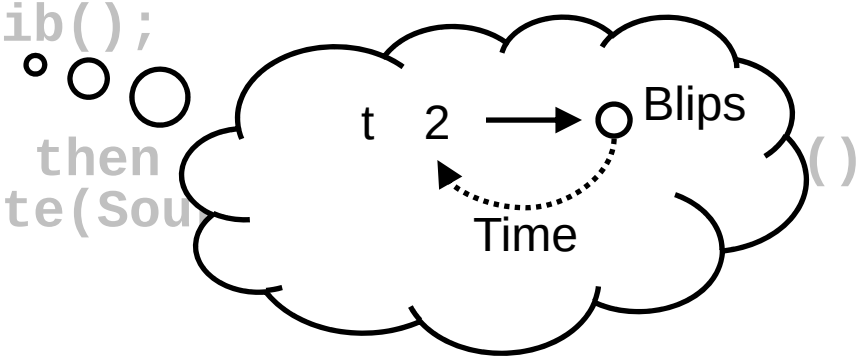


Existence of radar blips depends on existence and locations of aircraft

BLOG Model for Aircraft Tracking

```
…
#Aircraft ~ NumAircraftDistrib();

State(a, t)
  if t = 0 then ~ InitState()
  else ~ StateTransition(State(a, t-1));

#Blip(Source = a, Time = t)
  ~ NumDetectionsDistrib(State(a, t));

#Blip(Time = t)
  ~ NumFalseAlarmsDistrib();

ApparentPos(r)
  if (Source(r) = null) then ~ UniformPos()
  else ~ ObsDistrib(State(Source(r), Time(r)));
```



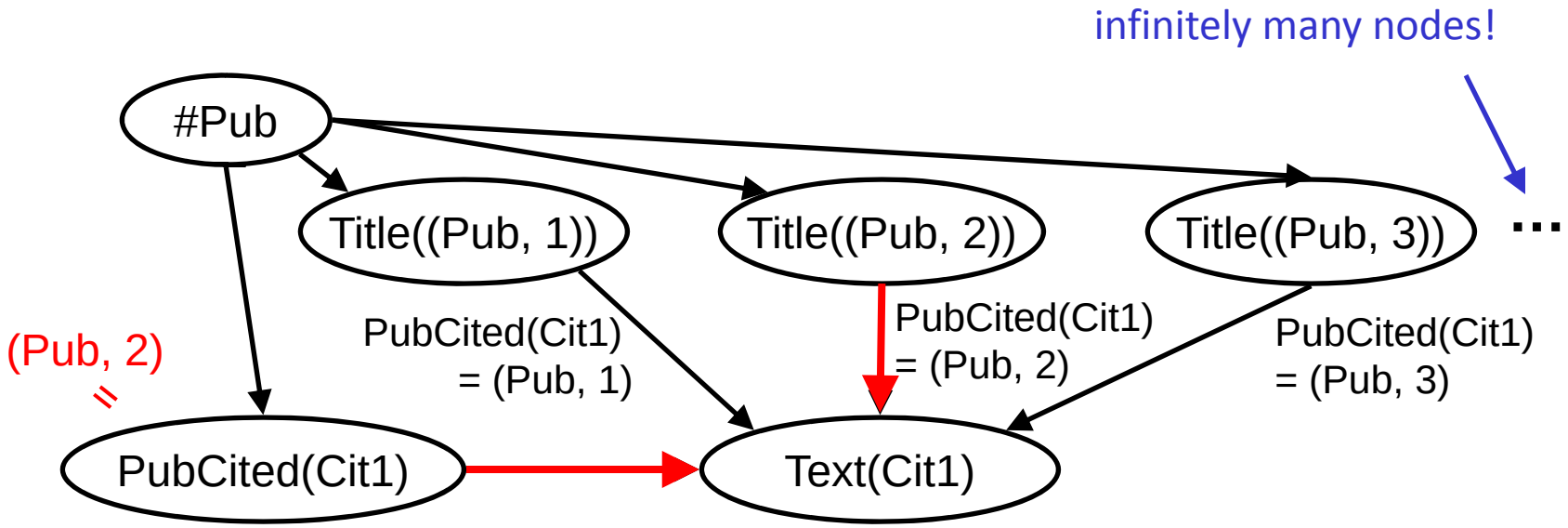


Basic Random Variables (RVs)

- 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

Contingent Bayesian Network

- Each BLOG model defines contingent Bayesian network (CBN) over basic RVs
  - Edges active only under certain conditions



[Milch *et al.*, AI/Stats 2005]

Probability Distribution

- Through its CBN, BLOG model specifies:
  - Conditional distributions for basic RVs
  - Context-specific independence properties *e.g.*, Text(Cit1) indep of Title((Pub, 1)) given PubCited(Cit1) = (Pub, 3)
- *Theorem*: Under certain "context-specific ordering" conditions, every BLOG model fully defines a distribution over possible worlds

[Milch *et al.*, IJCAI 2005]

Inference with Unknown Objects

- Does infinite set of basic RVs prevent inference?
- No: Sampling algorithms only need to instantiate finite set of relevant variables
- Generic algorithms:
  - Rejection sampling [Milch *et al.*, IJCAI 2005]
  - Guided likelihood weighting [Milch *et al.*, AI/Stats 2005]
- More practical: MCMC over partial worlds

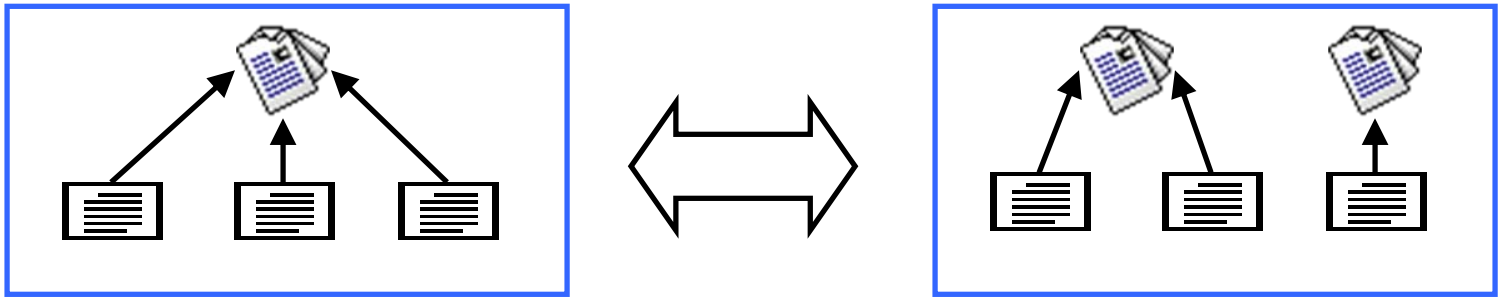
Toward General-Purpose MCMC with Unknown Objects

- Successful applications of MCMC with domain-specific proposal distributions:
  - Citation matching [Pasula et al., 2003]
  - Multi-target tracking [Oh et al., 2004]
- But each application requires new code for:
  - Proposing moves
  - Representing MCMC states
  - Computing acceptance probabilities
- Goal:
  - User specifies model and proposal distribution
  - General-purpose code does the rest

Proposer for Citations

[Pasula *et al*., NIPS 2002]

- Split-merge moves:



  - Propose titles and author names for affected publications based on citation strings
- Other moves change total number of publications

MCMC States

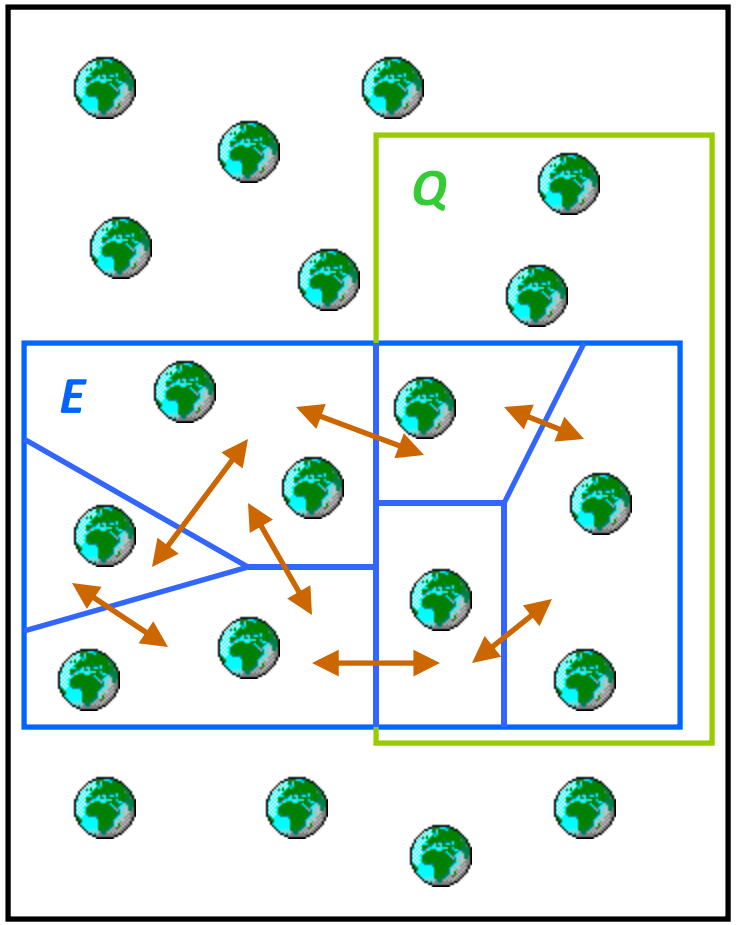
- Not complete instantiations!
  - No titles, author names for uncited publications
- States are partial instantiations of random variables

  #Pub = 100, PubCited(Cit1) = (Pub, 37), Title((Pub, 37)) = “Calculus”

  - Each state corresponds to an event: set of outcomes satisfying description

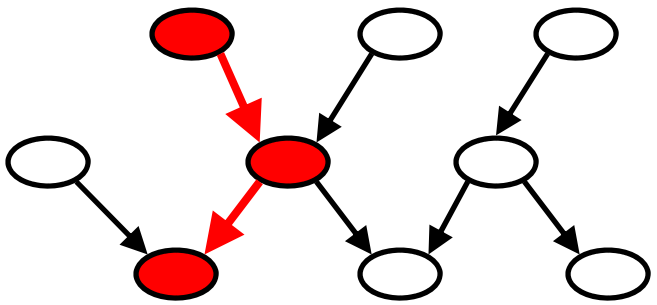
MCMC over Events

- Markov chain over events σ, with stationary distrib. proportional to $\boldsymbol{p}(\sigma)$
- *Theorem:* Fraction of visited events in $\boldsymbol{Q}$ converges to $\boldsymbol{p}(\boldsymbol{Q}|\boldsymbol{E})$ if:
  - Each σ is either subset of $\boldsymbol{Q}$ or disjoint from $\boldsymbol{Q}$
  - Events form partition of $\boldsymbol{E}$



Computing Probabilities of Events

- Engine needs to compute $P(\sigma')$ / $P(\sigma_n)$ efficiently (without summations)
- Use instantiations that include all active parents of the variables they instantiate



- Then probability is product of CPDs:

$$P(\sigma) = \prod_{X \in \text{vars}(\sigma)} p_X\left(\sigma(X) \mid \sigma(\text{Pa}_\sigma(X))\right)$$

States That Are Even More Abstract

- Typical partial instantiation:
  #Pub = 100, PubCited(Cit1) = (Pub, 37), Title((Pub, 37)) = “Calculus”,
  PubCited(Cit2) = (Pub, 14), Title((Pub, 14)) = “Psych”
  - Specifies *particular* publications, even though publications are interchangeable
- Let states be abstract partial instantiations:
  $\exists x \exists y \neq x$ [#Pub = 100, PubCited(Cit1) = ***x***, Title(***x***) = “Calculus”,
  PubCited(Cit2) = ***y***, Title(***y***) = “Psych”]
- There are conditions under which we can compute probabilities of such events

Outline

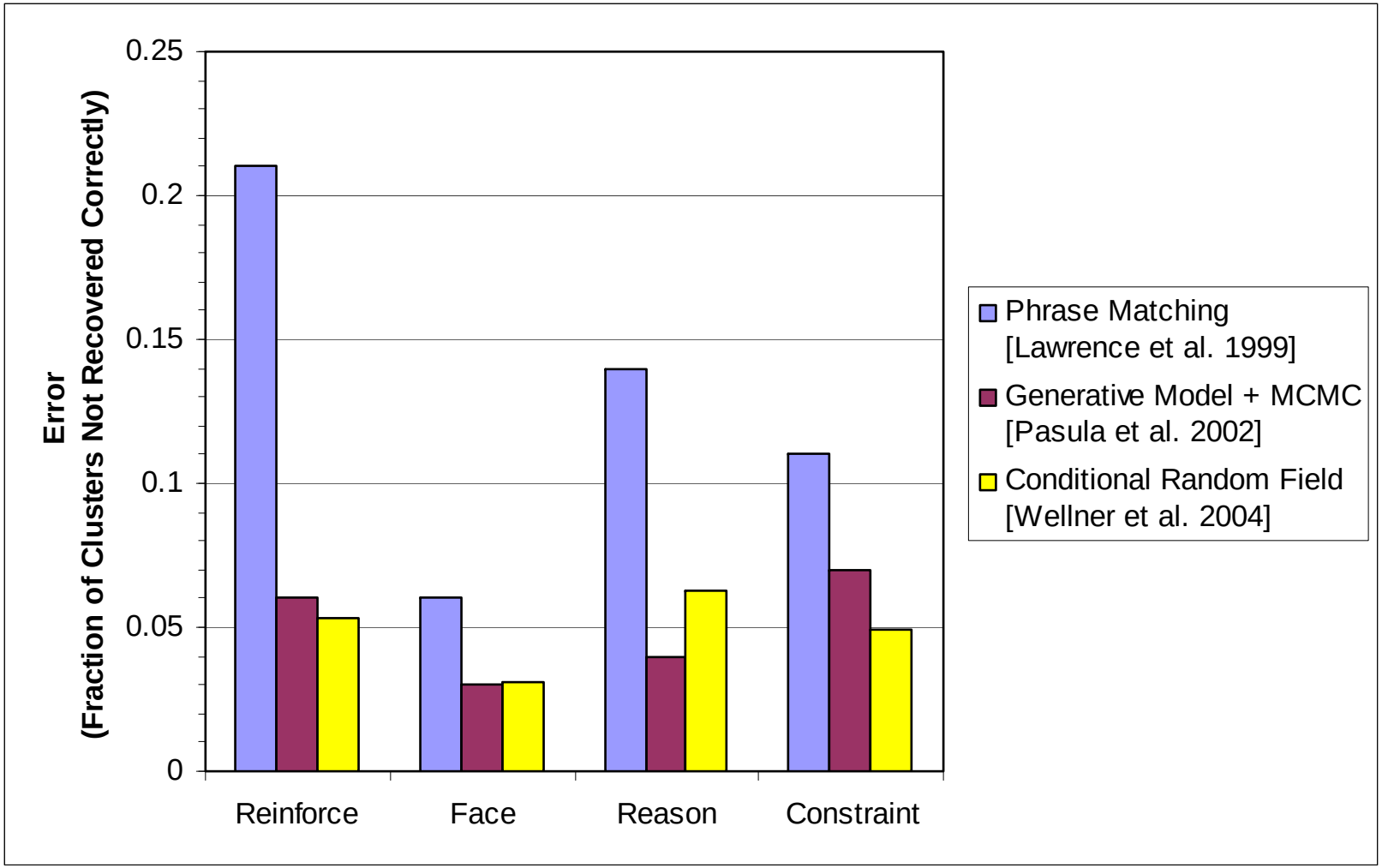
- Learning relational probability models
- Structural uncertainty
  - Uncertainty about relations
  - Uncertainty about object existence and identity
- Applications of BLOG

Citation Matching

- Elaboration of generative model shown earlier
- Parameter estimation
  - Priors for names, titles, citation formats learned offline from labeled data
  - String corruption parameters learned with Monte Carlo EM
- Inference
  - MCMC with split-merge proposals
  - Guided by “canopies” of similar citations
  - Accuracy stabilizes after ~20 minutes

[Pasula *et al*., NIPS 2002]

Citation Matching Results



Four data sets of ~300-500 citations, referring to ~150-300 papers

Cross-Citation Disambiguation

Wauchope, K. Eucalyptus: Integrating Natural Language Input with a Graphical User Interface. NRL Report NRL/FR/5510-94-9711 (1994).

Is "Eucalyptus" part of the title, or is the author named K. Eucalyptus Wauchope?

Kenneth Wauchope (1994). Eucalyptus: Integrating natural language input with a graphical user interface. NRL Report NRL/FR/5510-94-9711, Naval Research Laboratory, Washington, DC, 39pp.

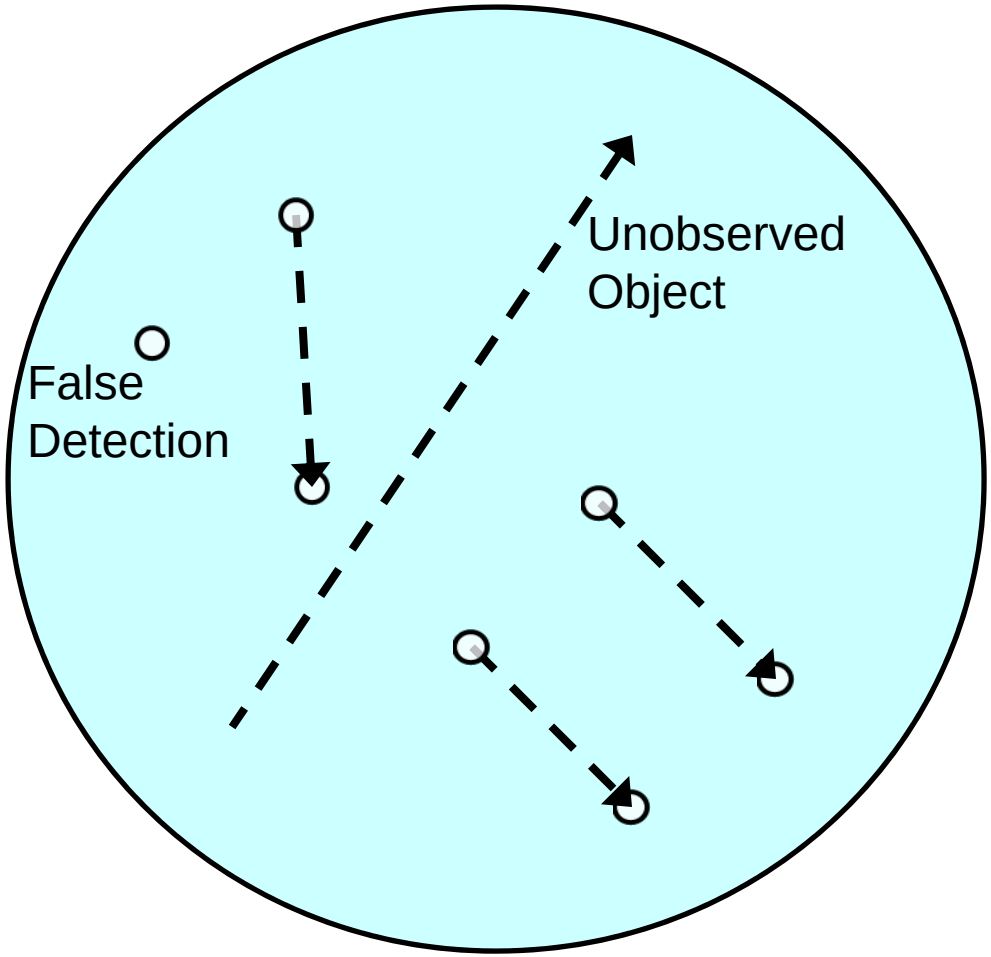
Second citation makes it clear how to parse the first one

Preliminary Experiments: Information Extraction

- P(citation text | title, author names) modeled with simple HMM
- For each paper: recover title, author surnames and given names
- Fraction whose attributes are recovered perfectly in last MCMC state:
  - among papers with one citation: 36.1%
  - among papers with multiple citations: 62.6%

Can use inferred knowledge for disambiguation

Multi-Object Tracking



State Estimation for “Aircraft”

```
#Aircraft ~ NumAircraftPrior();

State(a, t)
   if t = 0 then ~ InitState()
   else ~ StateTransition(State(a, Pred(t)));

#Blip(Source = a, Time = t)
   ~ NumDetectionsCPD(State(a, t));

#Blip(Time = t)
   ~ NumFalseAlarmsPrior();

ApparentPos(r)
   if (Source(r) = null) then ~ FalseAlarmDistrib()
   else ~ ObsCPD(State(Source(r), Time(r)));
```

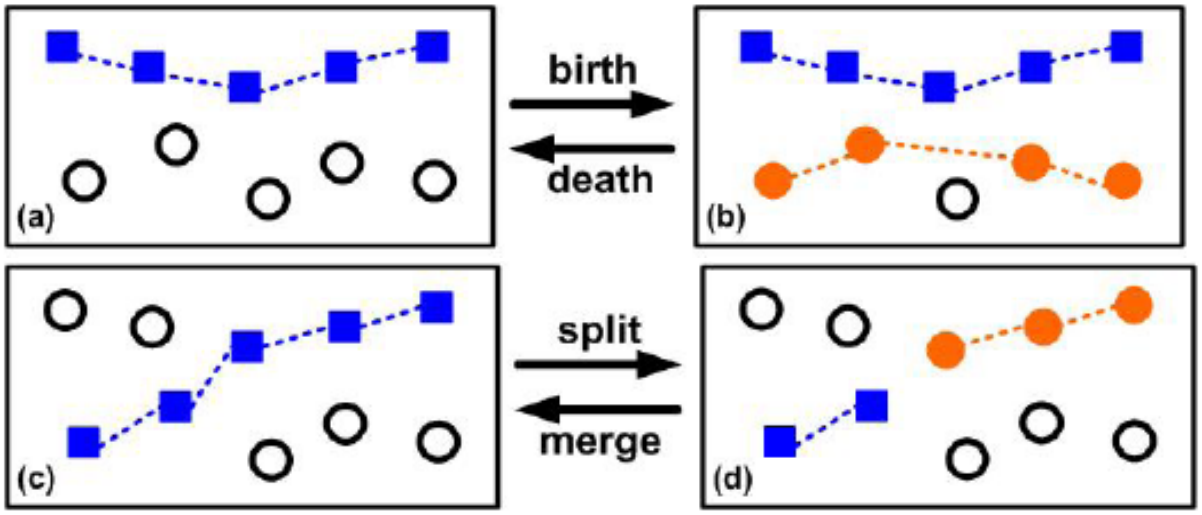
Aircraft Entering and Exiting

```
#Aircraft(EntryTime = t) ~ NumAircraftPrior();

Exits(a, t)
   if InFlight(a, t) then ~ Bernoulli(0.1);

InFlight(a, t)
   if t < EntryTime(a) then = false
   elseif t = EntryTime(a) then = true
   else = (InFlight(a, Pred(t)) & !Exits(a, Pred(t)));

State(a, t)
   if t = EntryTime(a) then ~ InitState()
   elseif InFlight(a, t) then
       ~ StateTransition(State(a, Pred(t)));

#Blip(Source = a, Time = t)
   if InFlight(a, t) then
       ~ NumDetectionsCPD(State(a, t));
```

…plus last two statements from previous slide

MCMC for Aircraft Tracking

- Uses generative model from previous slide (although not with BLOG syntax)
- Examples of Metropolis-Hastings proposals:

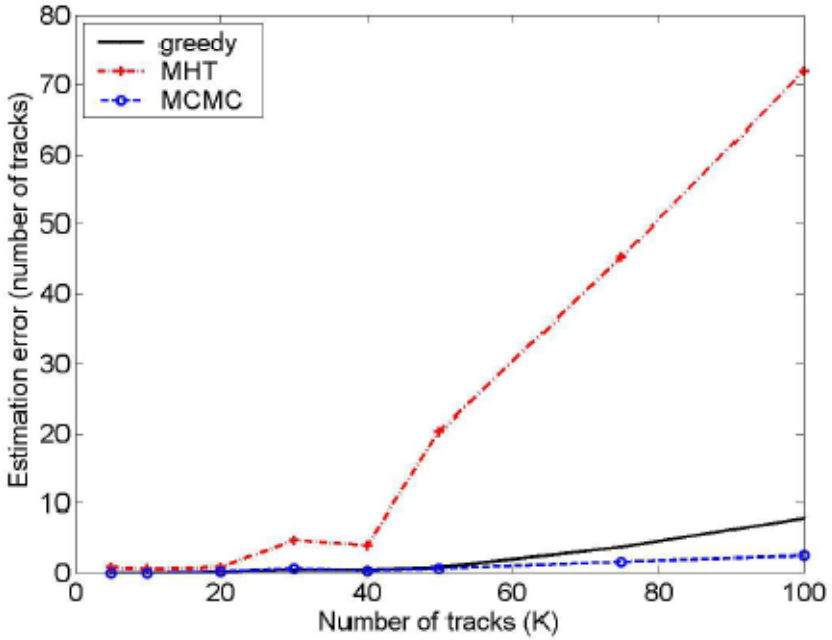


[Figures by Songhwai Oh]

[Oh et al., CDC 2004]

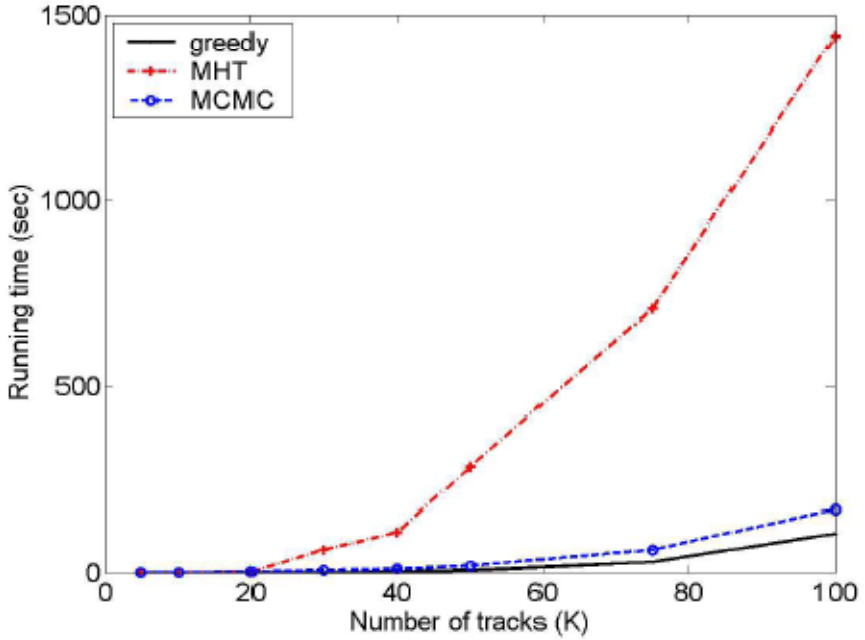
Aircraft Tracking Results

Estimation Error



MCMC has smallest error, hardly degrades at all as tracks get dense

Running Time



MCMC is nearly as fast as greedy algorithm; much faster than MHT

[Figures by Songhwai Oh]

[Oh et al., CDC 2004]

BLOG Software

- Bayesian Logic inference engine available:

  http://people.csail.mit.edu/milch/blog

References

- Friedman, N., Getoor, L., Koller, D.,and Pfeffer, A. (1999) "Learning probabilistic relational models". In *Proc. 16th Int'l Joint Conf. on AI*, pages 1300-1307.
- Taskar, B., Segal, E., and Koller, D. (2001) "Probabilistic classification and clustering in relational data". In *Proc. 17th Int'l Joint Conf. on AI*, pages 870-878.
- Getoor, L., Friedman, N., Koller, D., and Taskar, B. (2002) "Learning probabilistic models of link structure". *J. Machine Learning Res.* 3:679-707.
- Taskar, B., Abbeel, P., and Koller, D. (2002) "Discriminative probabilistic models for relational data". In *Proc. 18th Conf. on Uncertainty in AI*, pages 485-492.
- Dzeroski, S. and Lavrac, N., eds. (2001) *Relational Data Mining*. Springer.
- Flach, P. and Lavrac, N. (2002) "Learning in Clausal Logic: A Perspective on Inductive Logic Programming". In *Computational Logic: Logic Programming and Beyond (Essays in Honour of Robert A. Kowalski),* Springer Lecture Notes in AI volume 2407, pages 437-471.
- Pasula, H. and Russell, S. (2001) "Approximate inference for first-order probabilistic languages". In *Proc. 17th Int'l Joint Conf. on AI*, pages 741-748.
- Milch, B., Marthi, B., Russell, S., Sontag, D., Ong, D. L., and Kolobov, A. (2005) "BLOG: Probabilistic Models with Unknown Objects". In *Proc. 19th Int'l Joint Conf. on AI*, pages 1352-1359.

References

- Milch, B., Marthi, B., Russell, S., Sontag, D., Ong, D. L., and Kolobov, A. (2005) "BLOG: Probabilistic Models with Unknown Objects". In *Proc. 19th Int'l Joint Conf. on AI*, pages 1352-1359.
- Milch, B., Marthi, B., Sontag, D., Russell, S., Ong, D. L., and Kolobov, A. (2005) "Approximate inference for infinite contingent Bayesian networks". In *Proc. 10th Int'l Workshop on AI and Statistics*.
- Milch, B. and Russell, S. (2006) "General-purpose MCMC inference over relational structures". In *Proc. 22nd Conf. on Uncertainty in AI*, pages 349-358.
- Pasula, H., Marthi, B., Milch, B., Russell, S., and Shpitser, I. (2003) "Identity uncertainty and citation matching". In *Advances in Neural Information Processing Systems 15*, MIT Press, pages 1401-1408.
- Lawrence, S., Giles, C. L., and Bollacker, K. D. (1999) "Autonomous citation matching". In *Proc. 3rd Int'l Conf. on Autonomous Agents*, pages 392-393.
- Wellner, B., McCallum, A., Feng, P., and Hay, M. (2004) "An integrated, conditional model of information extraction and coreference with application to citation matching". In *Proc. 20th Conf. on Uncertainty in AI*, pages 593-601.

References

- Pasula, H., Russell, S. J., Ostland, M., and Ritov, Y. (1999) “Tracking many objects with many sensors”. In *Proc. 16th Int'l Joint Conf. on AI*, pages 1160-1171.
- Oh, S., Russell, S. and Sastry, S. (2004) “Markov chain Monte Carlo data association for general multi-target tracking problems”. In *Proc. 43rd IEEE Conf. on Decision and Control*, pages 734-742.