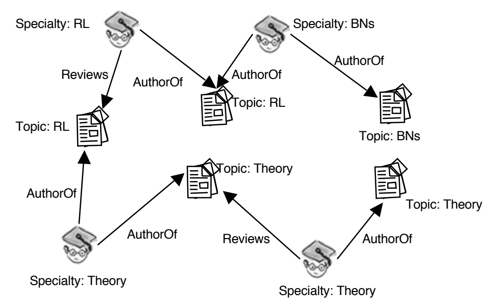
Relational Probability Models

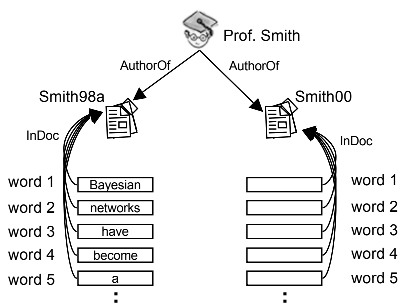
Brian Milch
MIT

IPAM Summer School
July 16, 2007

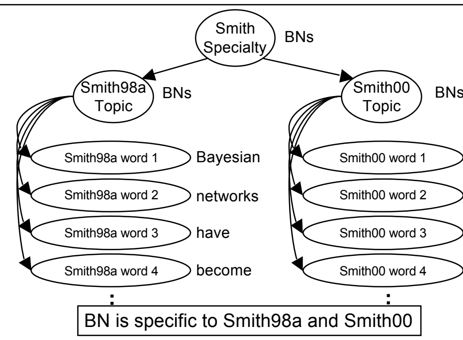
Objects, Attributes, Relations



Specific Scenario



Bayesian Network for This Scenario



Abstract Knowledge

- Humans have abstract knowledge that can be applied to any individuals
- How can such knowledge be:
  - Represented?
  - Learned?
  - Used in reasoning?

Outline

- Relational probability models (RPMs)

  Abstract probabilistic model for attributes + relational skeleton: objects & relations → graphical model

  - Representation
  - Inference
  - Learning
- Relational uncertainty: extending RPMs with probabilistic models for relations

Representation

- Have to represent
  - Set of variables
  - Dependencies
  - Conditional probability distributions (CPDs)

  All depend on relational skeleton
- Many proposed languages
- We'll use Bayesian logic (BLOG) [Milch *et al.* 2005]

7

Typed First-Order Logic

- Objects divided into types
  Researcher, Paper, WordPos, Word, Topic
- Express attributes and relations with functions and predicates
  FirstAuthor(paper) → Researcher (non-random)
  Specialty(researcher) → Topic (random)
  Topic(paper) → Topic (random)
  Doc(wordpos) → Paper (non-random)
  WordAt(wordpos) → Word (random)

8

Set of Random Variables

- For random functions, have variable for each tuple of argument objects

Researcher: Smith, Jones   Paper: Smith98a, Smith00, Jones00
WordPos: Smith98a_1, …, Smith98a_3212, Smith00_1, etc.

Specialty: Specialty(Smith) Specialty(Jones)
Topic: Topic(Smith98a) Topic(Smith00) Topic(Jones00)
WordAt: WordAt(Smith98a_1) … WordAt(Smith98a_3212)
WordAt(Smith00_1) … WordAt(Smith00_2774)
WordAt(Jones00_1) … WordAt(Jones00_4893)

9

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)));
```

Logical term identifying parent node

```
                             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)));
```

10

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)));
```

11

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

12

Aggregation

- Aggregate distributions

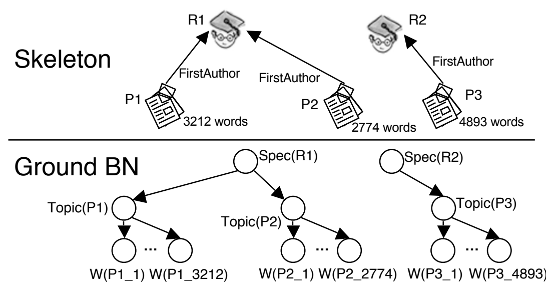
```
                              multiset defined by formula
Topic(p) ~ TopicAggCPD({Specialty(r) for Researcher r :
                                        AuthorOf(r, p)});
```

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

- Aggregate values

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

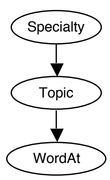

Semantics: Ground BN



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

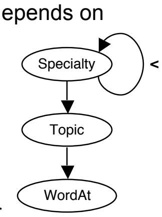


Acyclic Relations

[Friedman et al., IJCAI 1999]

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

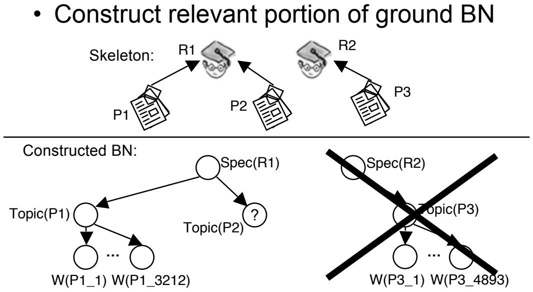
```
Specialty(r)
  if Advisor(r) != null then
     ~ SpecCPD(Specialty(Advisor(r)))
  else ~ SpecialtyPrior();
```



- Symbol graph has self-loop!
- Require certain nonrandom functions to be acyclic: F(***x***) < ***x*** under some partial order
- Label edges with "<" and "=" signs; get stronger acyclicity theorem

Inference: Knowledge-Based Model Construction (KBMC)

- Construct relevant portion of ground BN



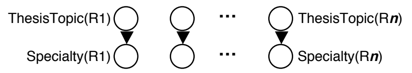
[Breese 1992; Ngo & Haddawy 1997]

Inference on Constructed Network

- Run standard BN inference algorithm
  - Exact: variable elimination/junction tree
  - Approx: Gibbs sampling, loopy belief propagation
- Exploit some repeated structure with lifted inference [Pfeffer *et al.*, UAI 1999; Poole, IJCAI 2003; de Salvo Braz *et al.*, IJCAI 2005]

Lifted Inference

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



- Could sum out ThesisTopic(R) nodes one by one
- 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***)

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

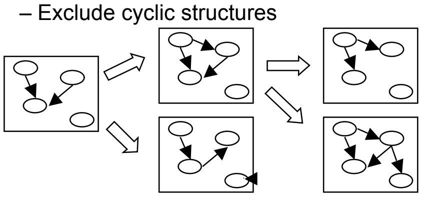
Learning

- 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 task: learn dependency structure

Structure Learning for BNs

- Find BN structure M that maximizes

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



Logical Structure Learning

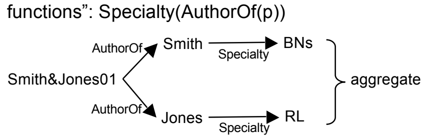
- In RPM, want logical specification of each node's parent set
- Deterministic analogue: inductive logic programming (ILP)
  [Dzeroski & Lavrac 2001; Flach and Lavrac 2002]
- Classic work on RPMs by Friedman, Getoor, Koller & Pfeffer [1999]
  - We'll call their models FGKP models
    (they call them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

FGKP Models

- Each dependency statement has form:

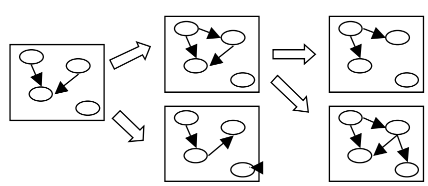
  `Func(x) ~ TabularCPD[...](s1,..., sk)`

  where $s_1,\ldots,s_k$ 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

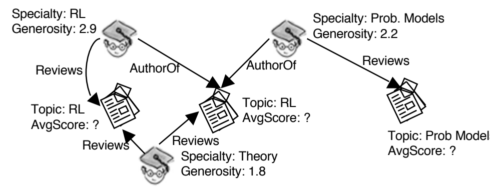
- Relational probability models (RPMs)

  Abstract probabilistic model for attributes + relational skeleton: objects & relations → graphical model

  - Representation
  - Inference
  - Learning
- Relational uncertainty: extending RPMs with probabilistic models for relations

25

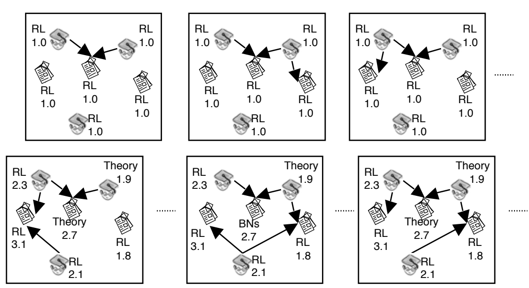
Relational Uncertainty: Example



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

26

Possible Worlds



27

Simplest Approach to Relational Uncertainty

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

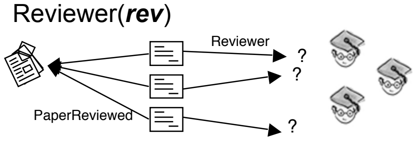
```
Reviews(r, p) ~ ReviewCPD(Specialty(r), Topic(p));
```

- 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] 28

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] 29

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]

30

Examples of 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

31

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

32

Semantics: Ground BN

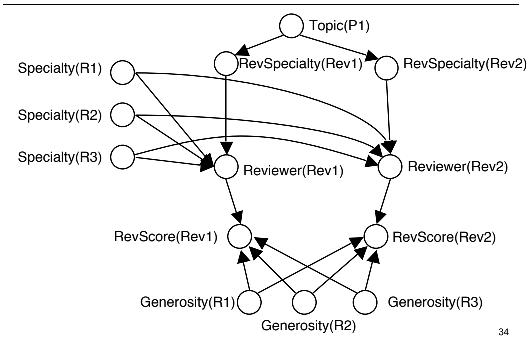
- Can still define ground BN
- Parents of node ***X*** are all basic RVs whose values are potentially relevant in evaluating the right hand side of ***X***'s dependency statement
- Example: for RevScore(Rev1)…

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

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

33

Ground BN



34

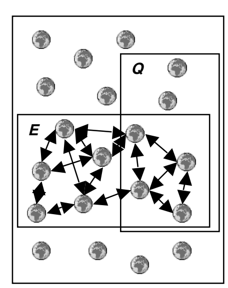
Inference

- 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

35

Markov Chain Monte Carlo (MCMC)

- Markov chain ω_1, ω_2, ... over worlds in ***E***
- Designed so unique stationary distribution is proportional to $\boldsymbol{p}(\omega)$
- Fraction of $\omega_1, \omega_2, \ldots, \omega_{\boldsymbol{N}}$ in ***Q*** converges to $P(\boldsymbol{Q}|\boldsymbol{E})$ as $\boldsymbol{N} \to \infty$



36

Metropolis-Hastings MCMC

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

    [illegible]

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

Computing Acceptance Ratio Efficiently

- World probability is

  [illegible]

  where $\mathbf{pa}_{\omega}(\boldsymbol{X})$ is instantiation 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]

Summary

- Human knowledge is more abstract than basic graphical models
- Relational probability models
  - Logic-based representation
  - Structure learning by search over slot chains
  - Inference by KBMC
- Relational uncertainty
  - Natural extension to logic-based representation
  - Approximate inference by MCMC

References

- Wellman, M. P., Breese, J. S., and Goldman, R. P. (1992) "From knowledge bases to decision models". *Knowledge Engineering Review* 7:35-53.
- Breese, J.S. (1992) "Construction of belief and decision networks". *Computational Intelligence* 8(4):624-647.
- Ngo, L. and Haddawy, P. (1997) "Answering queries from context-sensitive probabilistic knowledge bases". *Theoretical Computer Sci.* 171(1-2):147-177.
- Koller, D. and Pfeffer, A. (1998) "Probabilistic frame-based systems". In *Proc. 15th AAAI National Conf. on AI*, pages 580-587.
- 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.
- Pfeffer, A., Koller, D., Milch, B., and Takusagawa, K. T. (1999) "SPOOK: A System for Probabilistic Object-Oriented Knowledge". In *Proc. 15th Conf. on Uncertainty in AI*, pages 541-550.
- 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.

References

- Poole, D. (2003) "First-order probabilistic inference". In *Proc. 18th Int'l Joint Conf. on AI*, pages 985-991.
- de Salvo Braz, R. and Amir, E. and Roth, D. (2005) "Lifted first-order probabilistic inference." In *Proc. 19th Int'l Joint Conf. on AI*, pages 1319-1325.
- 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.