First-Order Probabilistic Languages: Into the Unknown

Stuart Russell and Brian Milch
UC Berkeley

Outline

- Background and Motivation
  - Why we need more expressive formal languages for probability
  - Why unknown worlds matter
- Technical development
  - Relational models with known skeleton
  - Relational models with unknown relations
  - Unknown objects and identity uncertainty
- Applications
  - Citation matching
  - State estimation
- Open problems, future work
  - Why we need syntax and semantics

Assumed background

- Roughly, the intersection of backgrounds of modern AI, machine learning, learning theory, statistics
  - Basics of probability theory
  - Graphical models and algorithms (incl. MCMC)
  - Some acquaintance with basic concepts of logic (quantifiers, logical variables, relations, functions, equality)
- Intersection of motivations: { }
- Our motivation: programs that understand the real world

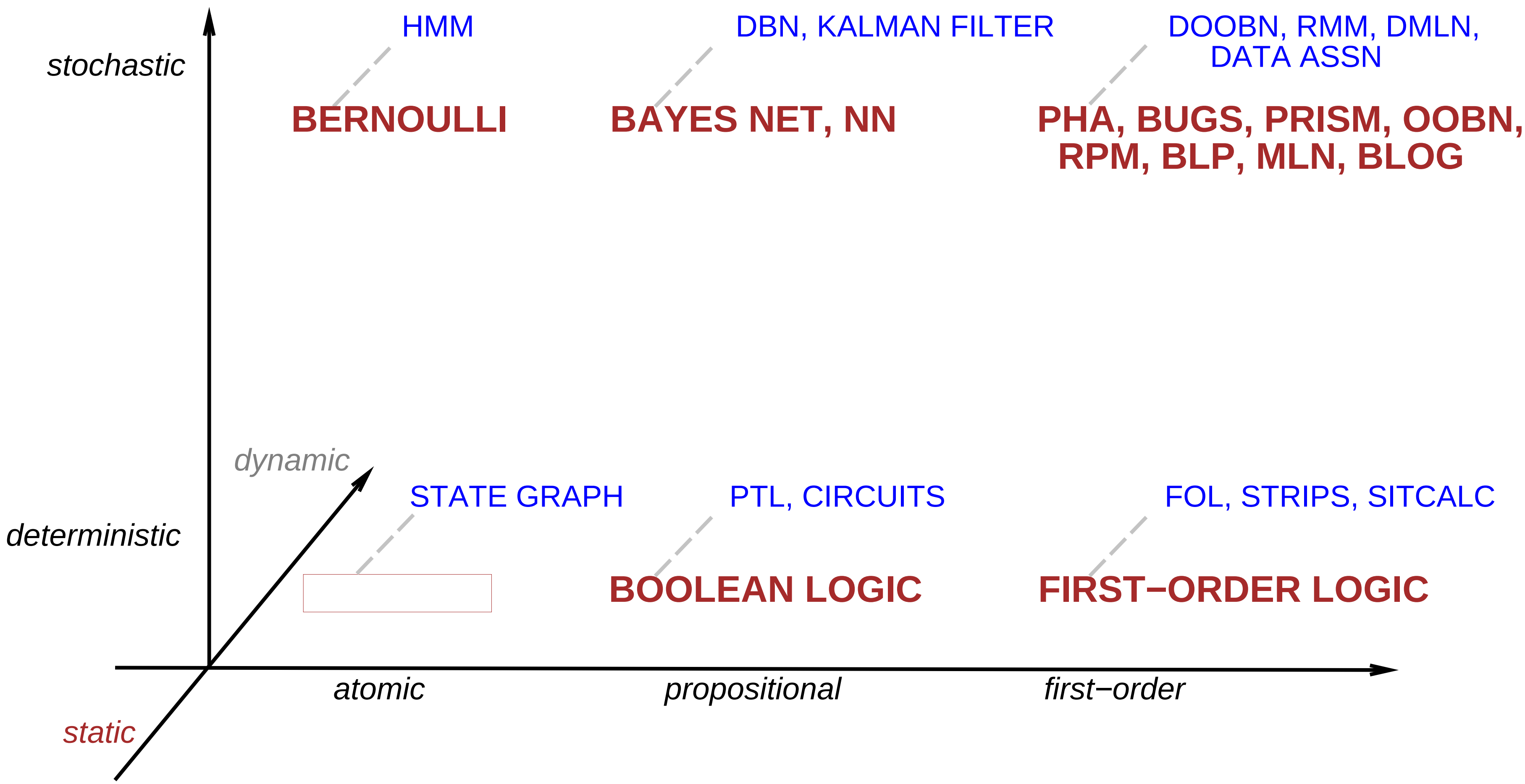
What to take away

- Understanding of purpose and mechanics (syntax, semantics) of expressive formal languages for probabilistic modelling
- Understanding of commonly identified levels of expressiveness beyond standard graphical models, including “unknown worlds”
- Ability to classify a proposed application according to the level of expressiveness required and to identify the relevant tools
- Familiarity with at least one expressive formal language (BLOG) that handles unknown worlds

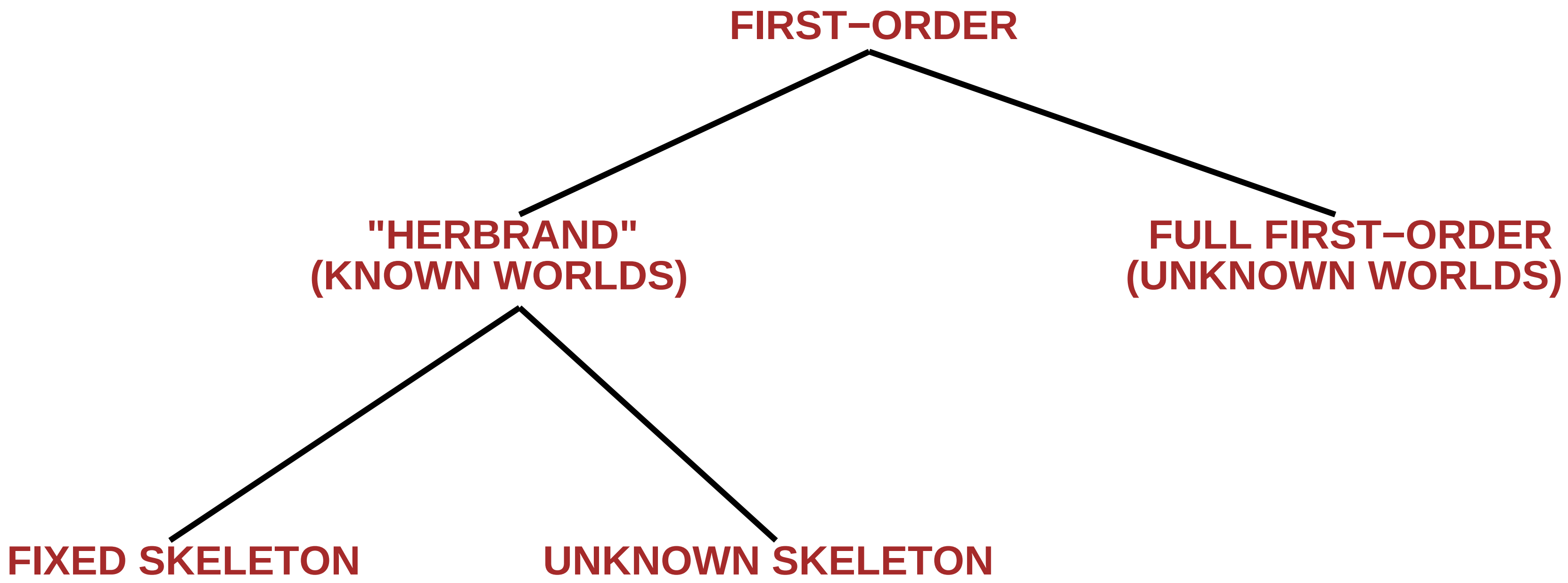
Expressiveness

- Expressive language => concise models => fast learning, sometimes fast inference
  - E.g., rules of chess: 1 page in first-order logic, 100,000 pages in propositional logic
  - E.g., DBN vs HMM inference
- Language ***A*** is as expressive as language ***B*** iff for every sentence ***b*** in ***B*** there is an equivalent sentence ***a*** in ***A*** such that $|\boldsymbol{a}| = O(1)|\boldsymbol{b}|$
- Recent trend towards expressive formal languages in statistics and machine learning
  - E.g., graphical models, plates, relational models

A crude classification



Refining the classification



Herbrand vs full first-order

Given

Father(Bill,William) and Father(Bill,Junior)

How many children does Bill have?

Herbrand (also relational DB) semantics:

2

First-order logical semantics:

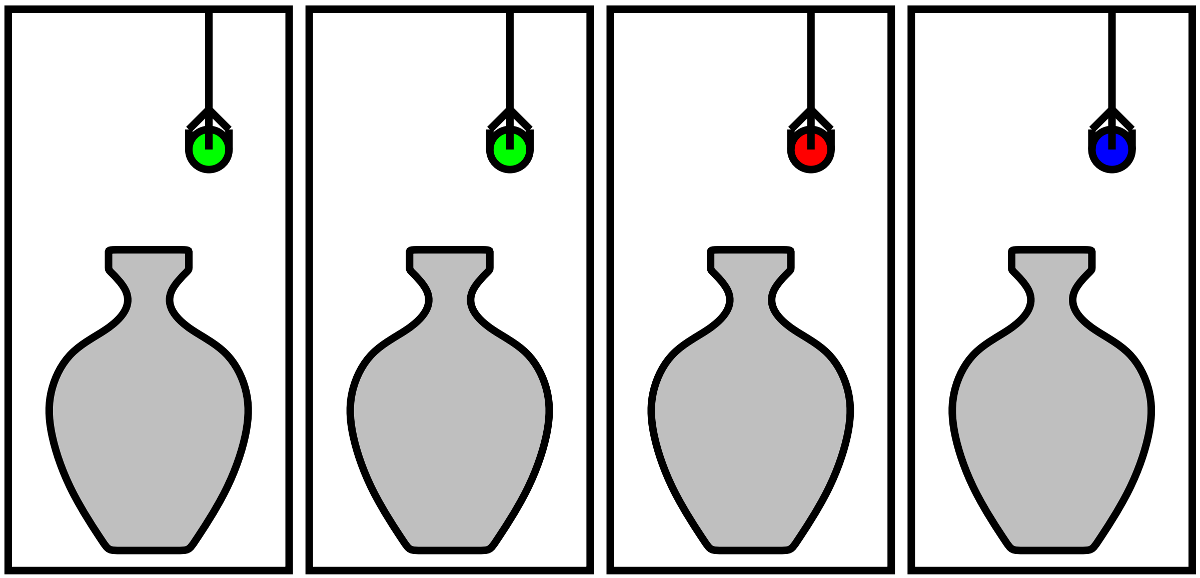
Between 1 and ∞

Unknown worlds

- Herbrand (and DB, Prolog) semantics assumes unique names and domain closure, so all possible worlds have the same, known, named objects
- First-order logic allows
  - different constants to refer to the same objects
  - objects that are not referred to by any constant

I.e. unknown worlds

Example: balls and urns

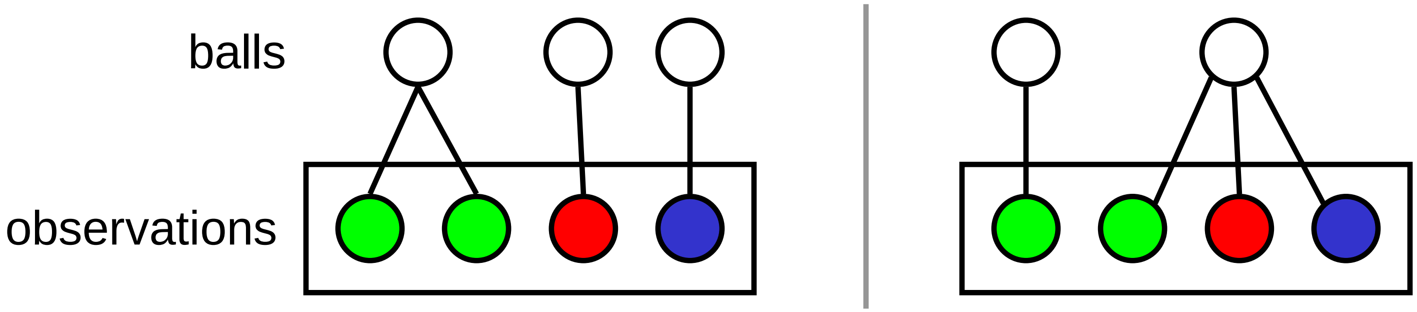


Sample balls w/ replacement, measure color

How many balls are in the urn?

Balls and urns contd.

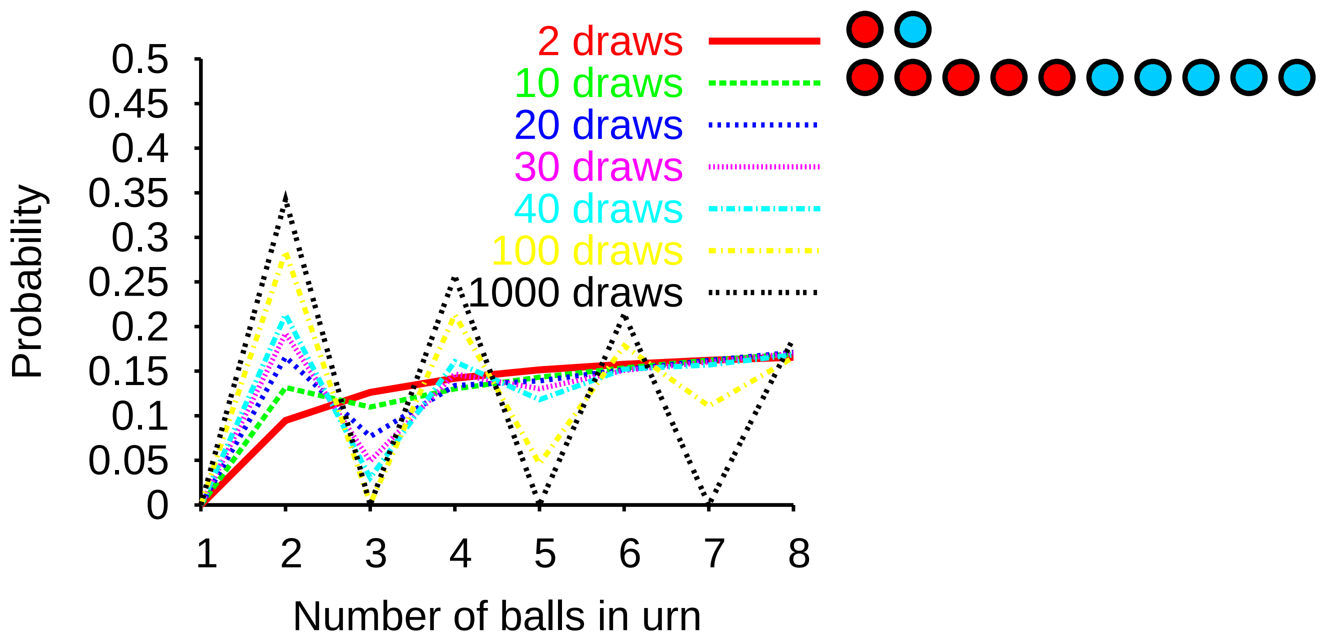
- N balls, prior distribution P(N)
- True colours $C_1,\ldots C_N$, identical priors $P(C_i)$
- k observations, observed colours $O=O_1,..,O_k$
- Assignment ω specifies which ball was observed in each observation



- Sensor model $P(O_j \mid C_{\omega(j)})$

Balls and urns contd.

- No identical balls
  - converge to true N as k → ∞
- Identical balls possible
  - all multiples of minimal N possible as k → ∞



Example: Citation Matching

[Lashkari et al 94] Collaborative Interface Agents, Yezdi Lashkari, Max Metral, and Pattie Maes, Proceedings of the Twelfth National Conference on Articial Intelligence, MIT Press, Cambridge, MA, 1994.

Metral M. Lashkari, Y. and P. Maes. Collaborative interface agents. In Conference of the American Association for Artificial Intelligence, Seattle, WA, August 1994.

Are these descriptions of the same object?

This problem is ubiquitous with real data sources, hence the record linkage industry

CiteSeer02: Russell w/4 Norvig

CiteSeer02: Russell w/4 Norvig

- Russell S, Norvig P (1995) Artificial Intelligence: A Modern Approach, Prentice Hall Series in Artificial Intelligence. Englewood Cliffs, New Jersey
- Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 1995.
- Russell S.; Norvig, P. Articial Intelligence - A Modern Approach. Prentice-Hall International Editions, 1995.
- Russell S.J., Norvig P., (1995) Artificial Intelligence, A Modern Approach. Prentice Hall.
- S. Russell and P. Norvig. Articial Intelligence, a Modern Approach. Prentice Hall, New Jersey, NJ, 1995.

- Stuart Russell and Peter Norvig. Artificial intelligence: A modern approach. Prentice-Hall Series on Artificial Intelligence. Prentice-Hall, Englewood Cliffs, New Jersey, 1995.
- S. Russell and P Norvig. Artifical Intelligence: a Modern Approach. Prentice Hall, 1995. Book Details from Amazon or Barnes \& Noble
- Stuart Russell and Peter Norvig. Articial Intelligence: A Modern Approach. Prentice Hall, 1995.
- S. J. Russell and P. Norvig. Artificial Intelligence, a modern approach. Prentice Hall, Upper Saddle River, New Jersey 07458, 1995.
- Stuart Russell and Peter Norvig. Artificial Intelligence. A modern approach. Prentice-Hall, 1995.
- S. J. Russell and P. Norvig. Articial Intelligence: A Modern Approach. Prentice Hall. 1995.
- S. Russell and P. Norvig, Artificial Intelligence A Modern Approach Prentice Hall 1995.
- S. Russell and P. Norvig. Introduction to Artificial Intelligence. Prentice Hall, 1995.

- Stuart Russell and Peter Norvig. Artficial Intelligence: A Modern Approach. Prentice-Hall, Saddle River, NJ, 1995.
- Stuart Russell and Peter Norvig. Articial Intelligence a modern approach. Prentice Hall series in articial intelligence. Prentice Hall, Upper Saddle River, New Jersey, 1995.
- Chapter 18 Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Norvig, Prentice-Hall, 2000.
- Dynamics of computational ecosystems. Physical Review A 40:404--421. Russell, S., and Norvig, P. 1995. Artificial Intelligence: A Modern Approach. Prentice Hall.
- S. Russell, P. Norvig: Artificial Intelligence -- A Modern Approach, Prentice Hall, 1995.
- Russell, S. \& Norvig, P. (1995) Artificial Intelligence: A Modern Appraoch (Englewood Cliffs, NJ: Prentice-Hall). Book Details from Amazon or Barnes \& Noble
- Stuart Russell and Peter Norvig. AI: A Modern Approach. Prentice Hall, NJ, 1995.
- S. Russell, P. Norvig. Artificial Intelligence: A Modem Approach. Prentice- Hall, Inc., 1995.

- 391-414. Russell SJ, Norvig P (
- Russell and Peter Norvig, "Artificial Intelligence - A Modern Approach (AIMA)", pp. 33
- Stuart Russell and Peter Norvig: Artificial Intelligence: A Modern Approach, Prentice-Hall, 1994.
- Russell, S. \& Norvig, P., An Introduction to Artificial Intelligence: A Modern Approach, Prentice Hall International, 1996.
- S. Russell, P. Norvig. Artician Intelligence. A modern approach. Prentice Hall, 1995.
- Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach. Prentice Hall, 1995. Contributing writers: John F. Canny, Jitendra M. Malik, Douglas D. Edwards. ISBN 0-13-103805-2.
- Stuart Russell and Peter Norvig. Artificial Intelligence: A Mordern Approach. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1995.

- In Proceedings of the Third Annual Conference on Evolutionary Programming (pp. 131--139). River Edge, NJ: World Scientific. Russell, S.J., \& Norvig, P. 1995. Artificial Intelligence, A Modern Approach. Englewood Cliffs, NJ: Prentice Hall.

- John Wiley. Russell, S., \& Norvig, P. (1995). Artificial Intelligence: A Modern Approach. Prentice-Hall, Inc.

- Stuart Russell and Peter Norvig: Artifcial Intelligence A Modern Approach, Englewood Clioes, NJ: Prentice Hall, 1995.

- In Scherer, K.R. \& Ekman, P. Approaches to Emotion, 13--38. Hillsdale, NJ: Lawrence Erlbaum. Russell, S.J. and Norvig, P. 1995. Artificial Intelligent: A Modern Approach. Englewood Cliffs, NJ: Prentice Hall.

- Rosales E, Forthcoming Masters dissertation, Department of Computer Science, University of Essex, Colchester UK Russell S and Norvig P (1995) Artificial Intelligence: A Modern Approach. Prentice Hall: Englewood Cliffs, New Jersey.

- S. Russell and P. Norvig (1995) Artificial Intelligence; A Modern Approach, Prentice Hall, New Jersey.

- S. Russell and P. Norvig. Articial Intelligence. A Modern Approach. Prentice-Hall, 1995. ISBN 0-13-360124-2.

- Stuart J. Russell and Peter Norvig. Articial Intelligence: A Modern Approach, chapter 17. Number 0-13-103805-2 in Series in Articial Intelligence. Prentice Hall, 1995.
- Stuart J. Russell and Peter Norvig. Articial Intelligence A Modern Approach. Prentice Hall, Englewood Cli s, New Jersey, USA, 1995. 32
- Morgan Kaufmann Publishers. Russell, S., and Norvig, P. 1995. Artificial Intelligence: A Modern Approach. Prentice Hall.
- Stuart J. Russell and Peter Norvig. Articial Intelligence: AModern Approach,chapter 17. Number 0-13-103805-2 in Series in Articial Intelligence. Prentice Hall, 1995.
- W. Shavlik and T. G. Dietterich, eds., Morgan Kaufmann, San Mateo, CA. Russell, S. and Norvig, P. (1995). Artificial Intelligence - A Morden Approach. Englewood Cliffs, NJ: Prentice-Hall.
- KeyGraph: Automatic indexing by co-occurrence graph based on building construction metaphor. In Advanced Digital Library Conference. to appear. Russell, S., and Norvig, P. 1995. Artificial Intelligence --A Modern Approach--.
- Prentice-Hall.
- Formal derivation of rule-based programs. IEEE Transactions on Software Engineering 19(3):277--296. Russell, S., and Norvig, P. 1995. Artificial Intelligence: A Modern Approach. Prentice Hall.

- Russell, Stuart and Peter Norvig, Artificial Intelligence, A Modern Approach, New Jersey, Prentice Hall, 1995.
- S. Russell, P. Norvig: Articial Intelligence: A modern approach; Prentice Hall (1995).
- Rechenberg, I. (89). Artificial evolution and artificial intelligence. In Forsyth, R. (Ed.), Machine Learning, pp. 83--103 London. Chapman. Russell, S., \& Norvig, P. (1995). Artificial Intelligence: A Modern Approach. Prentice Hall.
- Russell, S and Norvig, P. 1995. Articial Intelligence: A Modern Approach Prentice-Hall, Englewood Cli s, New Jersey, 1995.
- Russell, S., \& Norvig, P. (1995) . Artificial intelligence: A modern monitoring methods for information retrieval systems: From search approach. Prentice-Hall series on artificial intelligence. Upper Saddle product to search process. Journal
- of the American Society for Information Science, 47, 568 -- 583. River, NJ: Prentice-Hall.
- Stuart J. Russell and Peter Norvig. Artificial Intelligence: A Modern Approach, chapter 17. Number 0-13-103805-2 in Series in Artificial Intelligence. Prentice Hall, 1995.
- S. Russell and P. Norvig. Articial Intelligence A Modern Approach. Prentice Hall, Englewood Cli s, 1995.

- Russell, Stuart and Norvig, Peter: Artificial Intelligence: A Modern Approach, Prentice Hall, Englewood Cliffs NJ, 1995

- S. Russell and P. Norvig. ????????? ????????????? ? ?????? ????????. Prentice Hall, Englewood Cli s, NJ, 1995.

- S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach - The Intelligent Agent Book, Prentice Hall, NY, 1995.

- S. Russell and P. Norvig. Artificial Intelligence-aModern Approach. Prentice Hall International, Englewood Cliffs, NJ,USA,1995.

- S.J.Russell, P.Norvig: Arti cial intelligence. A modern approach", Prentice-Hall International, 1995.

- In Proceedings of the Third Annual Conference on Evolutionary Programming (pp. 131--139). River Edge, NJ: World Scientific. Russell, S.J., \& Norvig, P. 1995. Artificial Intelligence, A Modern Approach. Englewood Cliffs, NJ: Prentice
- Hall.

- In Working Notes of the IJCAI-95 Workshop on Entertainment and AI/ALife, 19--24. Russell, S., and Norvig, P. 1995. Artificial Intelligence: A Modern Approach. Prentice Hall.

- Stuart J. Russell and Peter Norvig. Artiilcial Intelligence: A Modern Approach. Prentice Hall, Englewood Cliffs, N J, 1995.

- Academic Press. 359--380. Russell, S., and Norvig, P. 1994. Artificial Intelligence: A Modern Approach. Prentice Hall.

- Stuart J. Russell, Peter Norvig, Artifical Intelligence: A Modern Appraoch, Prentice-Hall, Englewood Cliffs, New Jersey. 1994.

- Cambridge, MA: MIT Press. Russell, S. J., and Norvig, P. (1994). Artificial Intelligence: A Modern Approach. Englewood Cliffs, NJ: Prentice-Hall.

- Morgan Kauffman. Russell, S., and Norvig, P. 1994. Artificial Intelligence: A Modern Approach. Prentice Hall.

- Fast Plan Generation Through Heuristic Search Russell, S., \& Norvig, P. (1995). Artificial Intelligence: A Modern Approach. Prentice-Hall, Englewood Cliffs, NJ.

- Hoffmann \& Nebel Russell, S., \& Norvig, P. (1995). Artificial Intelligence: A Modern Approach. Prentice-Hall, Englewood Cliffs, NJ.

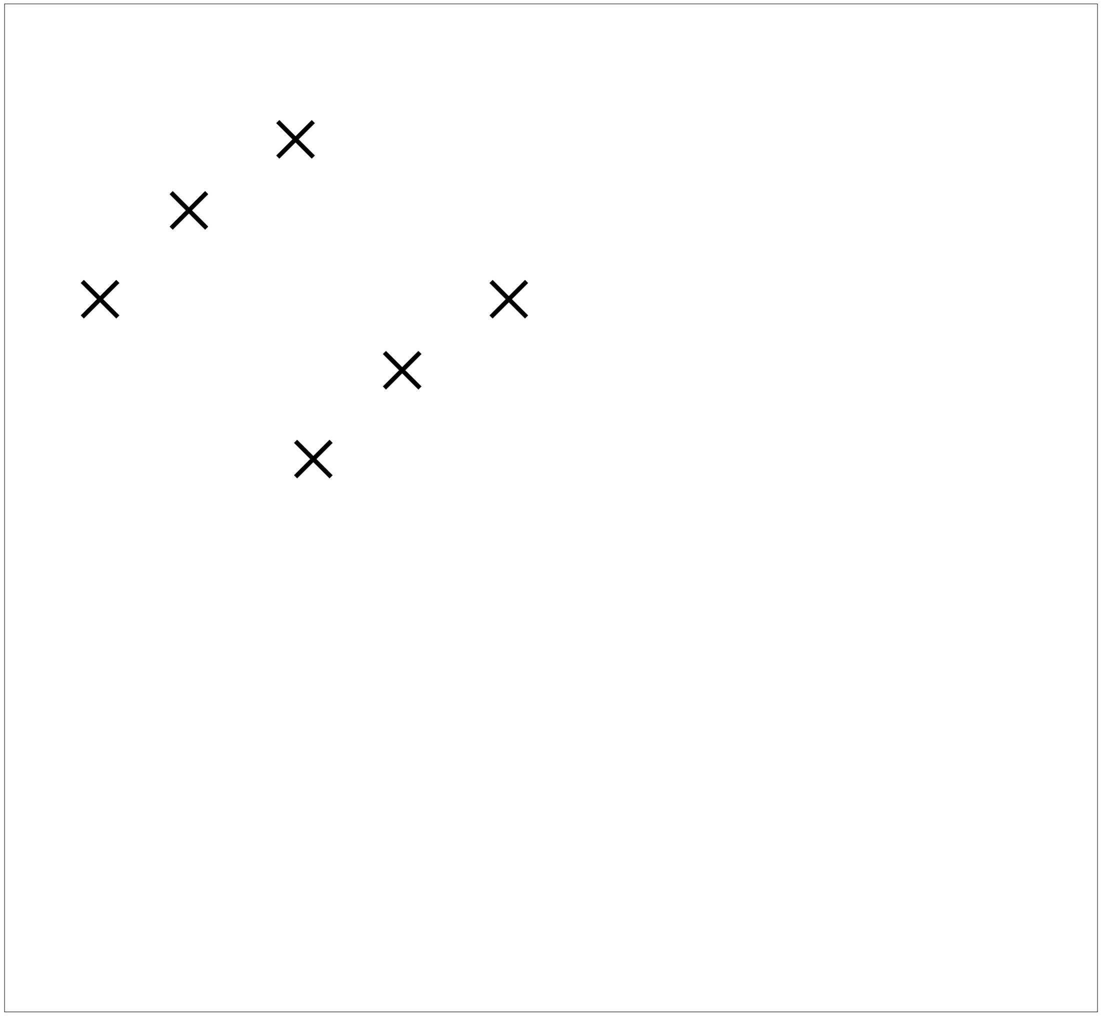
- Stuart Russel and Peter Norvig. Artificial Intelligence: A Modern Approach, chapter 12.1 - 12.3, pages 367--380. Prentice Hall, 1995.
- Stuart Russel and Peter Norvig. Artificial Intelligence, A Modern Approach. PrenticeHall, 1996. 2
- Stuart Russel, Peter Norvig, Articial Intelligence: A Modern Approach, Prentice Hall, New Jersey, US, 1995
- Russel, S., and Norvig, P. Articial Intelligence. A Modern Approach. Prentice Hall Series in Artificial Intelligence. 1995.
- S. Russel and P. Norvig. Artificial Intelligence, A Modern Approach, Prentice Hall: 1995.   Book Details from Amazon or Barnes \& Noble
- S. J. Russel and P. Norvig. Articial Intelligence A Modern Approach, chapter 14, pages 426-435. Prentice Hall Series in Articial Intelligence. Prentice Hall International, Inc., London, UK, rst edition, 1995. Exercise 14.3.
- Russel, S. and P. Norvig. Articial intelligence: A modern approach, Prentice Hall, 1995.   Book Details from Amazon or Barnes \& Noble

- S. Russel and P. Norvig Artificial Intelligence: A Modern Approach, MIT Press 1995.
- Russel, S. and Norvig, P., "Artificial Intelligence: A Modern Approch," p. 111-114, Prentice-Hall.
- J. Russel and P. Norvig. Artificial Intelligence, A Modern Approach. Prentice Hall, Upper Saddle River, NJ, 1995. 71
- Stuart Russel and Peter Norvig. A Modern, Agent-Oriented Approach to Introductory Artificial Intelligence. 1995.
- Stuart J. Russel and Peter Norvig. Artificial Intelligence---A Modern Approach, chapter 14, pages 426--435. Prentice Hall Series in Artificial Intelligence. Prentice Hall Internationall, Inc., London, UK, first edition, 1995. Excersice 14.3.
- Russel S. and Norvig P. (1995). Articial Intelligence. A Modern Approach. Prentice Hall Series in Artificial Intelligence.
- S. Russel, P. Norvig Articial Intelligence - A Modern Approach Prentice Hall, 1995
- Russel, S., P. Norvig. Artificial Intelligence: A Modern Approach Prentice Hall 1995.

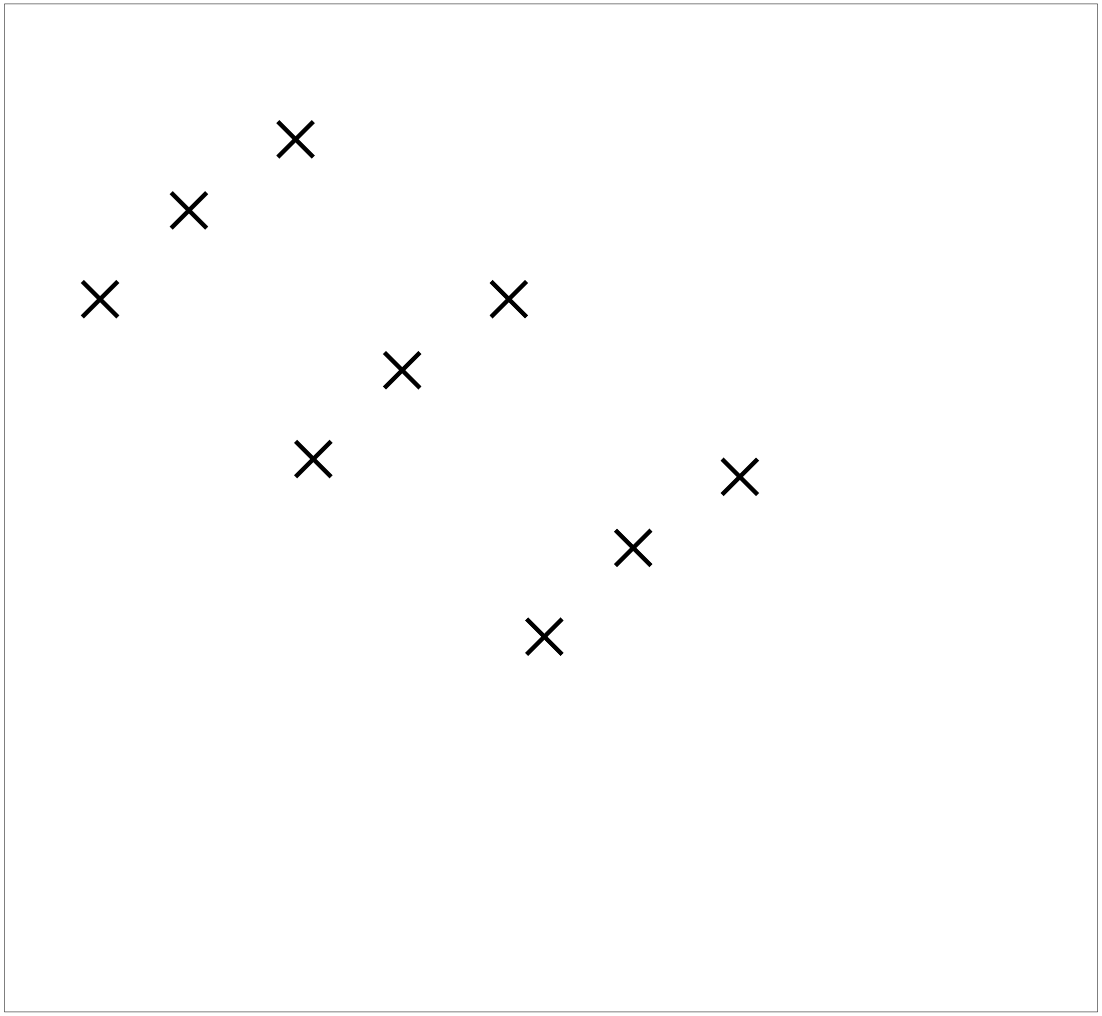
- Artificial Intelligence, S Russel \& P Norvig, Prentice Hall, 1995 21
- Russel, S.J, Norvig P: Artificial Intelligence. A Modern Approach, Prentice Hall Inc. 1995
- Russel, S., Norvig, P. (1995) Artificial Intellience - A modern approach. (Englewood Cliffs: Prentice Hall International).

Example: classical data association

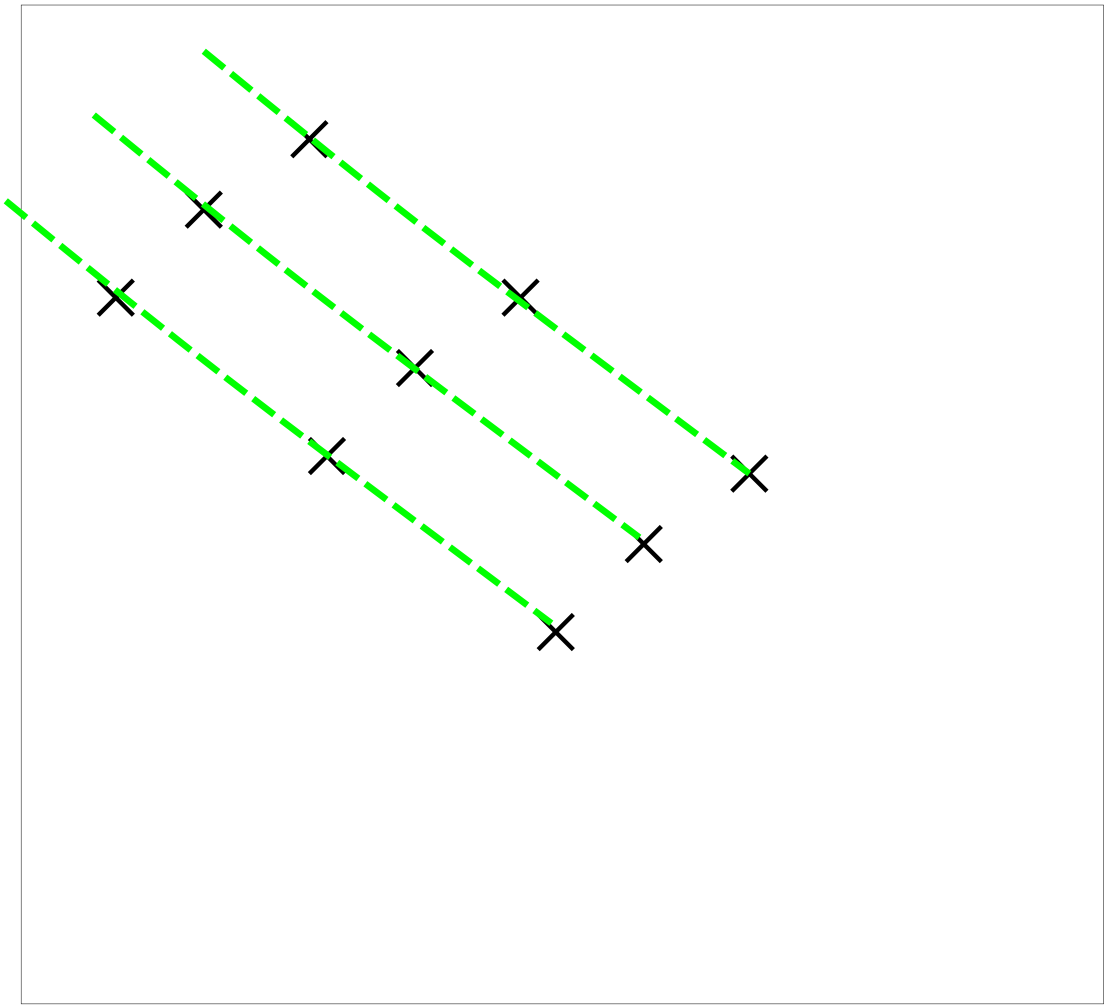
Example: classical data association



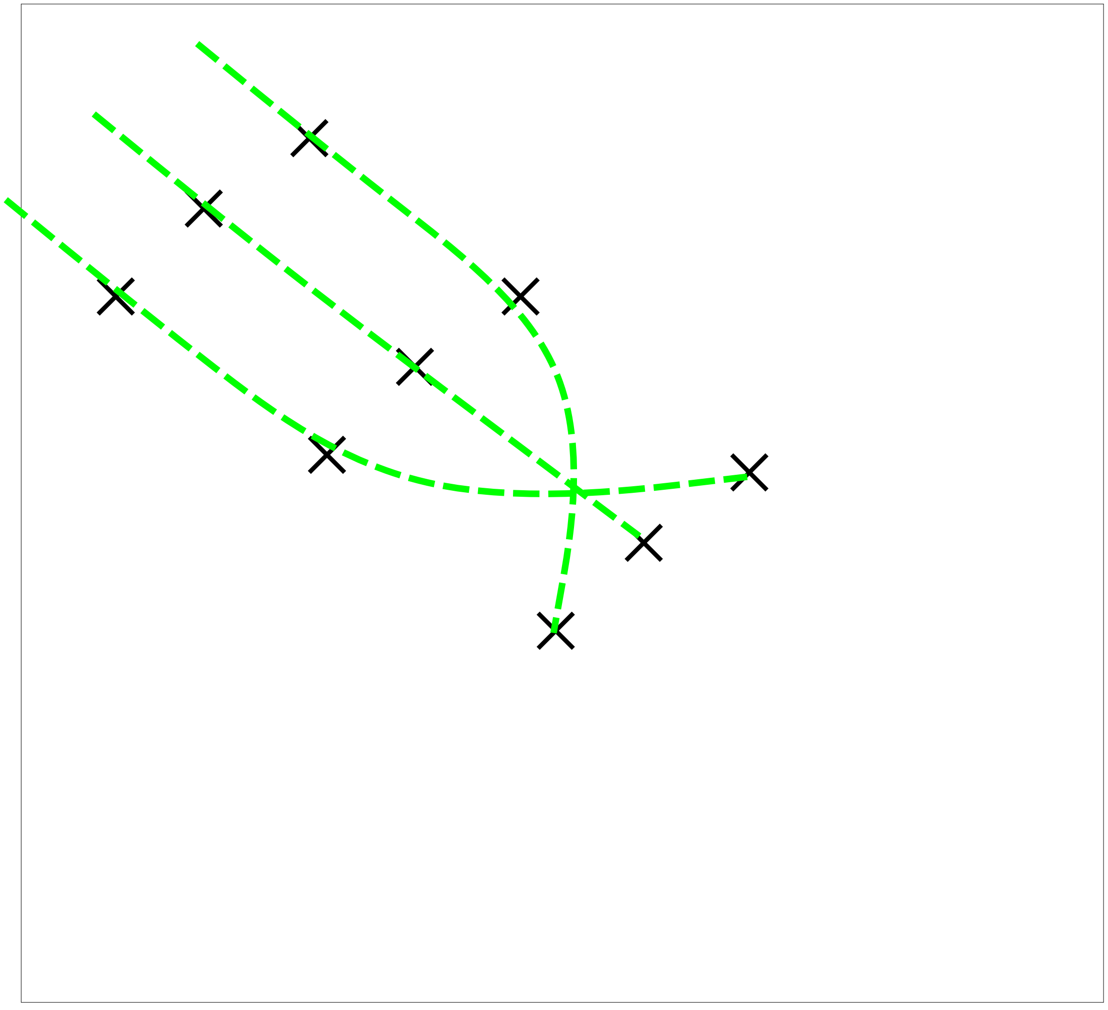
Example: classical data association



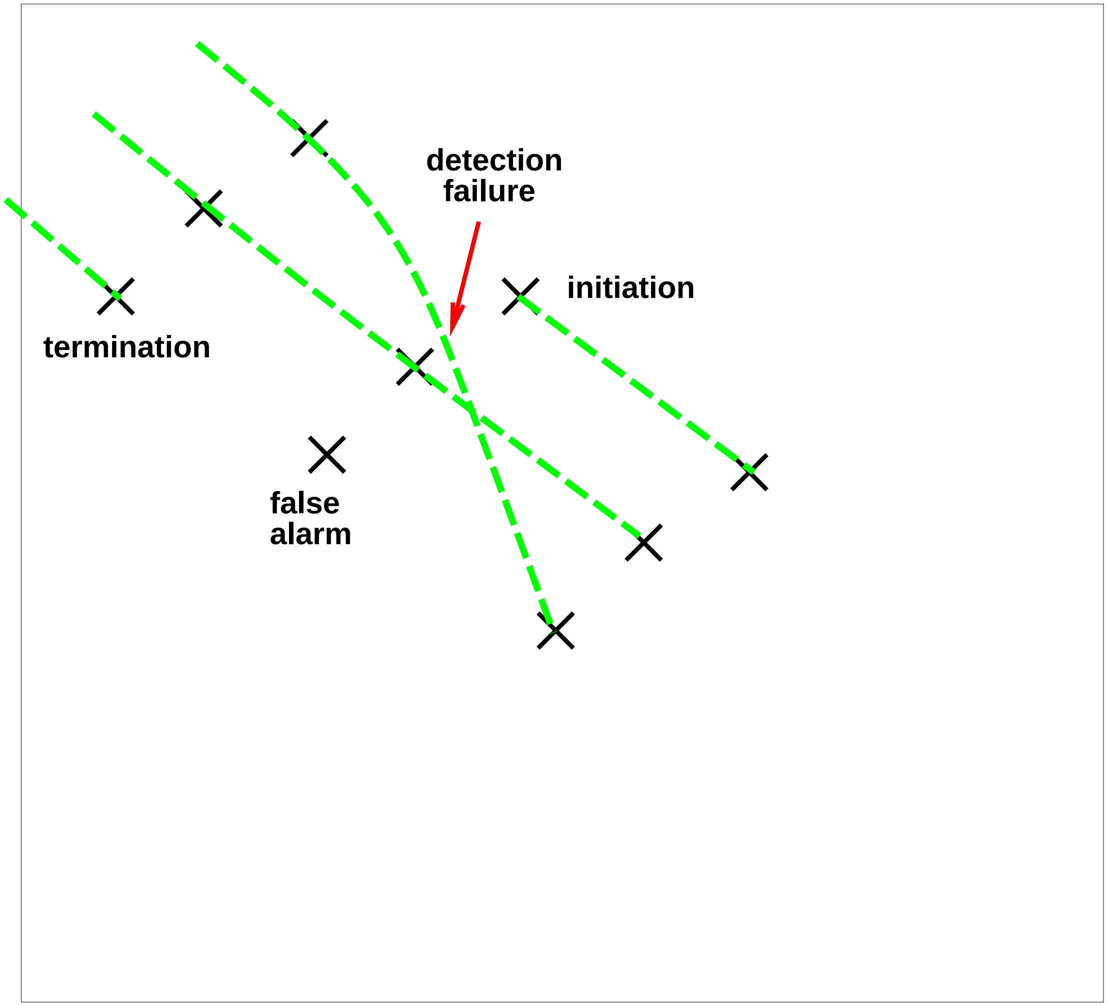
Example: classical data association



Example: classical data association



Example: classical data association



Example: modern data association

Modern data association

Same car?

Need to take into account competing matches!

Example: natural language

- What objects are referred to in the following natural language utterance?

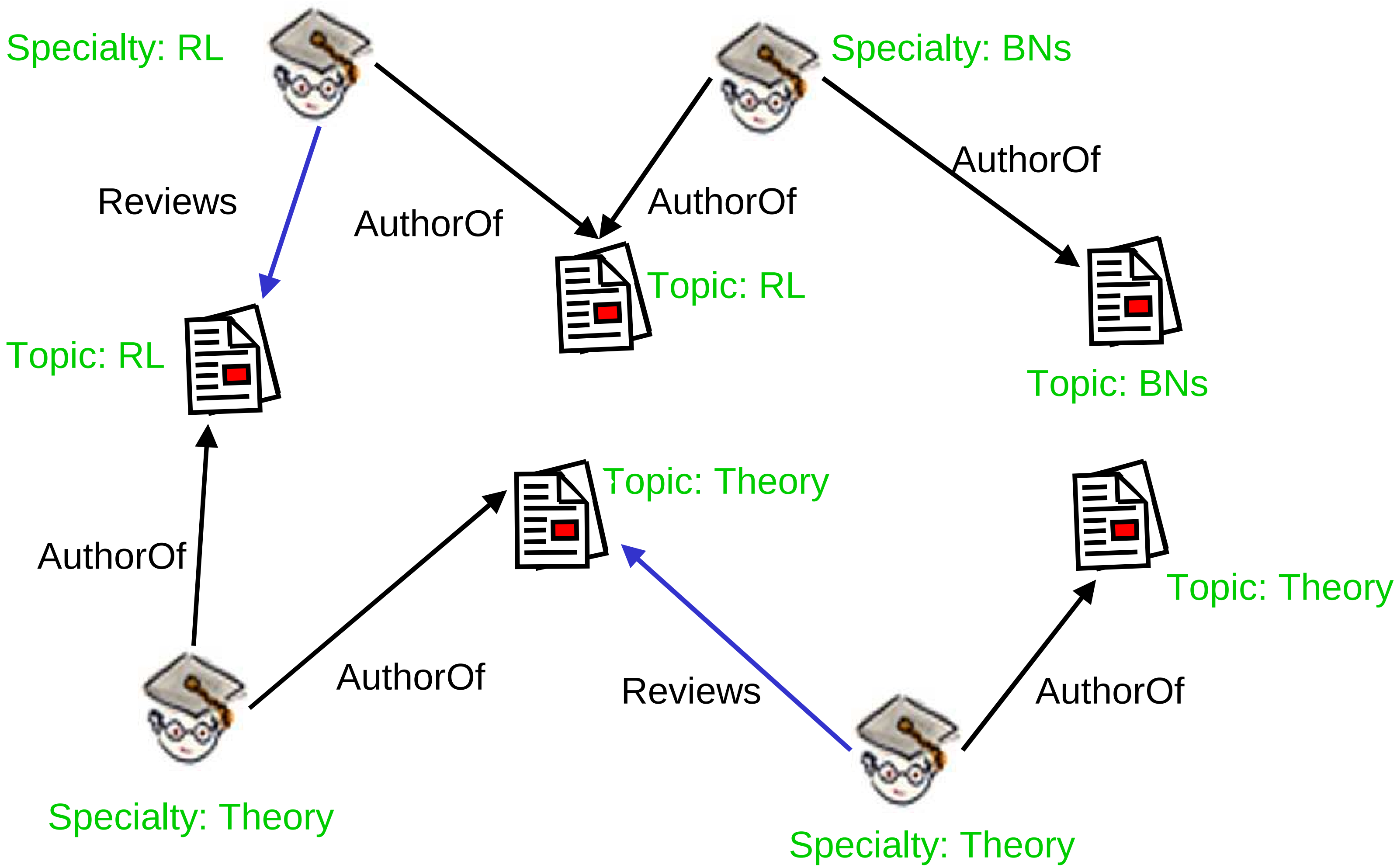
Example: vision

- What objects appear in this image sequence?

Outline

- Background and Motivation
  - Why we need more expressive formal languages for probability
  - Why unknown worlds matter
- Technical development
  - Relational models with known skeleton
  - Relational models with unknown relations
  - Unknown objects and identity uncertainty
- Applications
  - Citation matching
  - State estimation
- Open problems, future work
  - Why we need syntax and semantics

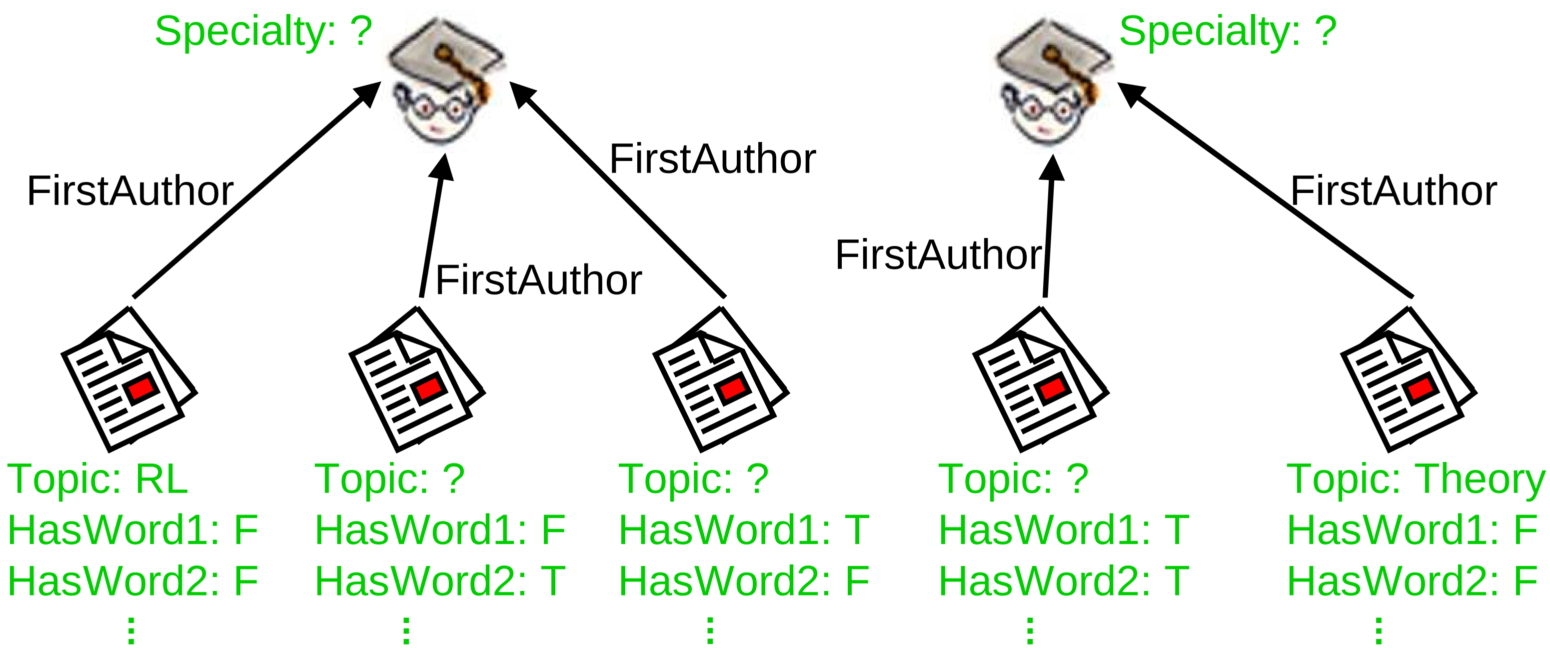
Objects, Attributes, Relations



Into the ~~Unknown~~ Random

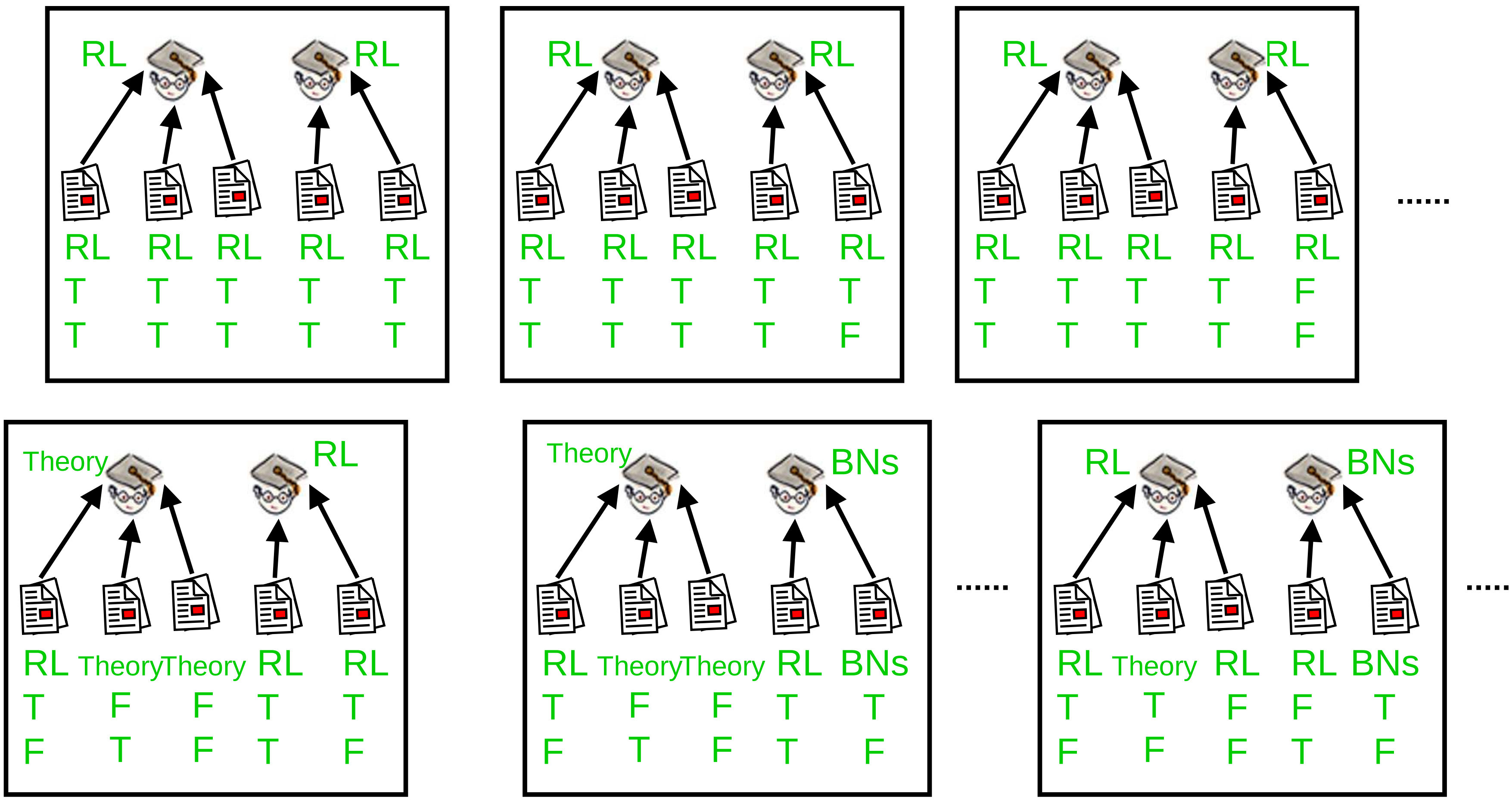
	Nonrandom, Fixed	Random (may be observed)
Attribute Uncertainty	Objects Relations	Attributes
Relational Uncertainty	Objects	Relations Attributes
Unknown Objects		Objects Relations Attributes

Attribute Uncertainty: Example

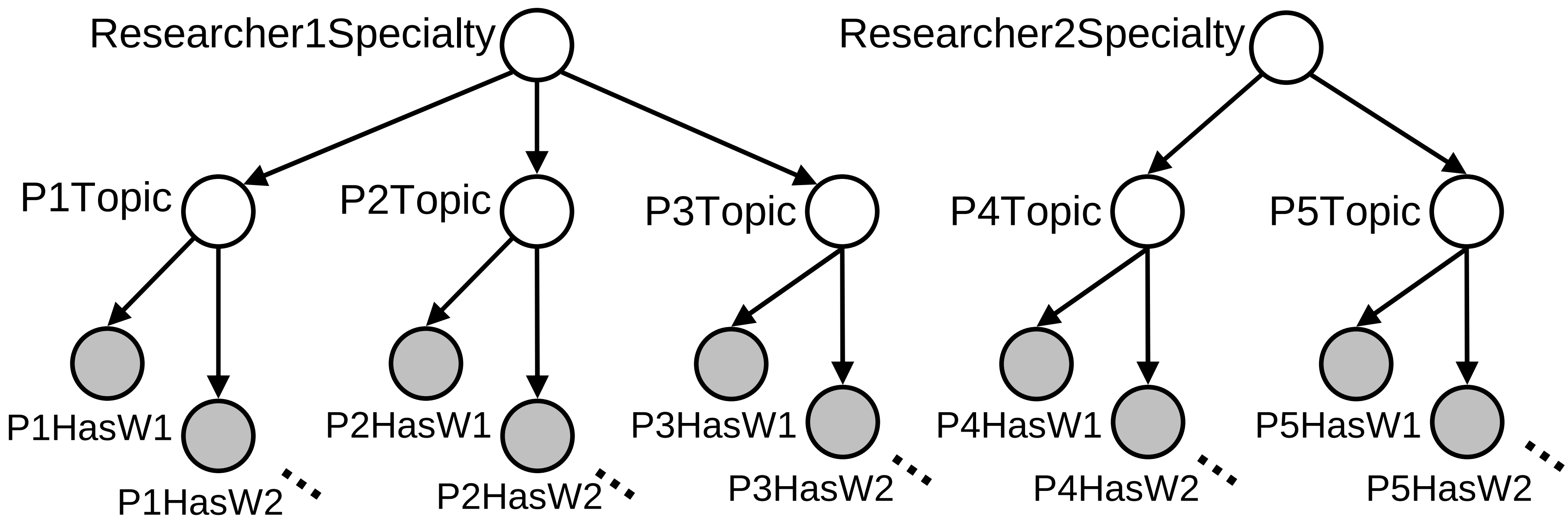


- Given paper text, relational structure, some topic labels
- Task: Classify remaining papers by topic
  - Collectively rather than in isolation

Possible Worlds



Bayesian Network



- Lots of repeated structure, tied parameters
- Different BN for each paper collection
- More compact representation?

Division of Labor

Lifted Probability Model	+	Relational Skeleton	→	Distribution over Outcomes
Dependency statements: "Topic(p) ~ …" Parameters Objects of closed types (Topic, Word)		Objects of open types (Researcher, Paper) Nonrandom relations		

- Assumptions: Same dependency statements and parameters apply
  - to all objects of open types
  - in all skeletons

First-Order Syntax

Typed Logic

- Types
  Researcher, Paper, Word,
  Topic, Boolean
- Functions, predicates
  FirstAuthor(p) $\rightarrow$ Researcher
  Speciality(r) $\rightarrow$ Topic
  Topic(p) $\rightarrow$ Topic
  HasWord(p, w) $\rightarrow$ Boolean

Statistics [e.g., BUGS by Gilks *et al*.]

- Index sets, value sets
  Researcher, Paper, Word
  Topic, {0, 1}
- Families of variables/parameters
  $\{A_j\}_{j \in \text{Paper}}$
  $\{S_r\}_{r \in \text{Researcher}}$
  $\{T_i\}_{i \in \text{Paper}}$
  $\{W_{ik}\}_{i \in \text{Paper},\, k \in \text{Word}}$

Surprisingly consistent!
We'll use Bayesian Logic (BLOG) notation [Milch et al., IJCAI 2005]

Dependency Statements

```
Specialty(r) ~ SpecialtyPrior();

Topic(p) ~ TopicCPD(Specialty(FirstAuthor(p)));
```

Logical term (nested function application) identifying parent node

— specifies how relations determine BN edges

```
HasWord(p, w) ~ WordCPD(Topic(p), w);
```

Conditional Dependencies

- Predicting the length of a paper
  - Conference paper: generally equals conference page limit
  - Otherwise: depends on verbosity of author
- Model this with conditional dependency statement

First-order formula as condition

```
Length(p)
    if ConfPaper(p) then ~ PageLimitPrior()
    else ~ LengthCPD(Verbosity(FirstAuthor(p)));
```

Variable Numbers of Parents

- What if we allow multiple authors?
  - Let skeleton specify predicate AuthorOf(***r***, ***p***)
- Topic(***p***) now depends on specialties of multiple authors
  - Number of parents depends on skeleton

Aggregation

- Can pass multiset into CPD

multiset defined by formula

```
Topic(p) ~ TopicAggCPD({Specialty(r) for Researcher r :
                                          AuthorOf(r, p)});
```

mixture of distributions conditioned on individual elements of multiset [Taskar *et al*., IJCAI 2001]

- Alternatively, apply aggregation function

aggregation function

```
Topic(p) ~ TopicCPD(Mode({Specialty(r) for Researcher r :
                                          AuthorOf(r, p)}));
```

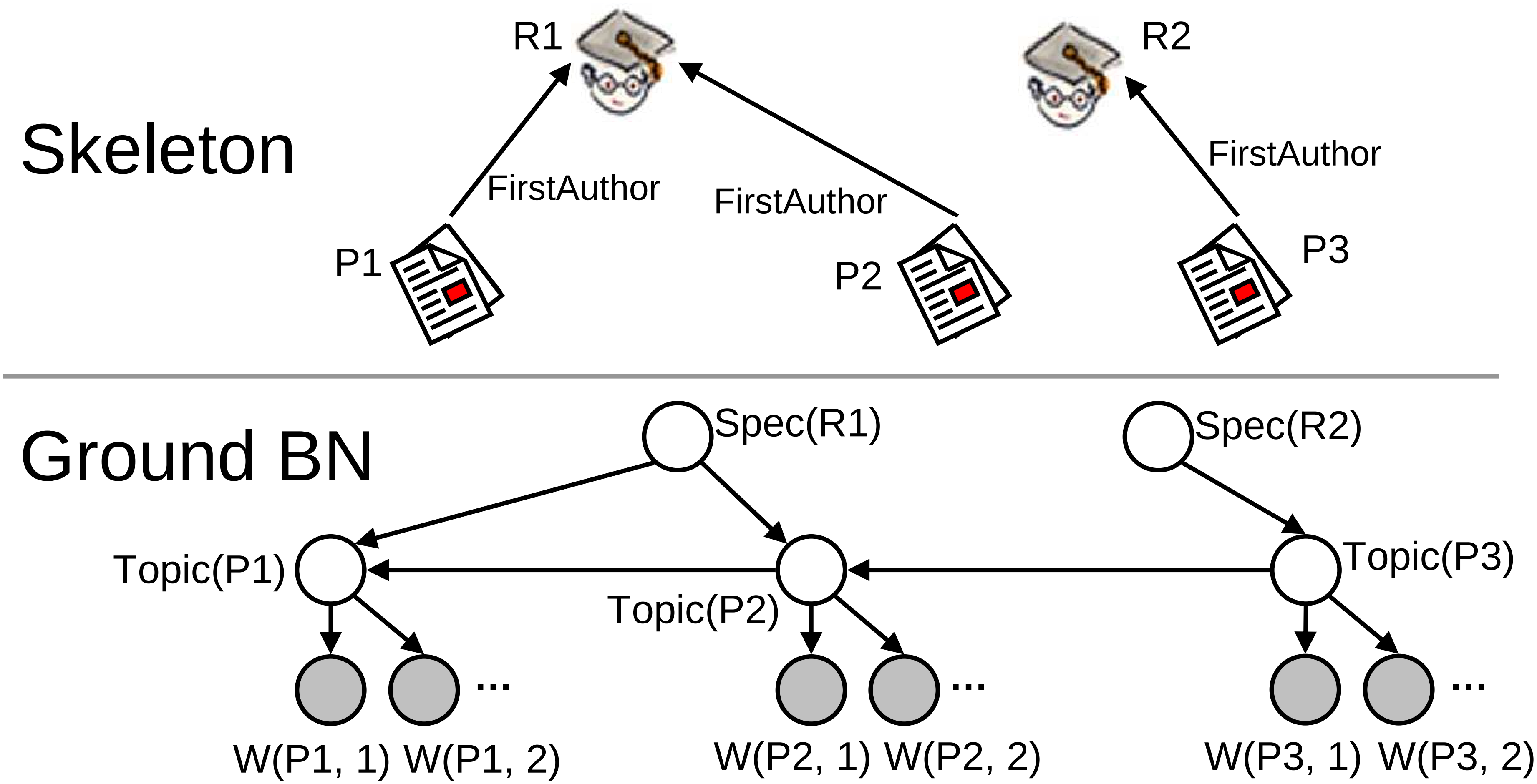
This is most of the syntax we need. On to semantics!

Semantics: Ground Bayes Net

- BLOG model defines ground Bayes net
- Nodes: one for each random function ***f*** and tuple of possible arguments ($\boldsymbol{o}_1,\ldots,\boldsymbol{o}_k$)
  - called basic random variables (RVs)
  - $\boldsymbol{o}_1,\ldots,\boldsymbol{o}_k$ are objects of closed types, or objects of open types listed in skeleton
- Edges and CPDs derived from dependency statements and skeleton

$$\texttt{Topic(p)} \sim \texttt{TopicCPD}\underbrace{\texttt{(Specialty(}\overbrace{\texttt{FirstAuthor(p)}}^{\text{specified by skeleton}}\texttt{))}}_{\text{specifies edge}}\texttt{;}$$

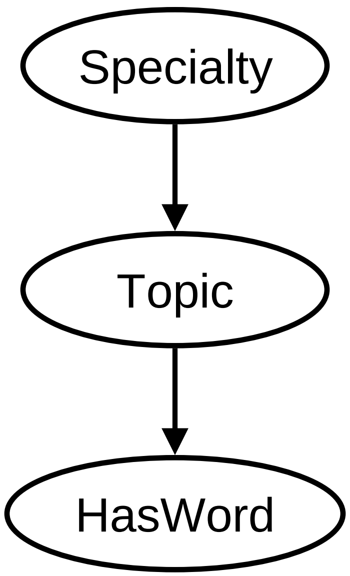
Ground BN


Skeleton
R1
R2
FirstAuthor
FirstAuthor
FirstAuthor
P1
P2
P3
Ground BN
Spec(R1)
Spec(R2)
Topic(P1)
Topic(P2)
Topic(P3)
...
...
...
W(P1, 1) W(P1, 2)
W(P2, 1) W(P2, 2)
W(P3, 1) W(P3, 2)

When Is Ground BN Acyclic?

[Koller & Pfeffer, AAAI 1998]

- Look at symbol graph
  - Node for each random function
  - Read off edges from dependency statements
- *Theorem*: If symbol graph is acyclic, then ground BN is acyclic for every skeleton

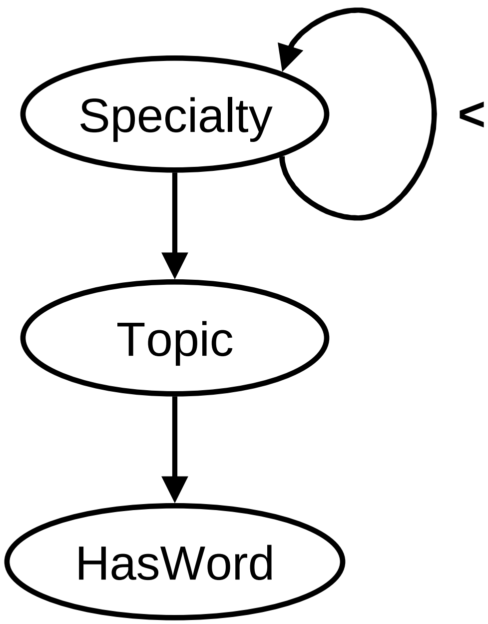


Acyclic Relations

[Friedman et al., ICML 1999]

- Suppose researcher’s specialty depends on his/her advisor’s specialty

```
Specialty(r)
  if Advisor(r) != null then
     ~ SpecCPD(Specialty(Advisor(r)))
  else ~ SpecialtyPrior();
```



- Symbol graph has self-loop!
- Require certain nonrandom functions to be acyclic: F(***x***) < ***x*** under some partial order
- Label edge B $\leftarrow$ A with:
  - “=”, if B(***x***) depends on A(***x***)
  - “<”, if B(***x***) depends on A(F(***x***)) for an acyclic F

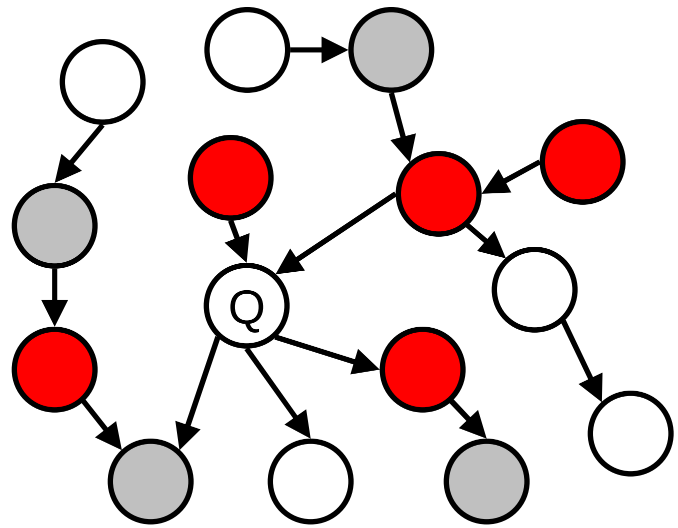
Acyclic Relations, cont'd

[Friedman et al., ICML 1999]

- Symbol graph is stratified if in every cycle, at least one edge is "<" and rest are "="
- *Theorem*: If symbol graph is stratified, then ground BN is acyclic for every skeleton that respects acyclicity constraints

Inference: Knowledge-Based Model Construction (KBMC)

- Construct relevant portion of the ground BN, apply standard inference algorithm
- A node is relevant if it:
  - is reachable from a query node along a path that is active given the evidence [Breese, *Comp. Intel.* 1992]
  - and is an ancestor of a query or evidence node

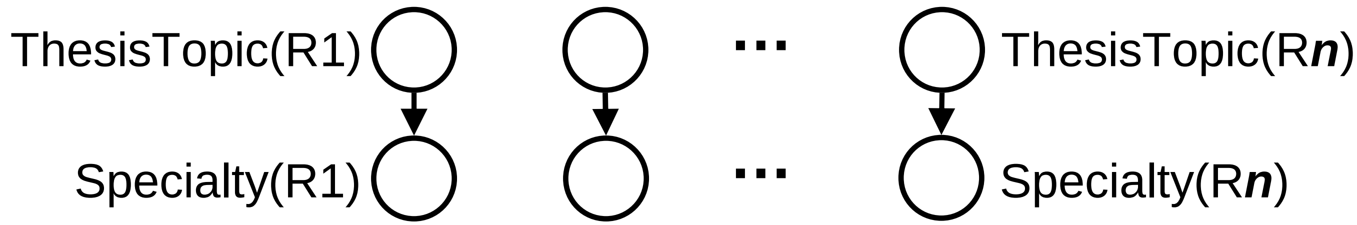


Do we have to construct ground BN at all?

First-Order Variable Elimination

[Pfeffer *et al*., UAI 1999; Poole, IJCAI 2003; Braz *et al*., IJCAI 2005]

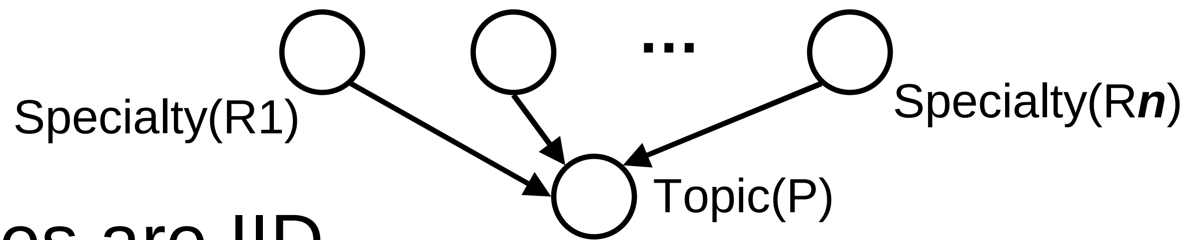
- Suppose: `Specialty(r) ~ SpecCPD(ThesisTopic(r));`
- With ***n*** researchers, part of ground BN is:



- Could sum out ThesisTopic(R) nodes one by one, taking O(nT^2) time for ***T*** topics
- But parameter sharing implies:
  - Summing same potential every time
  - Obtain same potential over Specialty(R) for each R
- Can just do summation once, eliminate whole family of RVs, store "lifted" potential on Specialty(***r***): time O(T^2)

First-Order VE and Aggregation

- Ground BN:



- Spec(***r***) variables are IID
- Topic(P) depends on them through an aggregation function
- In many cases, we know distribution for aggregate of IID variables [Pfeffer *et al*., IJCAI 1999]
  – mean, number having particular value, random sample, …
- Derive potential over Topic(P) analytically

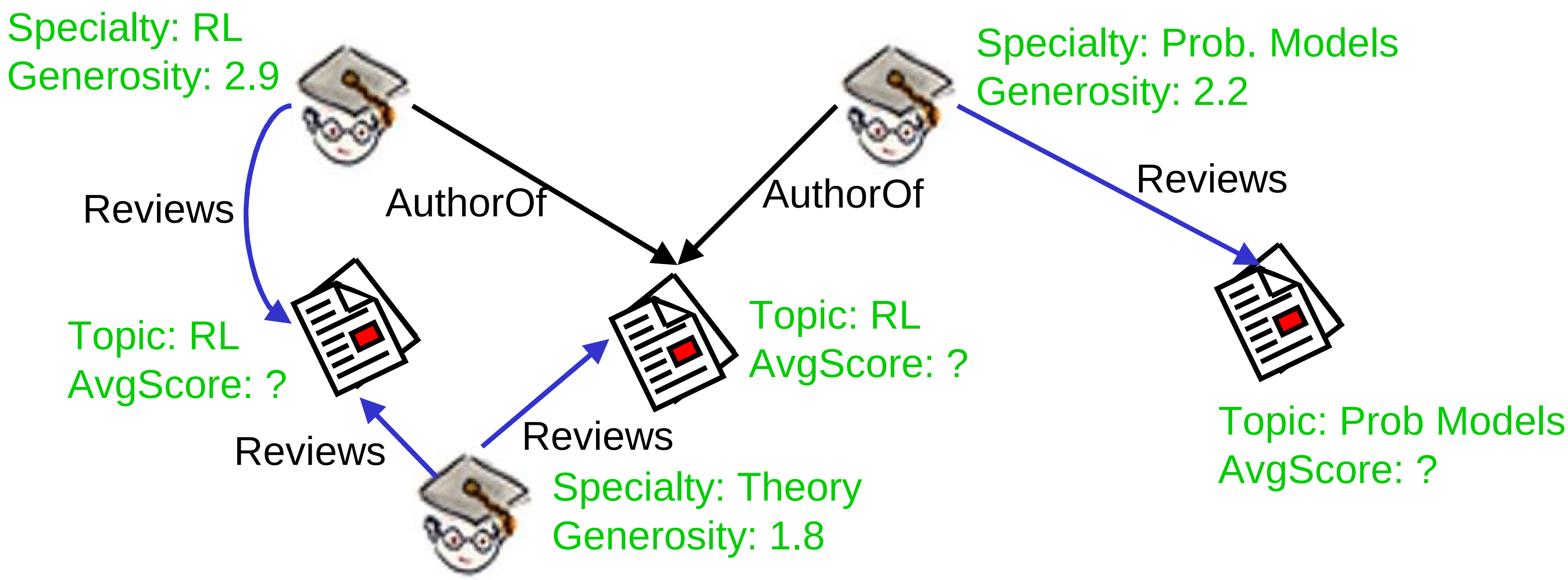
Limitations of First-Order VE

- Mass elimination of RVs only possible if they're generic: all have same potentials
- Elimination not efficient if RVs have many neighbors
  - Eliminating Specialty(R) for a researcher R who wrote many papers creates a potential over all those papers' Topic RVs

Into the Unknown

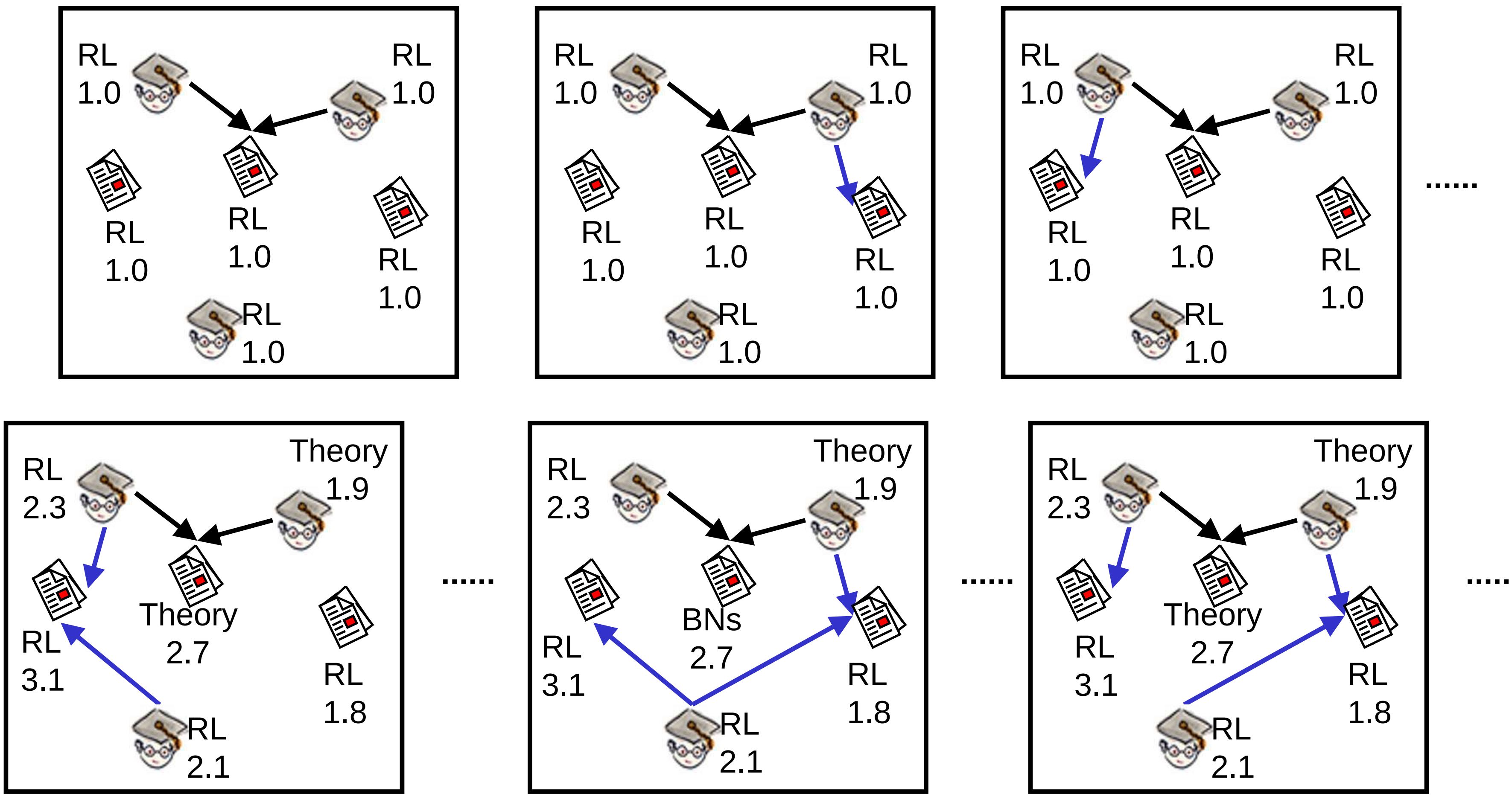
	Nonrandom, Fixed		Random
Attribute Uncertainty	Objects Relations		Attributes
Relational Uncertainty	Objects	→	Relations Attributes
Unknown Objects		→	Objects Relations Attributes

Relational Uncertainty: Example



- Questions: Who will review my paper, and what will its average review score be?
- Given: Authorship relation, paper topics, researcher specialties and generosity levels

Possible Worlds



Simplest Approach to Relational Uncertainty

[Getoor et al., ICML 2001]

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

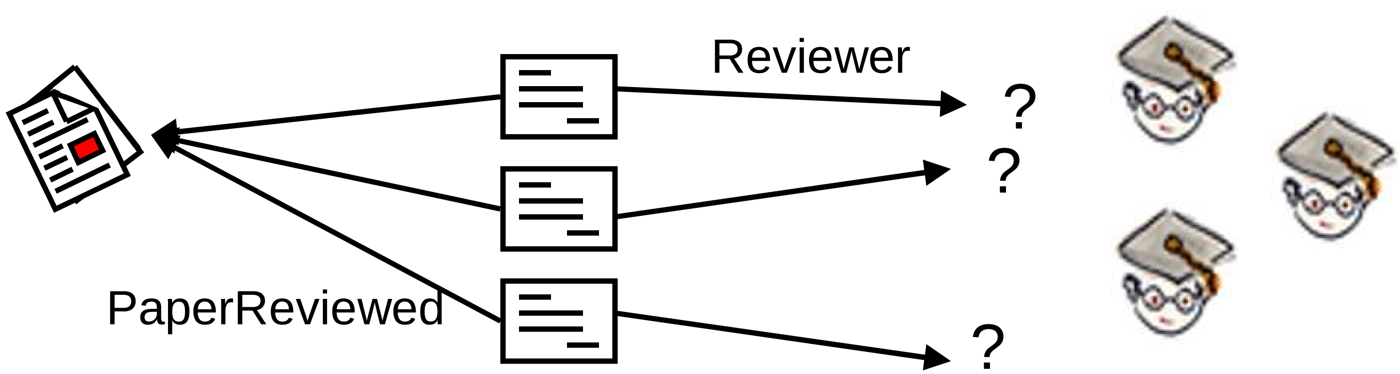
```
Reviews(r, p) ~ ReviewCPD(Specialty(r), Topic(p));
```

- 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

Another Approach: Reference Uncertainty

[Getoor et al., ICML 2001]

- 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***)



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., ICML 2001]
  - Weighted sampling, with weights determined by attributes [Pasula et al., IJCAI 2001]

BLOG Syntax for Reference Uncertainty

- Choosing based on Specialty attribute

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

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

- Choosing by weighted sampling:

```
Weight(rev, r) = CompatibilityWeight
                  (Topic(PaperReviewed(rev)), Specialty(r));

Reviewer(rev) ~ WeightedSample({(r, Weight(rev, r))
                                     for Researcher r});
```

set of pairs as CPD argument

Context-Specific Dependencies

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

random object

```
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

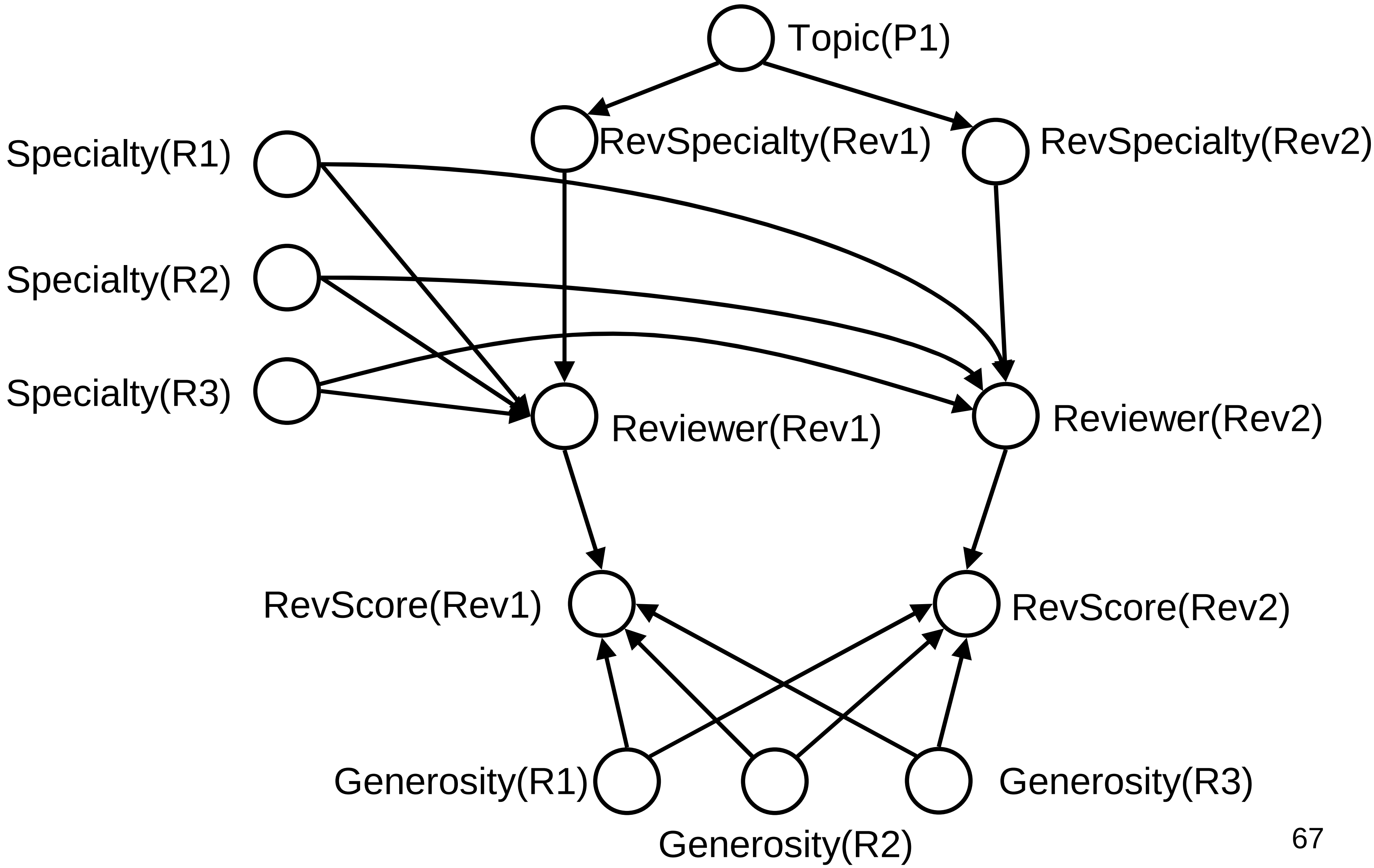
Semantics: Ground BN

- 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

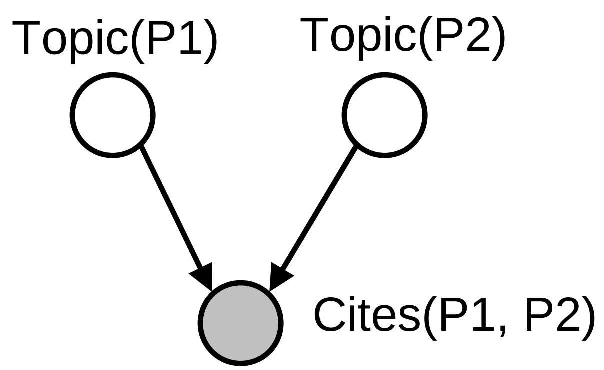

Topic(P1)
RevSpecialty(Rev1)
RevSpecialty(Rev2)
Specialty(R1)
Specialty(R2)
Specialty(R3)
Reviewer(Rev1)
Reviewer(Rev2)
RevScore(Rev1)
RevScore(Rev2)
Generosity(R1)
Generosity(R3)
Generosity(R2)

Random but Known Relations

- What a paper cites is an indicator of its topic
- Even if Cites relation is known, might want to model it as random [Getoor et al., ICML 2001]

```
Cites(p1, p2) ~ CitationCPD(Topic(p1), Topic(p2));
```

- Creates v-structures in ground BN, correlating topics of citing and cited papers



Inference

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

MCMC over Possible Worlds

- Metropolis-Hastings process: in world ω,
  - sample new world ω' from proposal distribution ***q***$(\omega'|\omega)$
  - 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 ***p***(ω)

Active Dependencies

- World probability $\boldsymbol{p}(\omega)$ is product over basic RVs
- For basic RV ***X***, active parents $\mathrm{Pa}_\omega(\boldsymbol{X})$ are RVs one must look at to evaluate right hand side of ***X***'s dependency statement in ω
- Example:

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

if Reviewer(Rev1) = Smith then

Pa_ω(RevScore(Rev1)) = {Reviewer(Rev1), Generosity(Smith)}

– other Generosity RVs are inactive parents

Computing Acceptance Ratio Efficiently

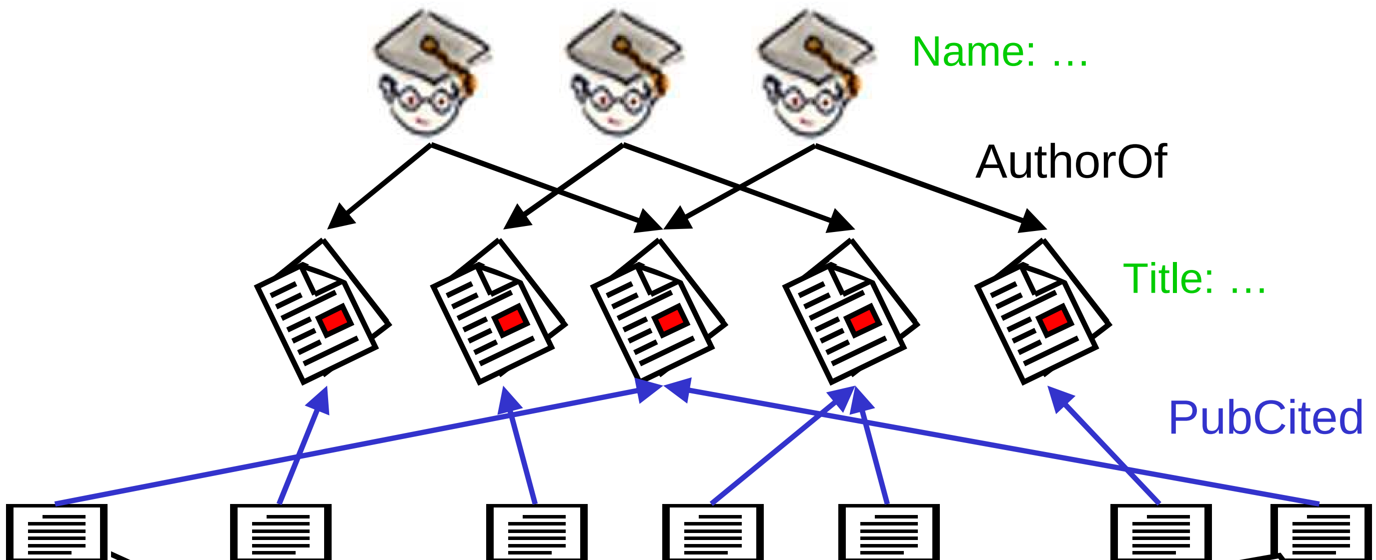
- World probability is $p(\omega) = \prod_X P(X = x_\omega \mid \mathrm{pa}_\omega(X))$

  where $\mathbf{pa}_\omega(\boldsymbol{X})$ is instantiation of $\mathrm{Pa}_\omega(\boldsymbol{X})$ in ω
- If proposal changes only RV ***X***, all factors not containing ***X*** cancel in $\boldsymbol{p}(\omega)$ and $\boldsymbol{p}(\omega')$
- And if $\mathbf{pa}_\omega(\boldsymbol{X})$ doesn't change, only need to compute $P(\boldsymbol{X}=\boldsymbol{x}_\omega \mid \mathbf{pa}_\omega(\boldsymbol{X}))$ up to normalization constant
  - If ***X*** gets value by weighted sampling, don't need to compute sum of weights [Pasula & Russell, IJCAI 2001]
- Result: Time to compute acceptance ratio often doesn't depend on number of objects

Into the Unknown

	Nonrandom, Fixed	Random
Attribute Uncertainty	Objects Relations	Attributes
Relational Uncertainty	Objects →	Relations Attributes
Unknown Objects	→	Objects Relations Attributes

Unknown Objects: Example



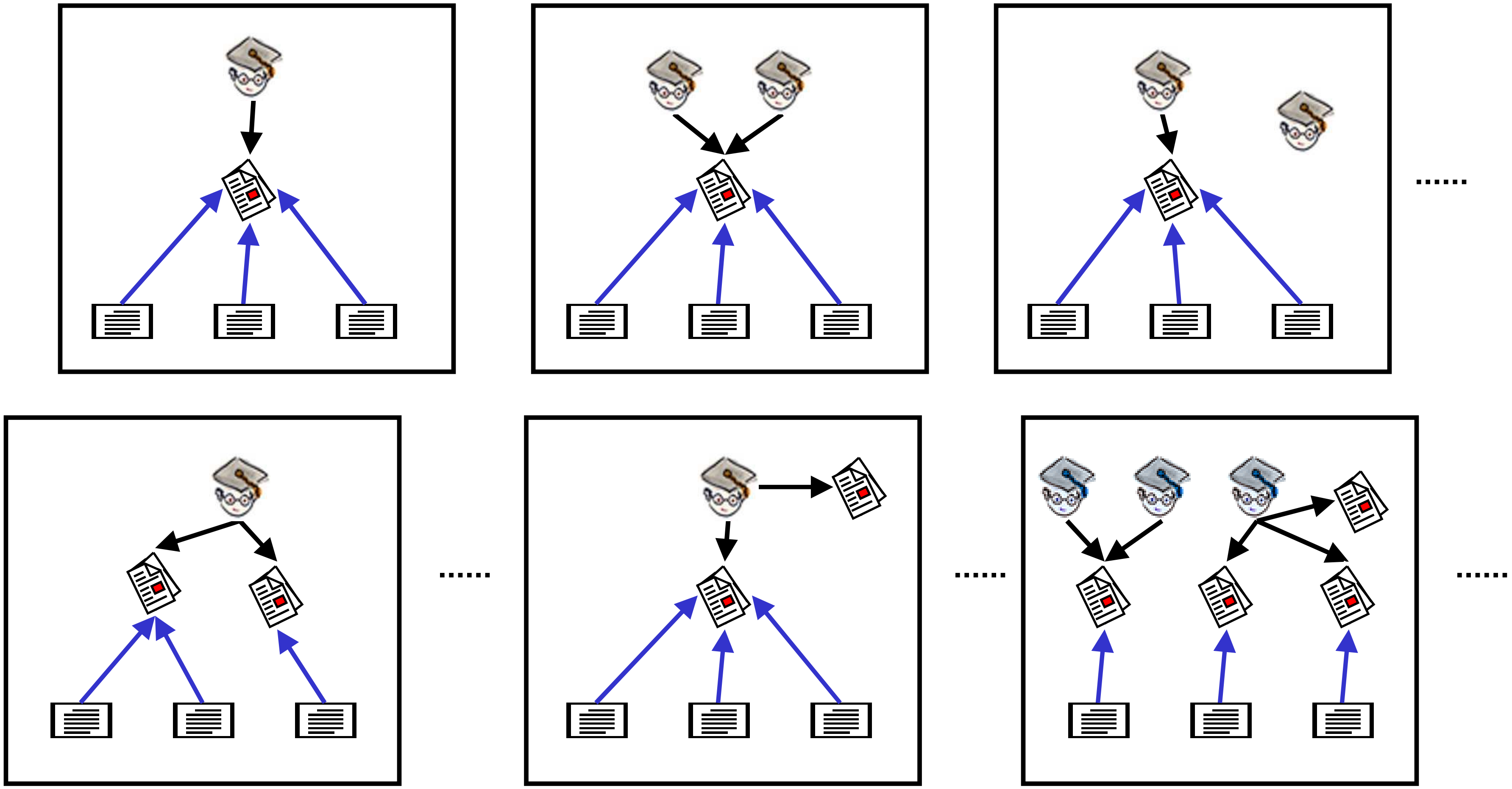
Russell, Stuart and Norvig, Peter. Articial Intelligence. Prentice-Hall, 1995.

S. Russel and P. Norvig (1995). Artificial Intelligence: A Modern Approach. Upper Saddle River, NJ: Prentice Hall.

$$\texttt{PubCited(Cit1)} \stackrel{?}{=} \texttt{PubCited(Cit7)}$$

Possible Worlds

(not showing attribute values)



How can we define a distribution over such outcomes?

Generative Process

[Milch *et al*., IJCAI 2005]

- 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
    - Includes setting the referent of a constant symbol (0-ary function)

Simplest Generative Process for Citations

```
#Paper ~ NumPapersPrior();
```

number statement

```
Title(p) ~ TitlePrior();
```

part of skeleton:
exhaustive list of distinct citations

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

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

familiar syntax for
reference uncertainty

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

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)));
```

Objects Generating Objects

- What if we want explicit distribution for |{Paper p: FirstAuthor(p) = r}|?
- Danger: Could contradict implicit distribution defined by:

```
#Paper ~ NumPapersPrior();
FirstAuthor(p) ~ Uniform({Researcher r});
```

- Solution:
  - Allow objects to generate objects
  - Designate FirstAuthor(***p***) as an origin function*
    - set when paper ***p*** is generated,
    - ties ***p*** back to the Researcher object that generated it
  - FirstAuthor(***p***) no longer has its own dependency statement

* Called “generating function” in [Milch et al., IJCAI 2005]

Number Statement Syntax

- Include FirstAuthor in number statement:

```
#Paper(FirstAuthor = r) ~ NumPapersPrior(Position(r));
```

CPD arguments can refer to generating objects

- Objects that satisfy this number statement applied to ***r*** are papers ***p*** such that FirstAuthor(***p***) = ***r***
- Right hand side gives distribution for number of objects satisfying this statement for any ***r***

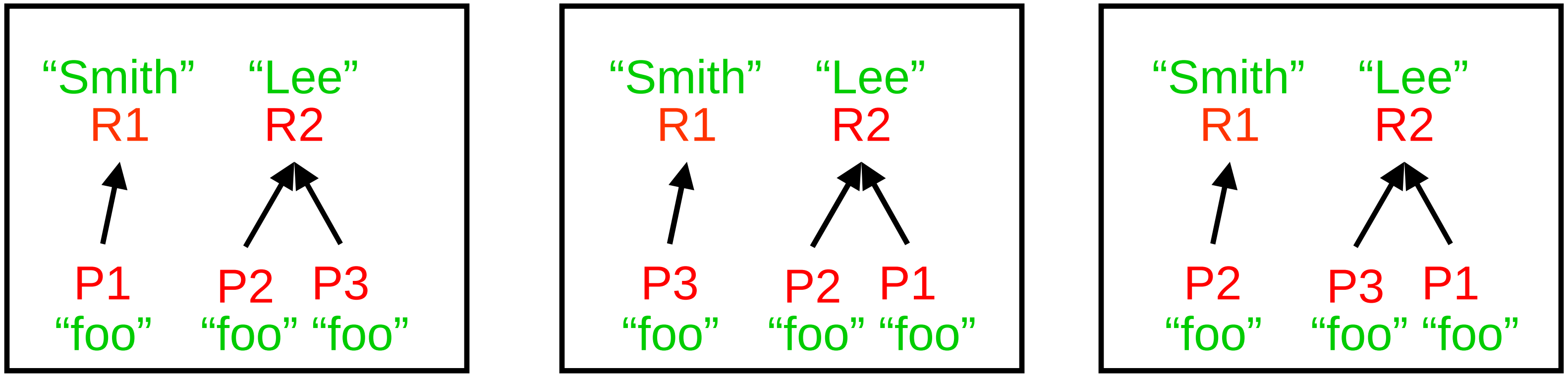
Semantics: First Try

- Have some set of potential objects that can exist in outcomes, e.g.

  R1, R2, R3, …
  P1, P2, P3, …

- Basic RVs:
  - Value of each random (non-origin) function on each tuple of potential objects
  - Number of objects that satisfy each number statement applied to each tuple of generating objects, e.g.,
    #Paper(FirstAuthor = R1), #Paper(FirstAuthor = R2), …
- Problem: Full instantiation of these RVs doesn't determine a world
  - Why not? Isomorphisms…

Isomorphic Worlds



- Worlds all correspond to same instantiation of basic RVs:

#Paper(FirstAuthor = R1) = 1, #Paper(FirstAuthor = R2) = 2, Title(P1) = "foo", …

- But differ in mapping from paper objects to researcher objects
- Proposal: Assign probabilities to basic RV instantiations, then divide uniformly over isomorphic worlds
  - Flaw: If infinitely many objects, then infinitely many isomorphic worlds

Solution: Structured Objects

[Milch *et al*., IJCAI 2005]

- Define potential objects to be nested tuples that encode generation histories

  (Researcher, 1)
  (Researcher, 2)

  (Paper, (FirstAuthor, (Researcher, 1)), 1)
  (Paper, (FirstAuthor, (Researcher, 1)), 2)

  (Paper, (FirstAuthor, (Researcher, 2)), 1)

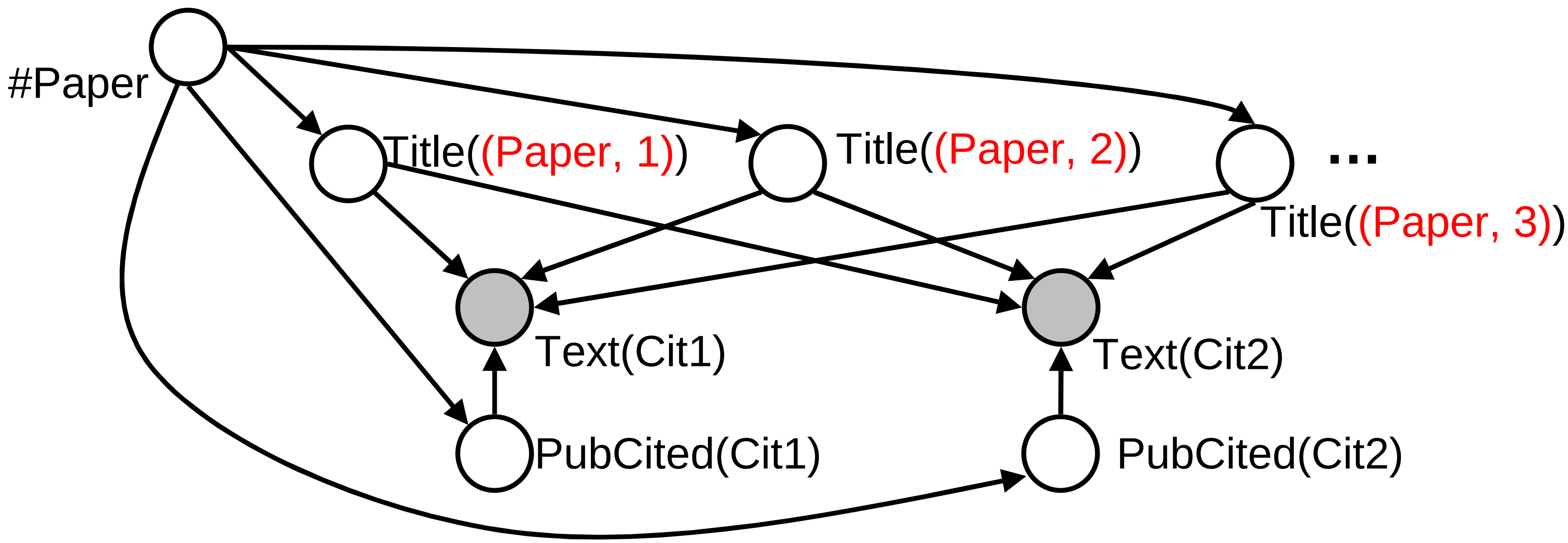
  …

- Restrict possible worlds so that, e.g.,

  FirstAuthor((Paper, (FirstAuthor, (Researcher, 1)), 1)) = (Researcher, 1)

- Now we have lemma: Full instantiation of basic RVs corresponds to at most one possible world

Semantics: Infinite Ground “BN”



- Infinitely many Title nodes, because infinitely many potential Paper objects
- Number RVs are parents of:
  - RVs indexed by objects that they generate
  - RVs that depend on set of generated objects

Semantics of Infinite BNs

- In finite case, BN asserts that probability of any full instantiation σ is product of CPDs:

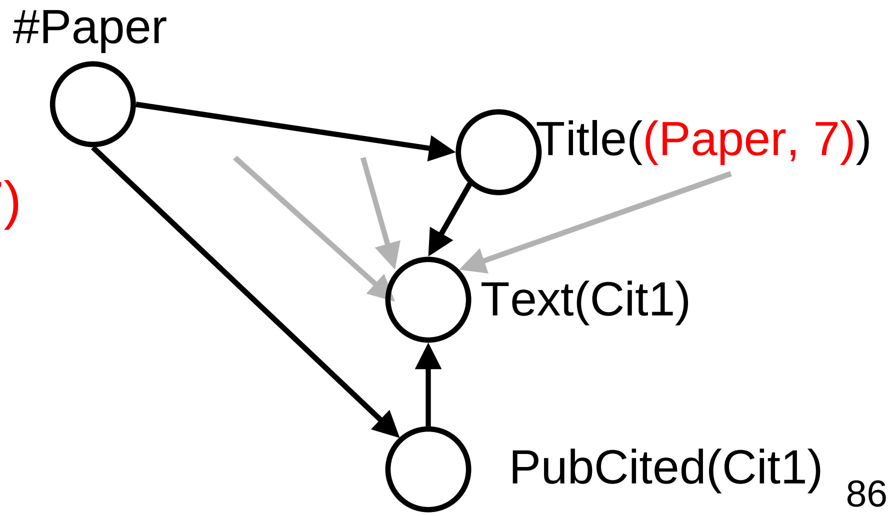
$$P(\sigma) = \prod_X p_X \underbrace{(\sigma_X \mid \sigma_{\mathrm{Pa}(X)})}_{\text{assumes vars}(\sigma)\text{ includes Pa}(\boldsymbol{X})}$$

- But with infinitely many variables, this infinite product is typically zero
- Fortunately, specifying probabilities for all *finite* instantiations determines joint distribution [Kolmogorov]
- But product expression only holds for certain finite instantiations

Self-Supporting Instantiations

- Instantiation σ is self-supporting if vars(σ) can be numbered $X_1,\ldots, X_N$ such that for each i, $\{X_1,\ldots, X_{i-1}\}$ includes all parents of X_i that are active given $\sigma_{(X_1,\ldots,X_{i-1})}$
  - Example:

#Paper = 12
Title((Paper, 7)) = "Foo"
PubCited(Cit1) = (Paper, 7)
Text(Cit1) = "foo"



Semantics of BLOG Models with Infinitely Many Basic RVs

- BLOG model asserts that for each finite, self-supporting instantiation σ,

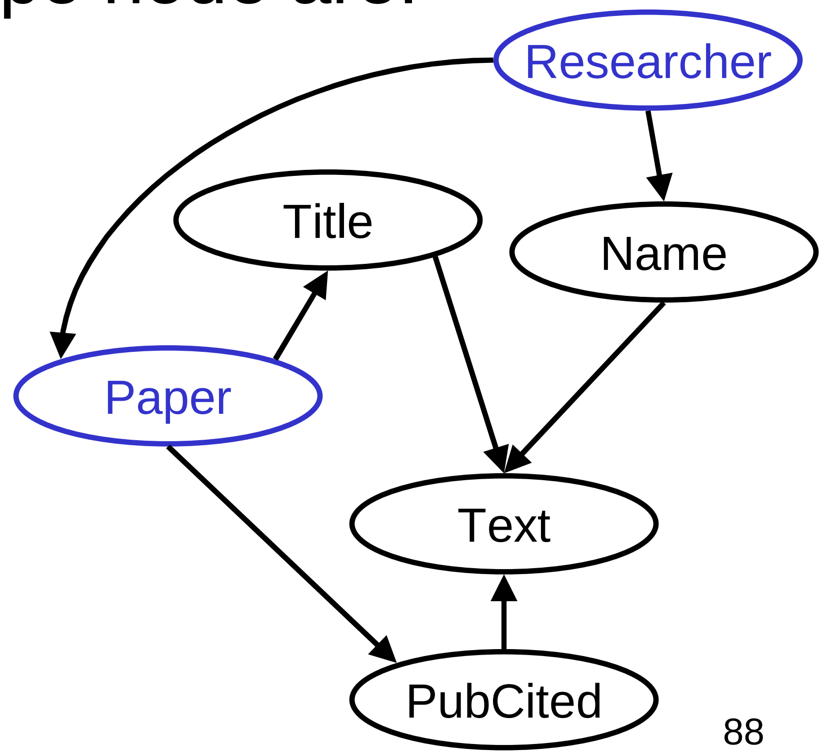
$$P(\sigma) = \prod_{X \in \text{vars}(\sigma)} p_{X_i}\left(\sigma_{X_i} \mid \sigma_{\{X_1,\ldots,X_{i-1}\}}\right)$$

- *Theorem 1*: If for each basic RV ***X*** and each possible world ω, there is a finite, self-supporting instantiation that agrees with ω and includes ***X***, then the BLOG model has a unique satisfying distribution

Can we tell when these conditions hold?

Symbol Graphs and Unknown Objects

- Symbol graph now contains not only random functions, but random types
- Parents of a function or type node are:
  - Functions and types that appear on the right hand side of dependency or number statements for this function/type
  - The types of this function/type's arguments or generating objects



Sufficient Condition for Well-Definedness

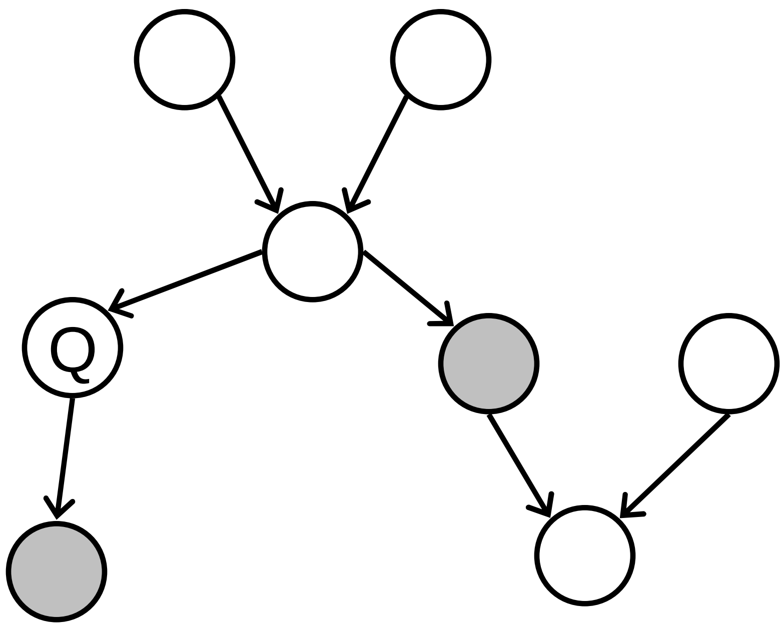
[Milch *et al*., IJCAI 2005]

- *Definition*: A BLOG model is well-formed if:
  - the symbol graph is stratified; and
  - all quantified formulas and set expressions can be evaluated by looking at a finite number of RVs in each possible world
- *Theorem 2*: Every well-formed BLOG model has a unique satisfying distribution

Inference for BLOG

- 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 3*: For any well-formed BLOG model, these sampling algorithms converge to correct probability for any query, using finite time per sampling step

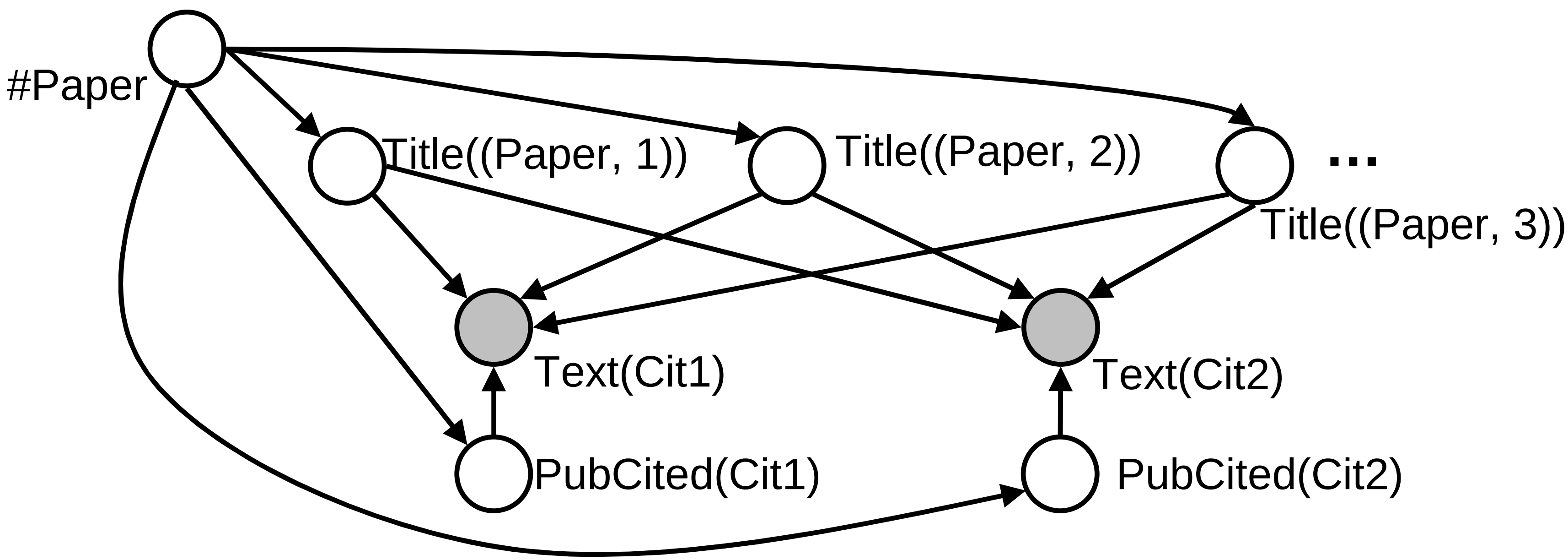
Approximate Inference by Likelihood Weighting



- Sample non-evidence nodes top-down
- Weight each sample by product of probabilities of evidence nodes given their parents
- Provably converges to correct posterior

Application to BLOG

- Only need to sample ancestors of query and evidence nodes
- But until we condition on PubCited(Cit1), Text(Cit1) has infinitely many parents
- Solution: interleave sampling and relevance determination



Likelihood Weighting for (Simplified) Citation Matching

Evidence:

Text(Cit1) = “foo”;
Text(Cit2) = “foob”;

Query:

#Paper

Instantiation

#Paper = 7
PubCited(Cit1) = (Paper, 3)
Title((Paper, 3)) = “Foo”
Text(Cit1) = “foo”
PubCited(Cit2) = (Paper, 3)
Text(Cit2) = “foob”

Weight: 1 x 0.8 x 0.2

Stack

Title((Paper, 3))
PubCited(Cit2)

Text(Cit2)
#Paper

```
#Paper ~ NumPapersPrior();
Title(p) ~ TitlePrior();
PubCited(c) ~ Uniform({Paper p});
Text(c) ~ NoisyCitationGrammar(Title(PubCited(c));
```

More realistically: use MCMC

Learning First-Order Models

- Parameters
  - Standard BN/MN learning with shared parameters
  - Can use EM if data is incomplete; leads back to the challenge of inference
- Structure
  - Maximize likelihood of data subject to model complexity penalty
  - Use some form of greedy local search [Friedman et al., IJCAI 1999; Getoor et al., ICML 2001; Kok and Domingos, ICML 2005]

BLOG and Mixture Models

- Simple BLOG model for citations is Bayesian mixture model with unknown number of clusters
  - Can also have relations among “clusters” (papers)
- BLOG and Dirichlet process mixtures
  - Can code up Dirichlet processes in BLOG
    - Special syntax introduced by [Carbonetto *et al*., UAI 2005]
    - Or represent stick-breaking process explicitly
  - Having infinitely many latent objects…
    - Sometimes makes sense, e.g., how many papers exist?
    - Sometimes doesn’t, e.g., how many aircraft are in the sky within ten miles of me?

Outline

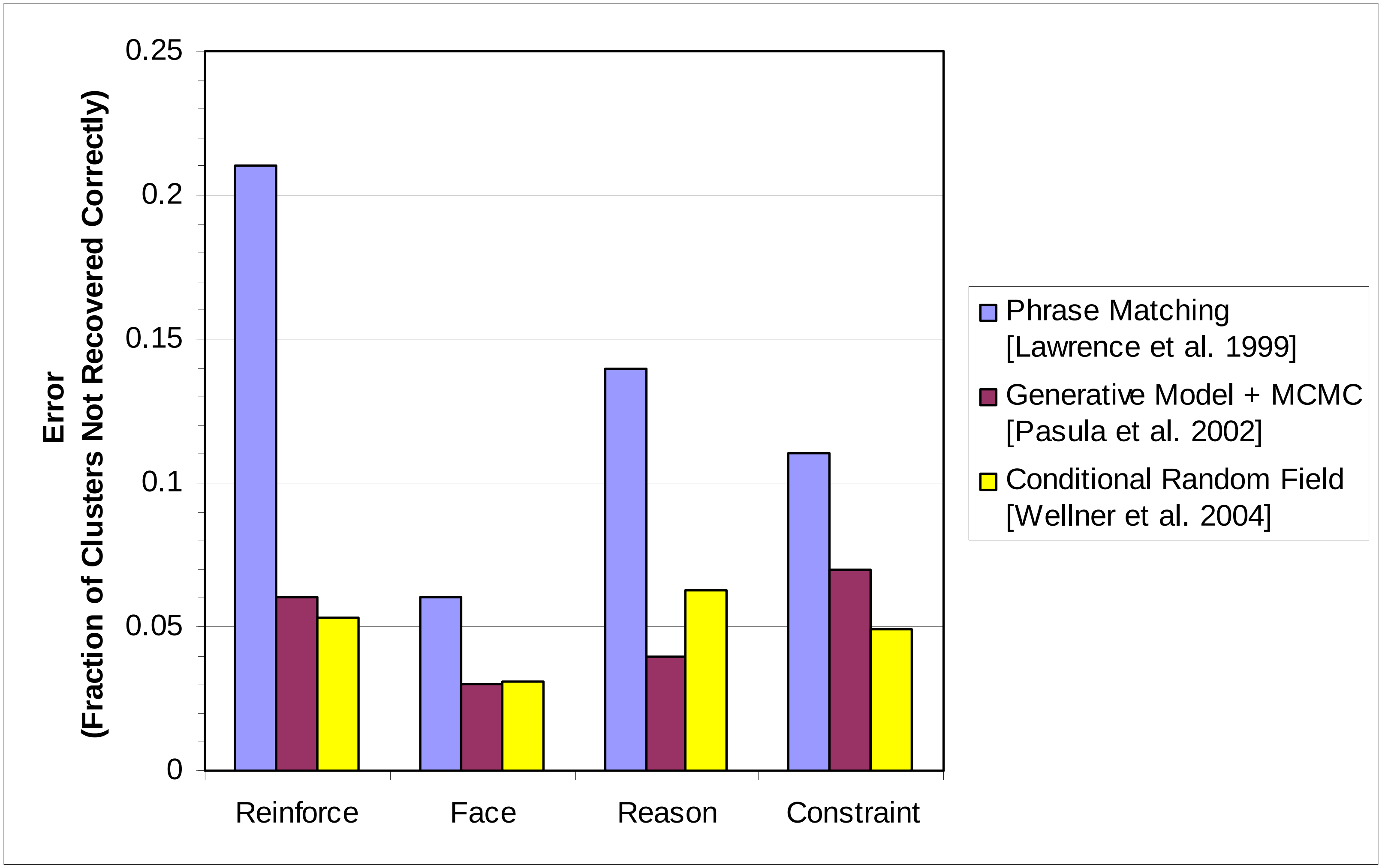
- Background and Motivation
  - Why we need more expressive formal languages for probability
  - Why unknown worlds matter
- Technical development
  - Relational models with known skeleton
  - Relational models with unknown relations
  - Unknown objects and identity uncertainty
- Applications
  - Citation matching
  - State estimation
- Open problems, future work
  - Why we need syntax and semantics

Citation Matching

[Pasula *et al*., NIPS 2002]

- 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 cluster recombination proposals
  - Guided by “canopies” of similar citations
  - Accuracy stabilizes after ~20 minutes

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%

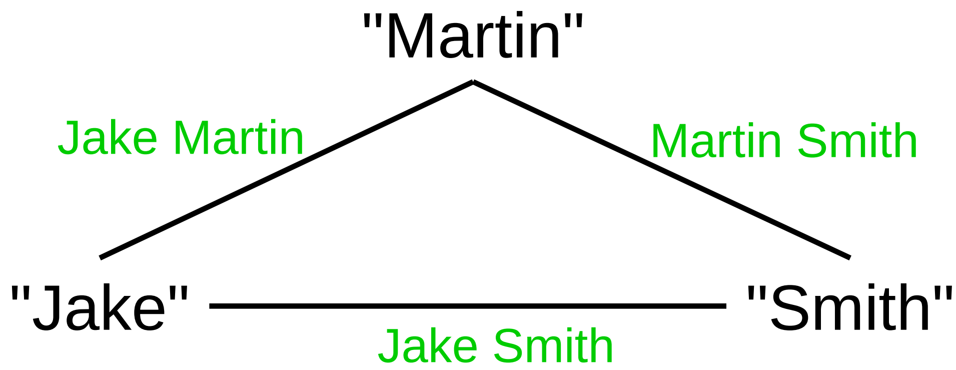
Can use inferred knowledge for disambiguation

Undirected Representation: Coref Variables

[McCallum & Wellner, NIPS 2004; Richardson & Domingos, SRL 2004]

- Don't represent unknown objects
- Instead, have predicate Coref(Cit1, Cit2)
- Advantage: set of RVs is fixed, finite
- Drawbacks:
  - parameters may be corpus-specific
  - true attributes of papers not represented anywhere
- Alternative: identify papers with subsets of citations [Culotta & McCallum, Tech Report 2005]

Where Pairwise Scores Fall Short



- Each pair of names is compatible
  - "Martin" serves as surname with "Jake", and as given name with "Smith"
- But it's unlikely that someone would be called by all three of these names

Pre-application: traffic monitoring

Goal: estimate current link travel time,
long-term origin-destination counts

Data association calculation

- Assignment ω specifies which observations belong to which vehicle
- $E(f|\text{data}) = \Sigma_\omega f(\omega,\text{data}) P(\text{data}|\omega) P(\omega)$

  $= \Sigma_\omega f(\omega,\text{data}) P(\omega) \Pi_i P(\text{data}_i)$

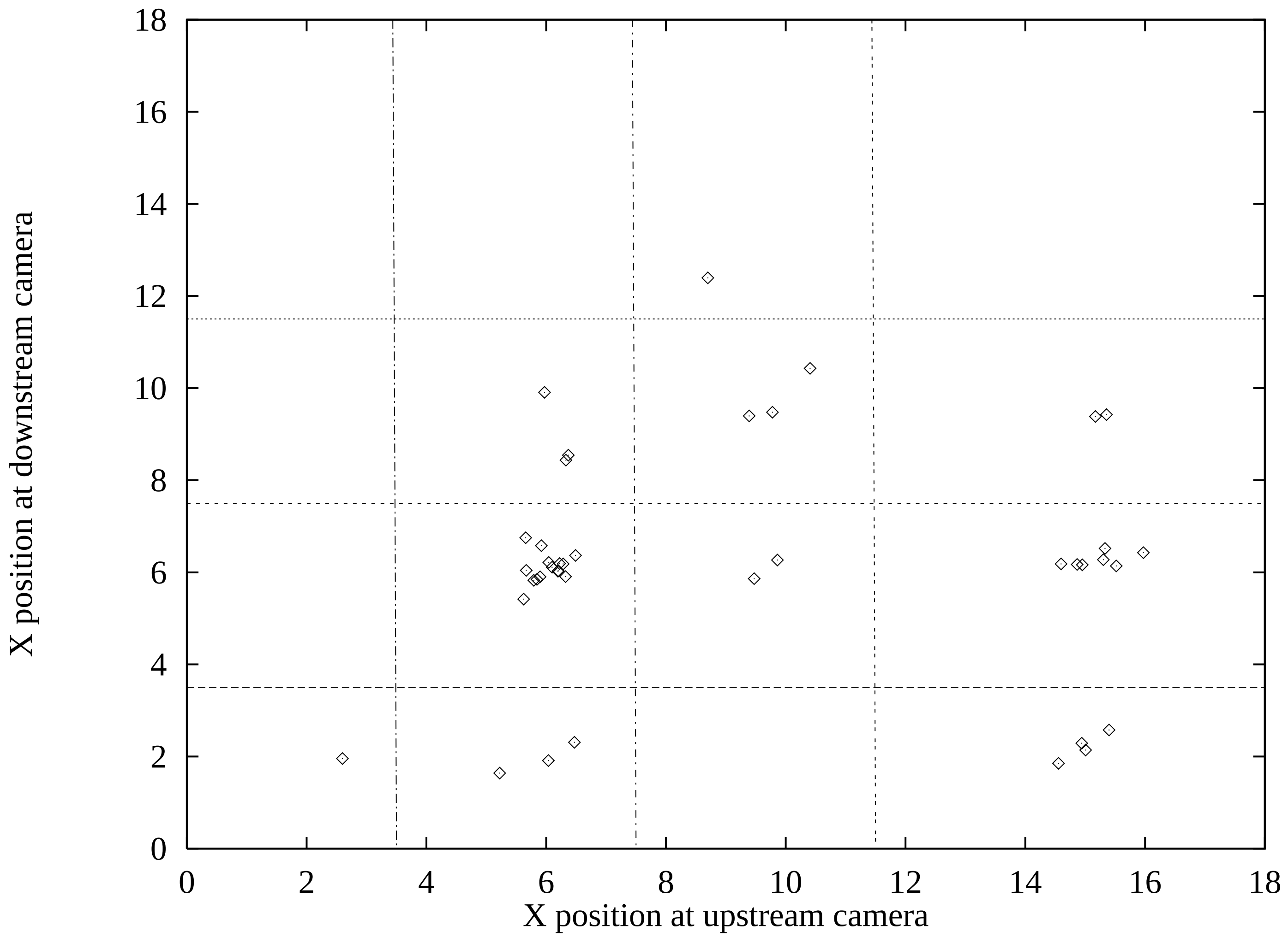
  i.e., likelihood factors over vehicles given a specific assignment

Observations and models

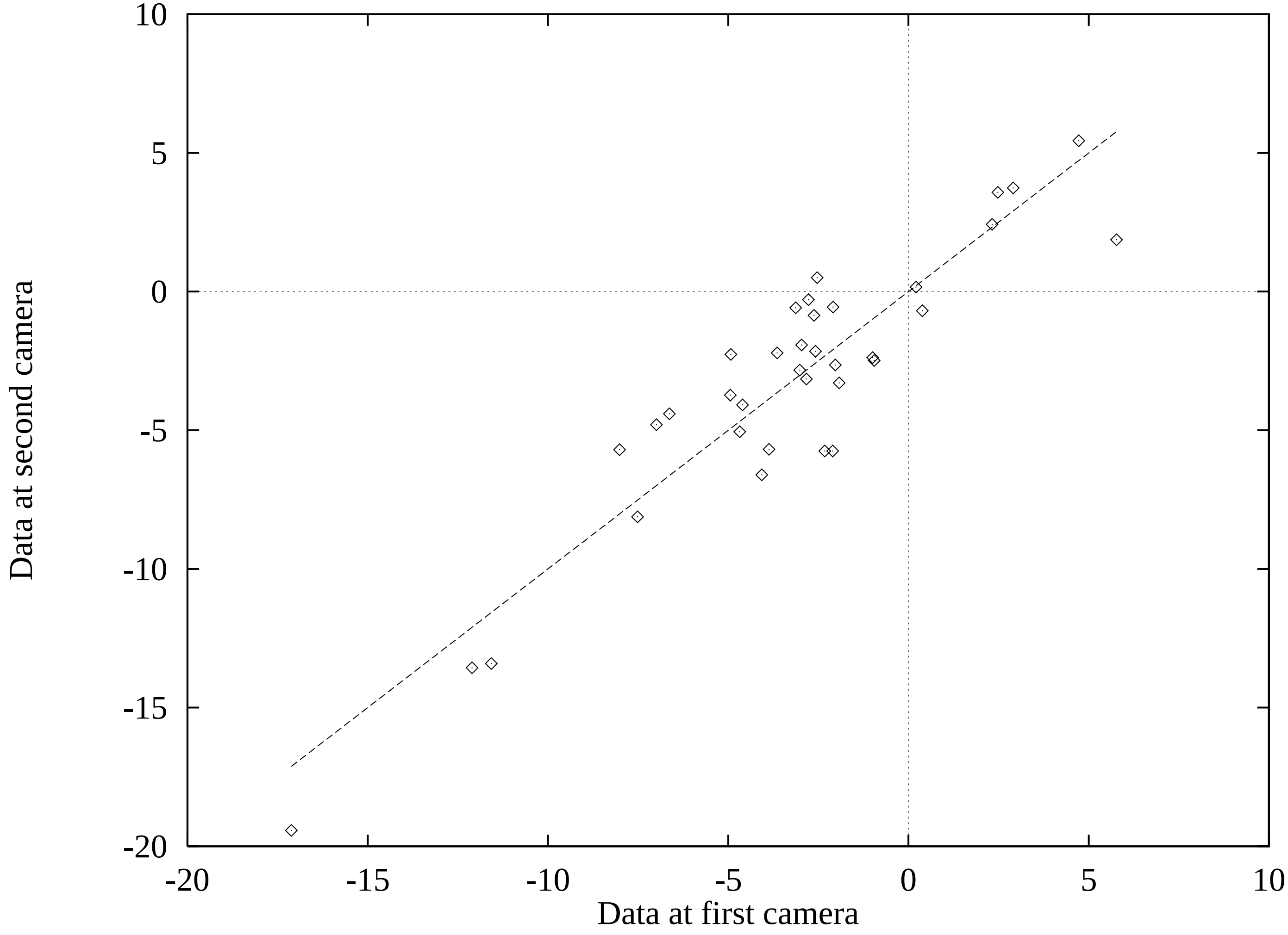
- Lane position (x)
  - Discrete model $P(x^d|x^u)$
- Arrival time t, speed s
  - $P(t^d|t^u)$ Gaussian with mean, variance dependent on x^u, x^d, s^d, s^u
- Colour -- h,s,v + colour histogram C
  - Camera-specific Gaussian noise
- Width, length+height
  - Camera-specific Gaussian noise

All parameters time-varying, learned online

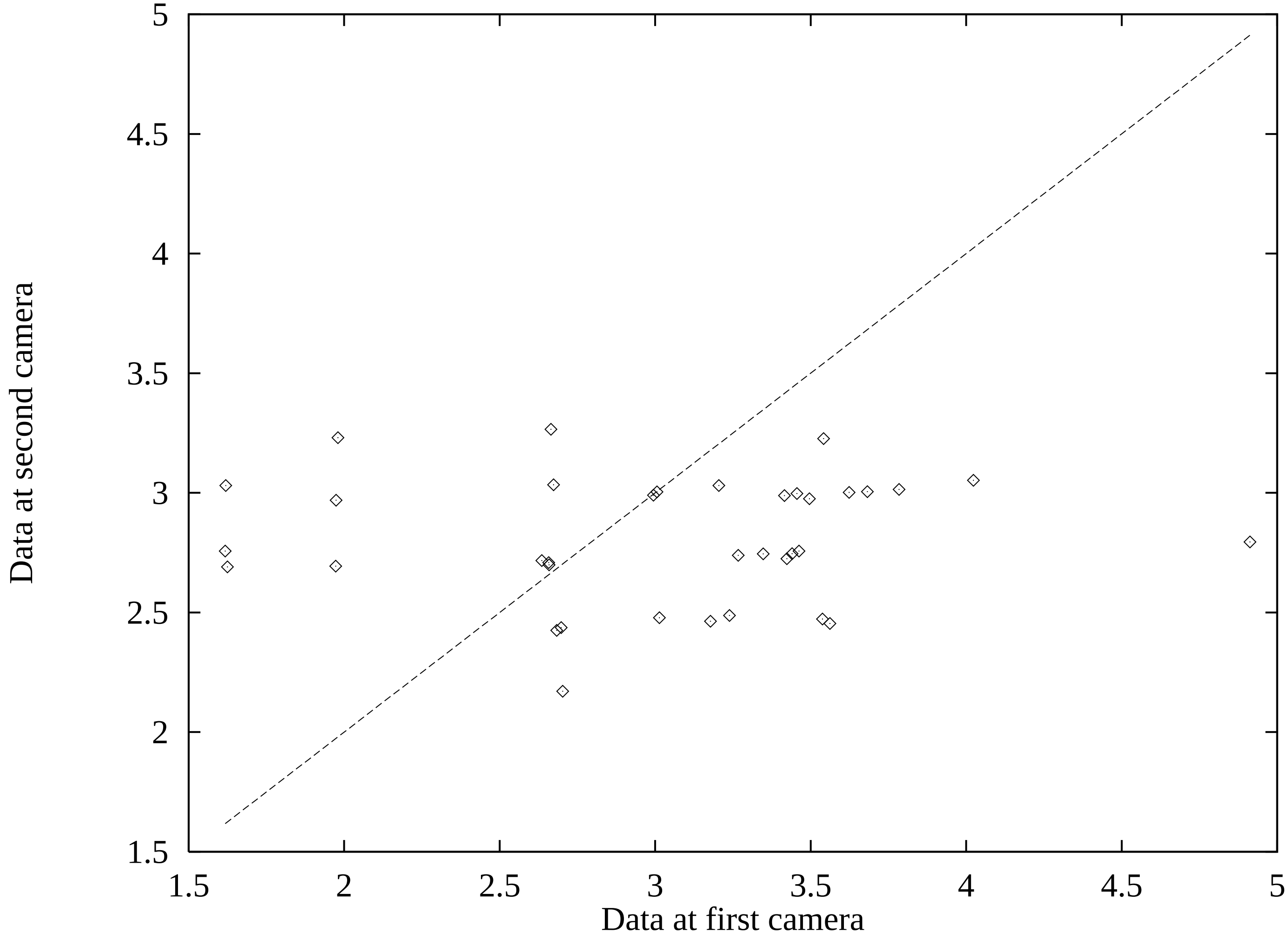
Lane correlation data



Hue correlation data



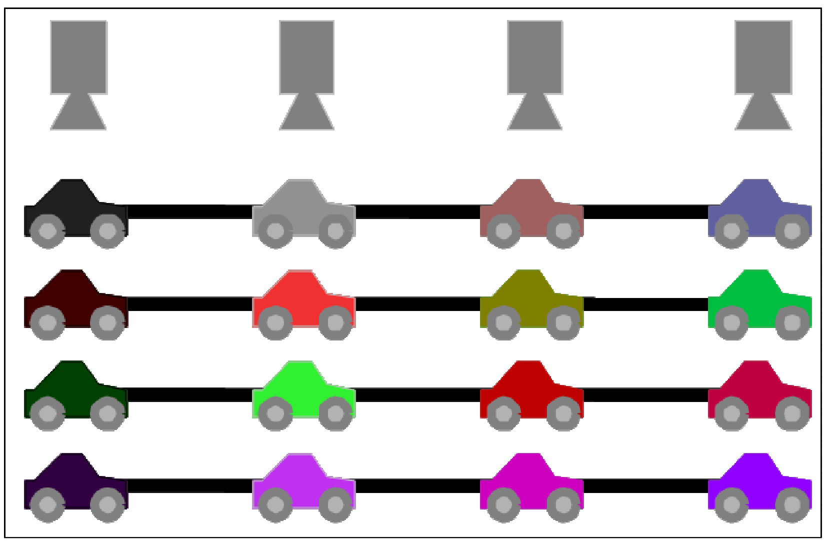
Width correlation data

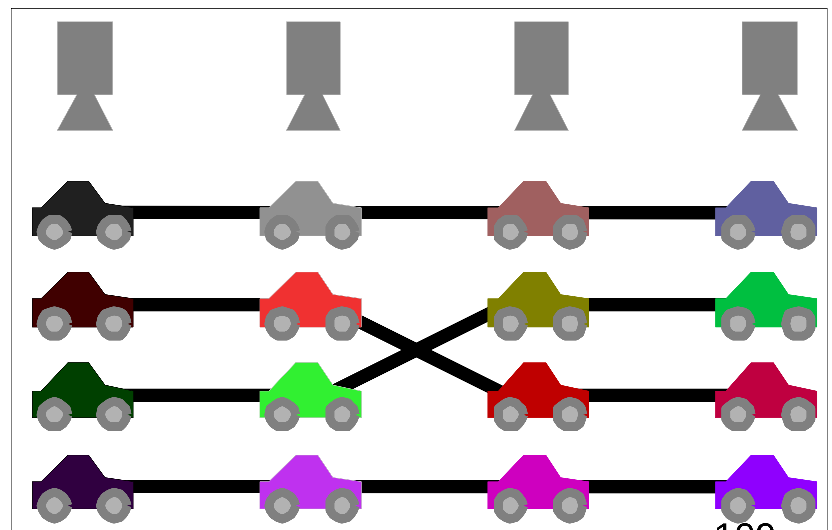


Inference

Rao-Blackwellized Decayed MCMC Filter

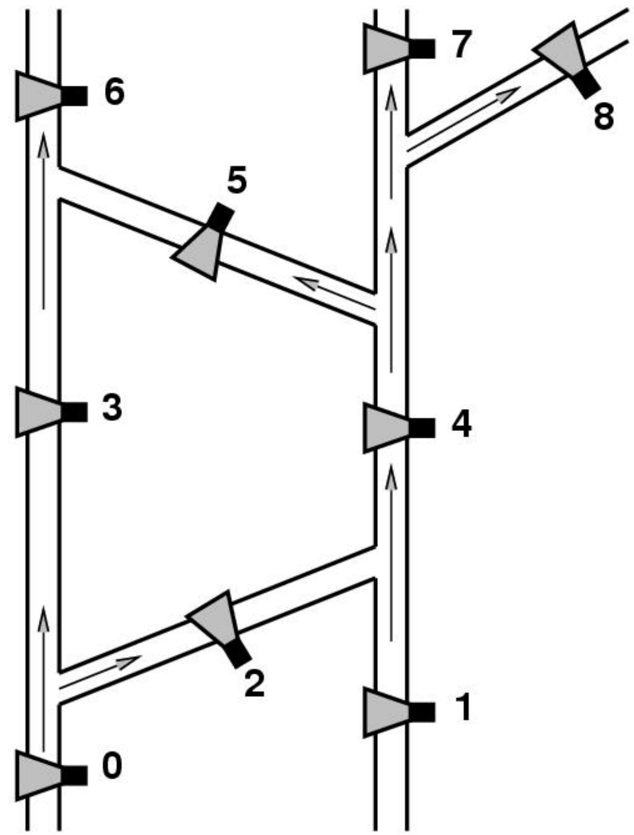
- Given assignment ω, likelihood factors into vehicle trajectories; Kalman filter on each
- MCMC proposes pairwise trajectory exchanges [polytime convergence for two cameras]

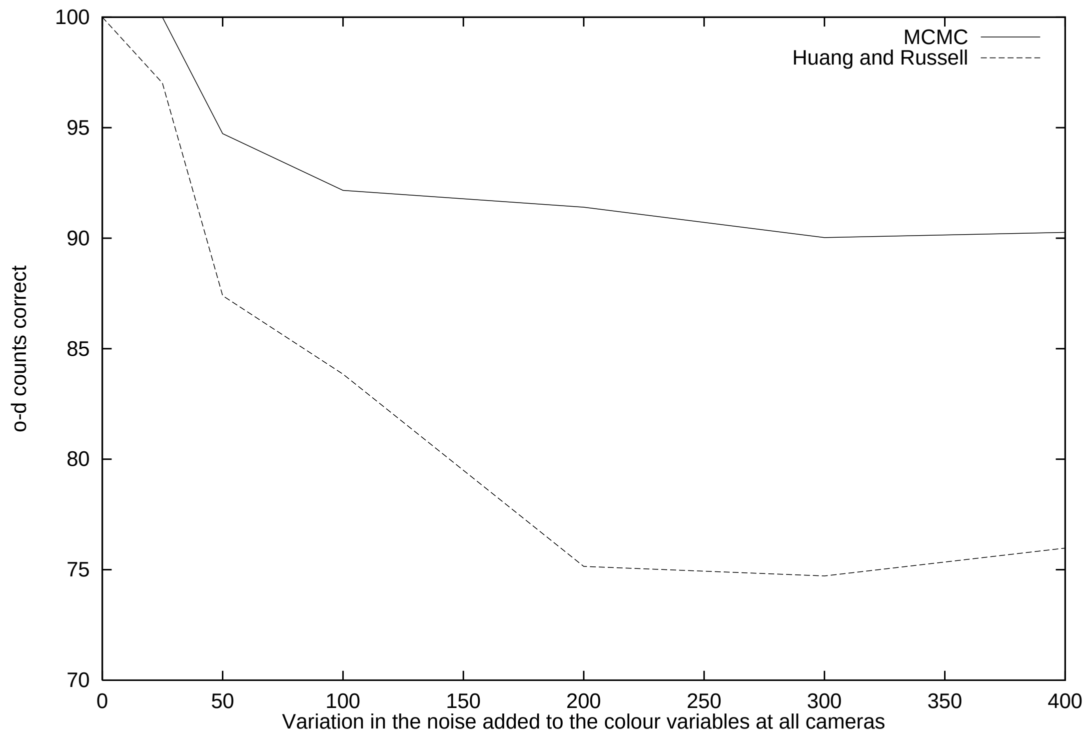




Results

Human-level performance on small real sample; beat previous best methods on 1200-vehicle simulation





State Estimation for “Aircraft”

- Dependency statements for simple model:

```
#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

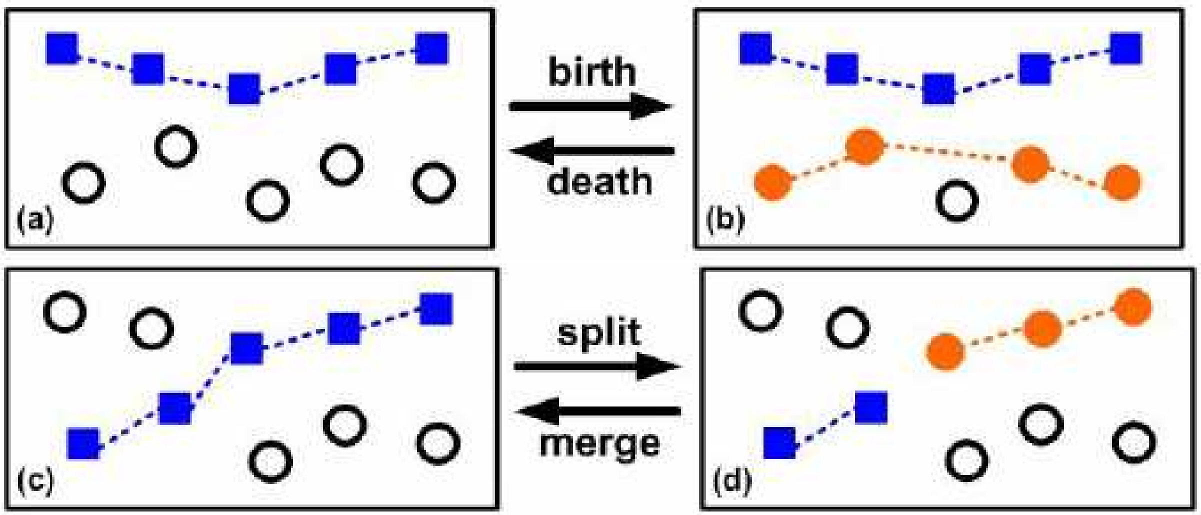
```
#Aircraft(EntryTime = t) ~ NumAircraftPrior();

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

[Oh et al., CDC 2004]

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

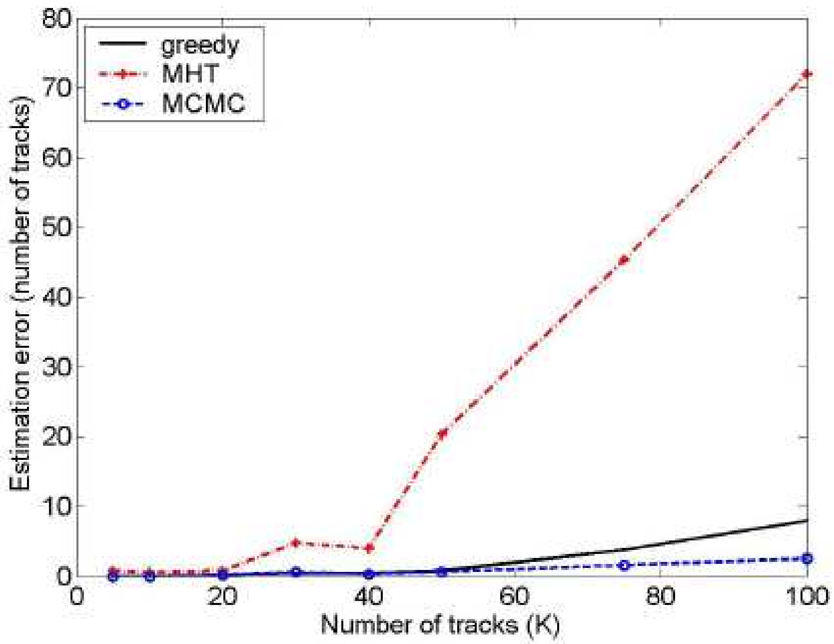


[Figures by Songhwai Oh]

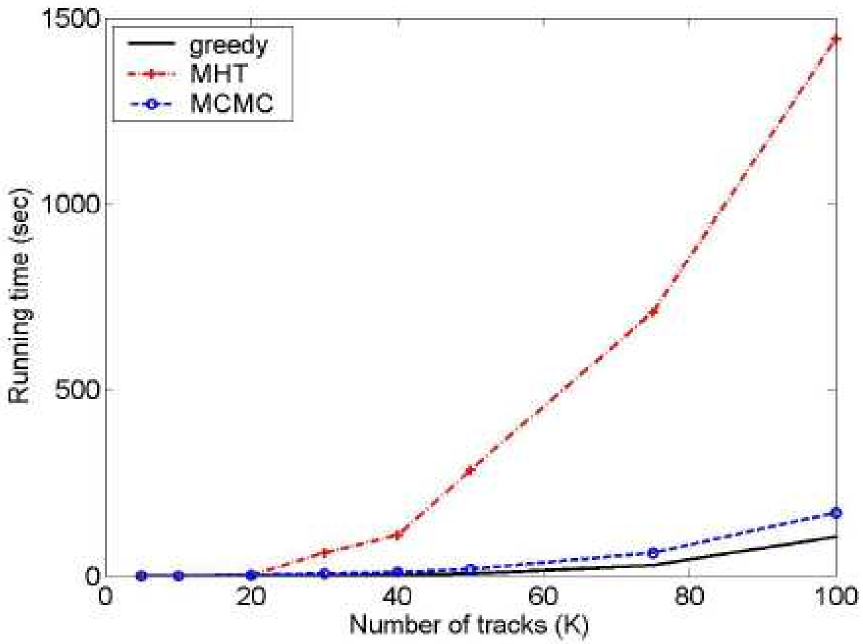
Aircraft Tracking Results

[Oh et al., CDC 2004]

(simulated data)



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



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

[Figures by Songhwai Oh]

Extending the Model: Air Bases

- Suppose aircraft don’t just enter and exit, but actually take off and land at bases
  - Want to track how many aircraft there are at each base
- Aircraft have destinations (particular bases) that they generally fly towards
- Assume set of bases is known

Extending the Model: Air Bases

```
#Aircraft(InitialBase = b) ~ InitialAircraftPerBasePrior();

CurBase(a, t)
    if t = 0 then = InitialBase(b)
    elseif TakesOff(a, Pred(t)) then = null
    elseif Lands(a, Pred(t)) then = Dest(a, Pred(t))
    else = CurBase(a, Pred(t));

InFlight(a, t) = (CurBase(a, t) = null);

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

Lands(a, t)
    if InFlight(a, t) then
        ~ LandingCPD(State(a, t), Location(Dest(a, t)));

Dest(a, t)
    if TakesOff(a, t) then ~ Uniform({Base b})
    elseif InFlight(a, t) then = Dest(a, Pred(t))

State(a, t)
    if TakesOff(a, Pred(t)) then
        ~ InitState(Location(CurBase(a, Pred(t))))
    elseif InFlight(a, t) then
        ~ StateTrans(State(a, Pred(t)), Location(Dest(a, t)));
```

Unknown Air Bases

- Just add two more lines:

```
#AirBase ~ NumBasesPrior();
Location(b) ~ BaseLocPrior();
```

BLOG Software

- Bayesian Logic inference engine available:
  http://www.cs.berkeley.edu/~milch/blog

Summary: Open Problems

- Inference
  - More widely applicable “lifted” inference
  - Approximation algorithms for problems with huge numbers of objects
  - Effective filtering algorithm for DBLOG
- Structure learning
  - Learning more complex dependency statements
  - Hypothesizing new random functions, new types

Syntax and semantics considered unnecessary

Caricature of a modern AI paper:

- define a probability model in English + LaTeX
- do some maths, get an efficient algorithm
- write 10,000 lines of code, get PhD

No need for any formal syntax or semantics, provided reader understands that the algorithm respects the intended meaning of the English + LaTeX

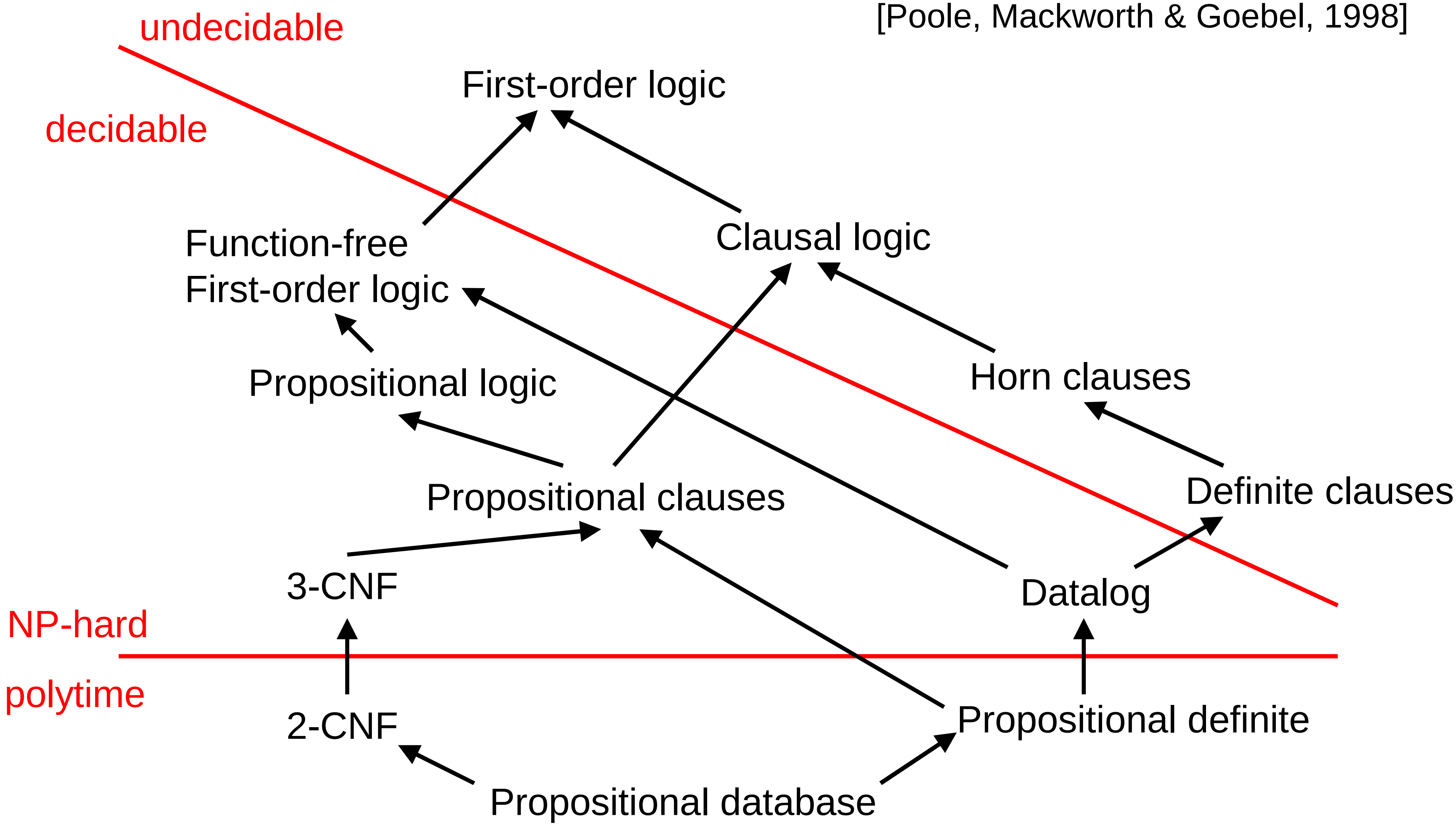
- write 5,000 lines + use BNT, get PhD faster

Syntax considered necessary

- Expressive notation increases scope of KR
  - (imagine English+LaTeX without Σ notation)
- Learning algorithms (esp. model selection) output syntactic representation of hypotheses
- Neural configurations and processing presumably implement a general domain-independent syntax and semantics (brains don’t do PhDs)

Expressiveness and complexity in logic

[Poole, Mackworth & Goebel, 1998]



What is the right syntax/semantics?

- No formal definitions for “good” syntax and semantics (but examples of “bad” can be convincing)
- Want concise, intuitive expressions for naturally occurring models

  => Need many experimental investigations
- Experience in programming languages suggests that decidability is not required