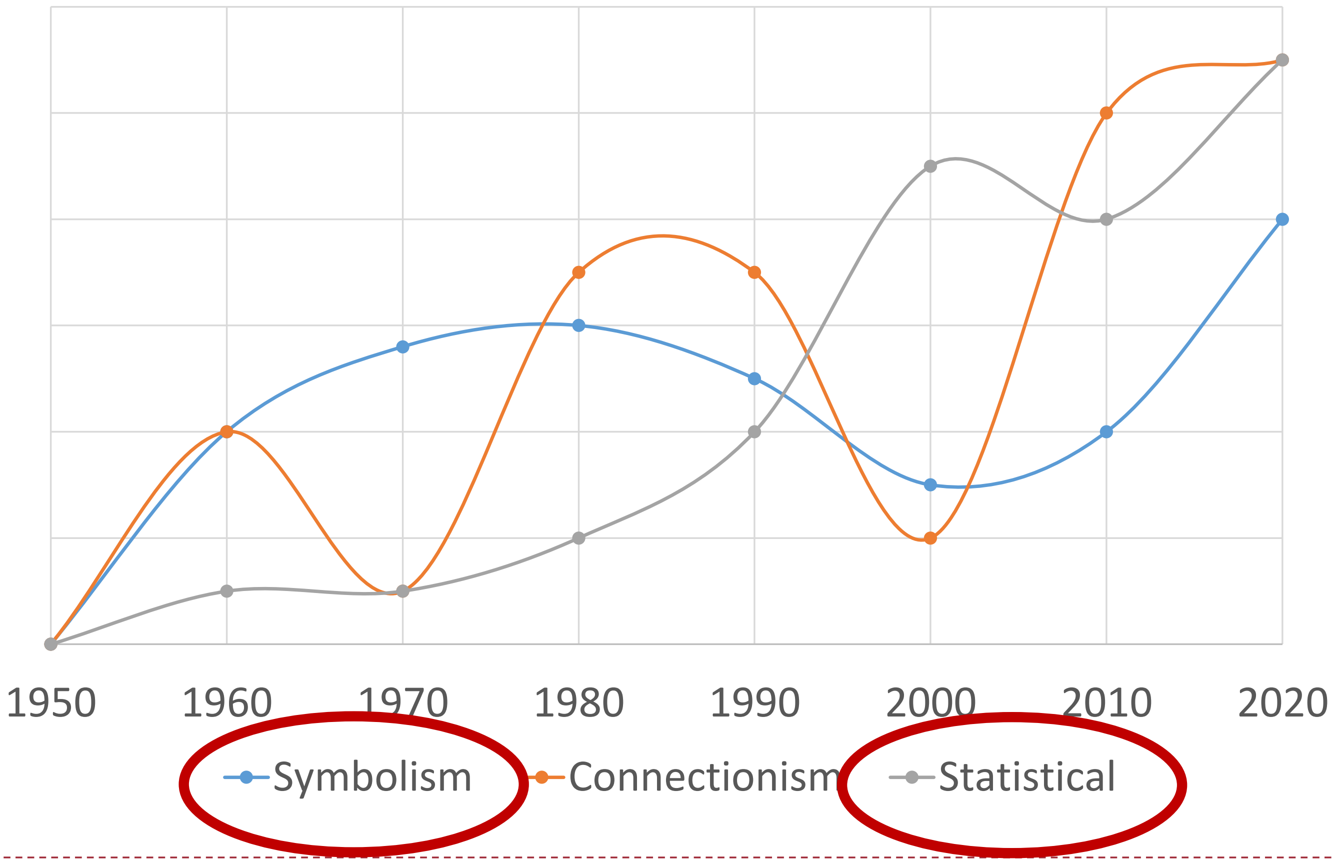
Three types of (strong) AI approaches



Probabilistic Logics

AIMA 14.6
Additional materials

[Adapted from slides by Lise Getoor at UC Santa Cruz and Pedro Domingos at U Washington]

Additional reference materials

- L. Getoor and B. Taskar (eds.), Introduction to Statistical Relational Learning, 2007. Cambridge, MA: MIT Press.
  - Ch 5: Probabilistic Relational Models
  - Ch 12: Markov Logic

Logics vs. Probabilistic Models

- Symbolic logics
  - FOL is very expressive
    - relations between objects, quantifiers
  - But it cannot model uncertainty

mix

- Probabilistic Models
  - BN/MN model uncertainty in a concise manner
  - But limited in expressiveness
    - BN/MN is essentially propositional

Probabilistic Logics

- Goal
  - Combine (subsets of) logic and probability into a single language
- A.k.a. Statistical Relational Learning

- Lots of approaches. We will cover two of them:
  - Probabilistic Relational Models PRM
  - Markov Logic

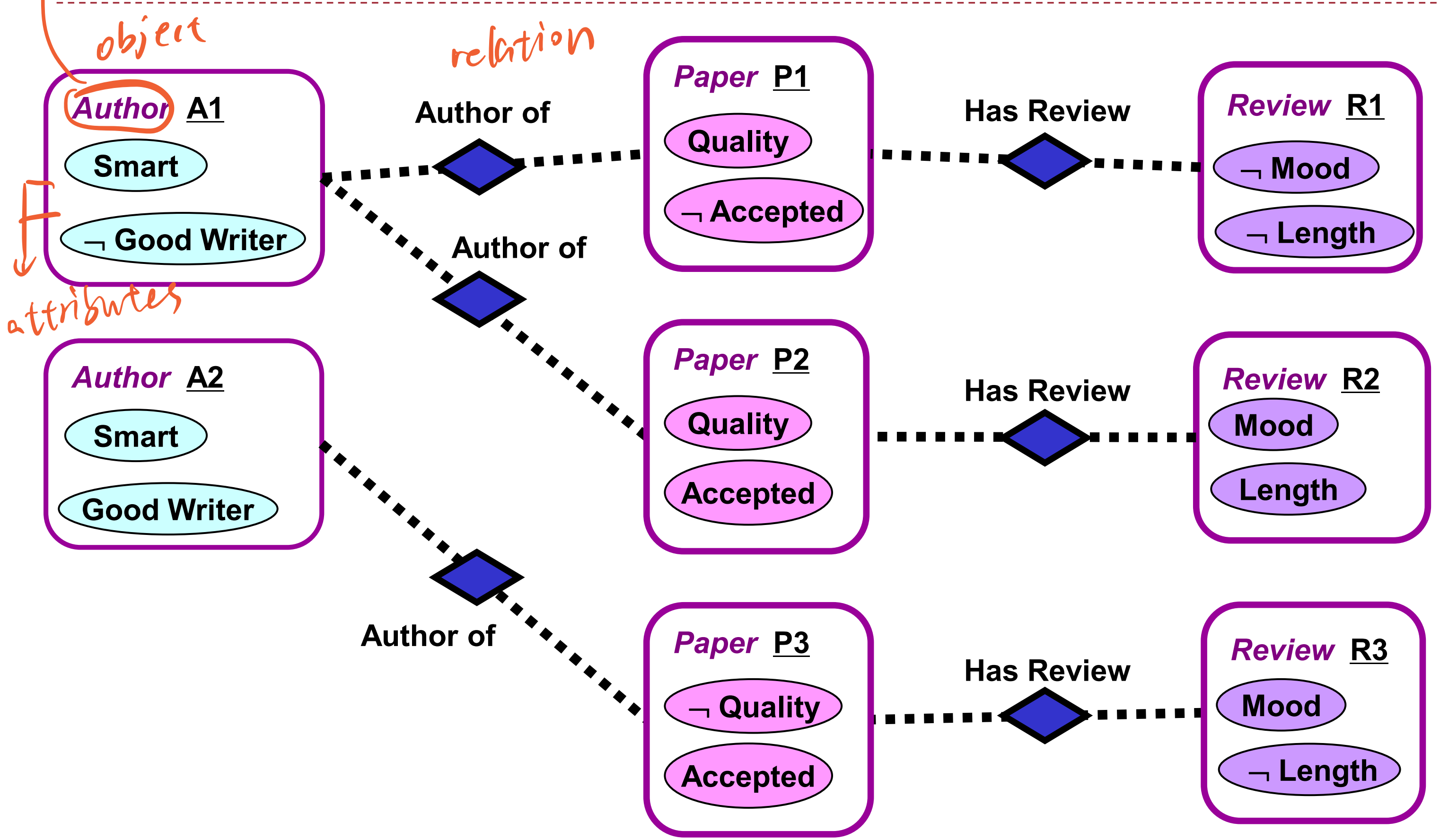
Probabilistic Relational Models

Probabilistic Relational Models

- Logical language
  - Frame (typed relational knowledge)
    - A subclass of FOL
- Probabilistic language
  - Bayes nets

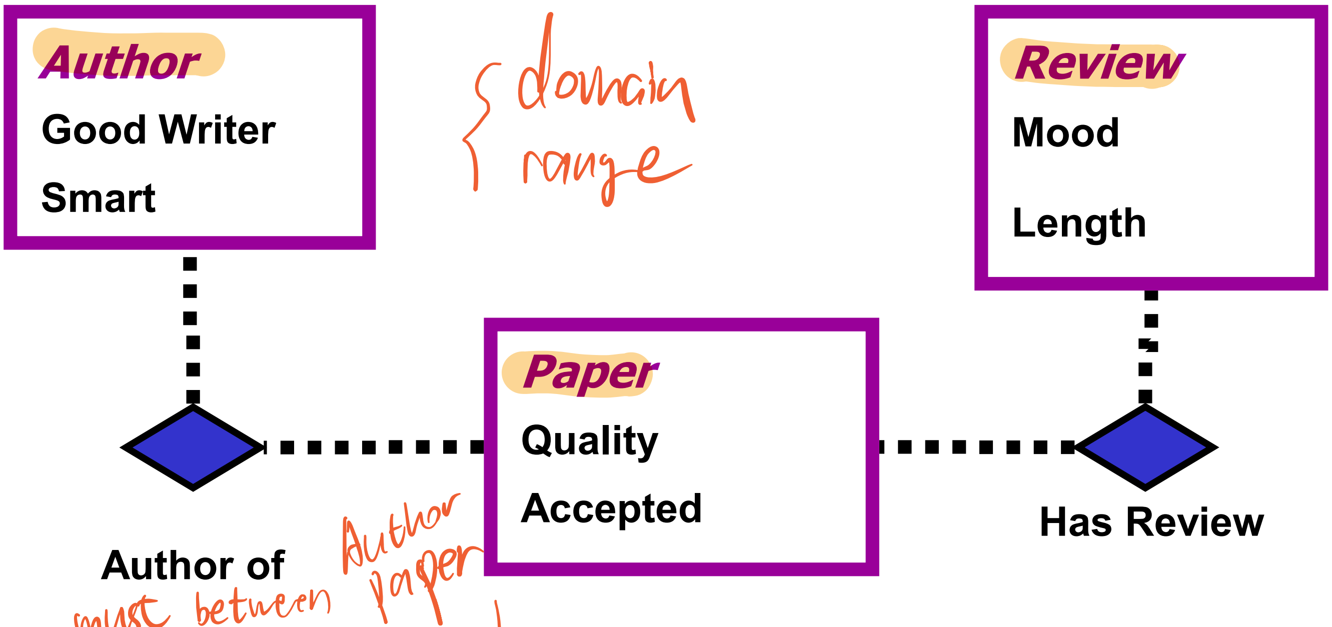
Subclass of FOL

Typed relational knowledge



Typed relational knowledge

Why is this a subclass of FOL?

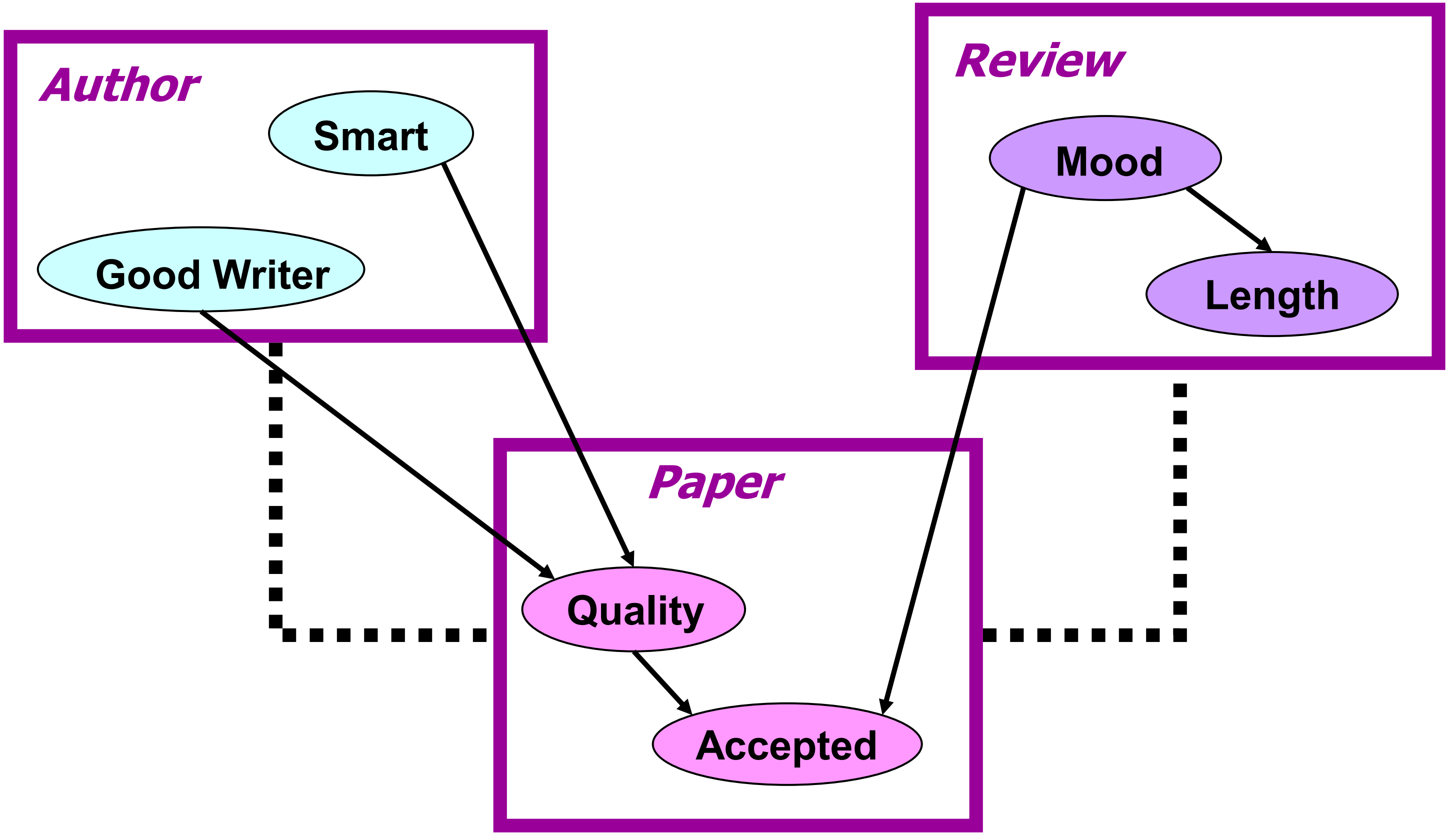


Ontology / Schema

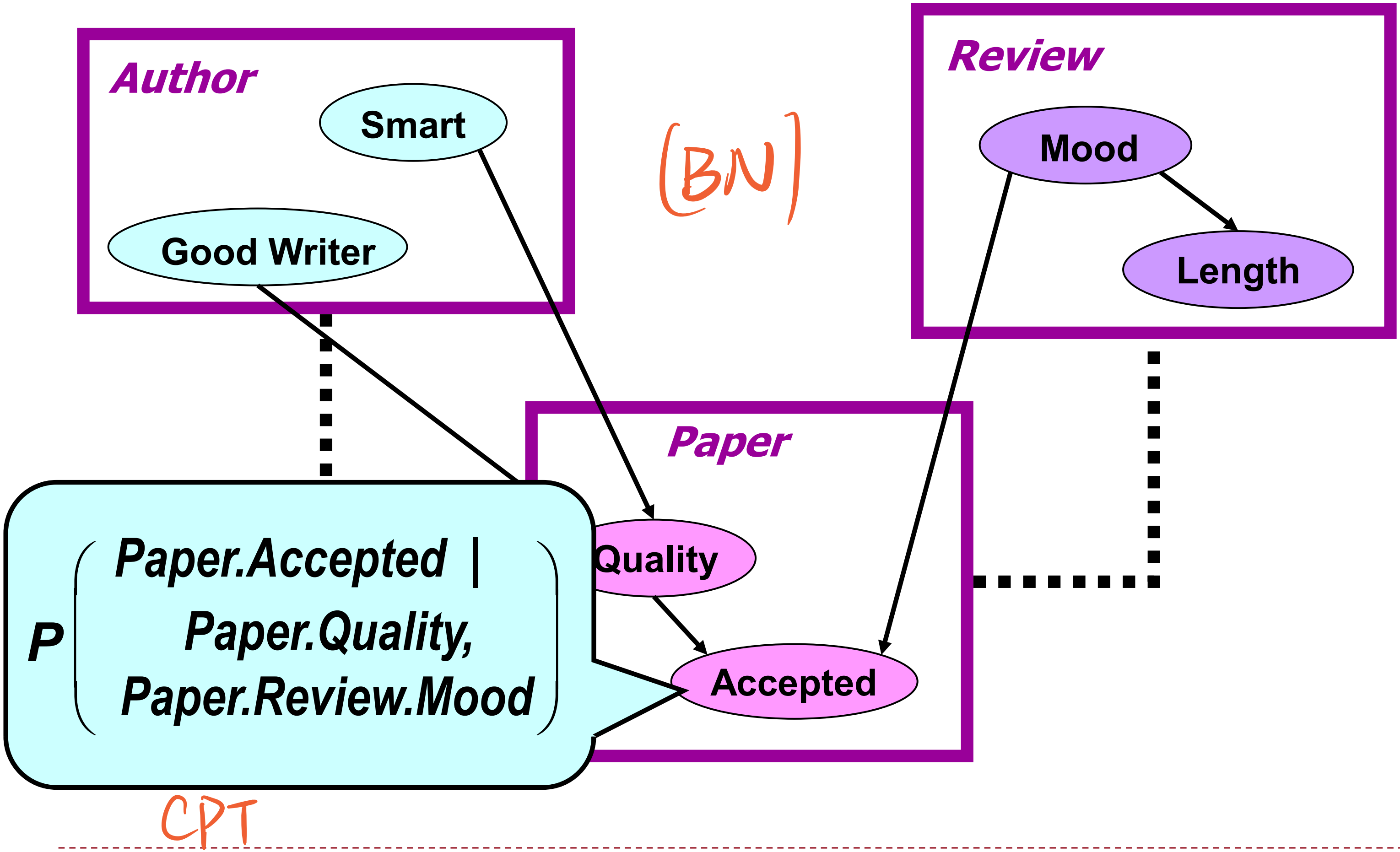
- The types of objects and their valid relations and object's attributes

Goal: To build relation between attributes?

Probabilistic Relational Model



Probabilistic Relational Model



Schema level

scheme level:

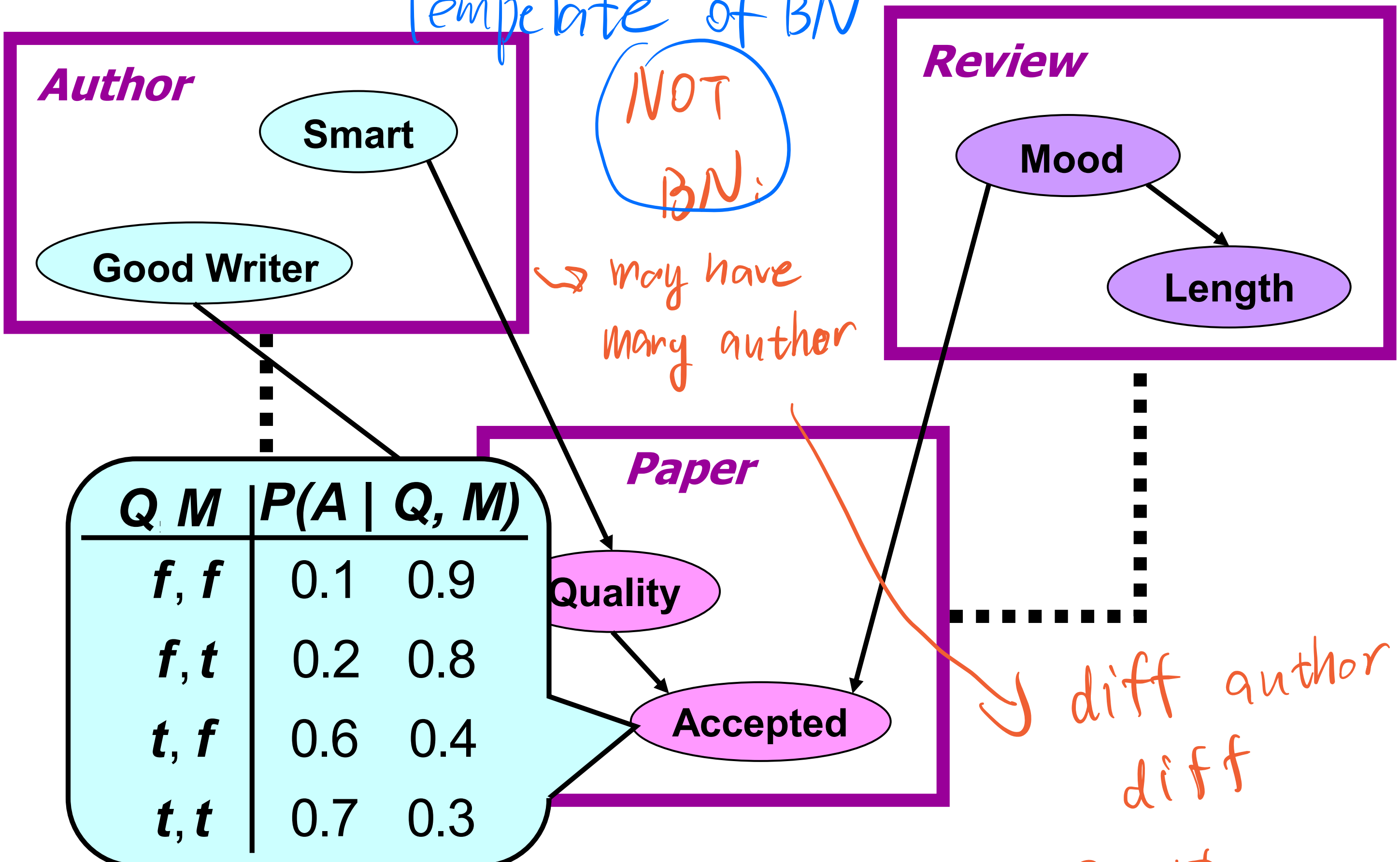
模板

Probabilistic Relational Model

Template of BN

NOT BN;

may have many author



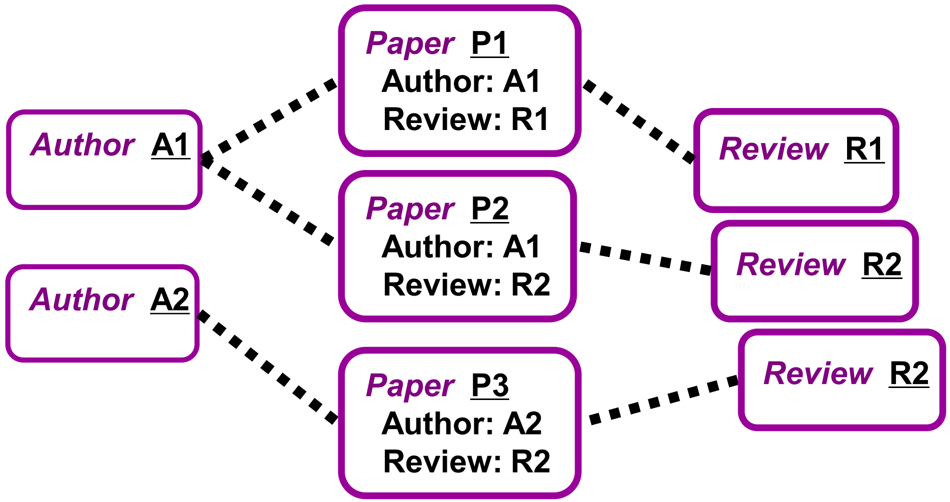
Q M	P(A | Q, M)	
f, f	0.1	0.9
f, t	0.2	0.8
t, f	0.6	0.4
t, t	0.7	0.3

diff author
diff
smart

=> Many variables

smart 1, smart 2, ...

Relational Skeleton

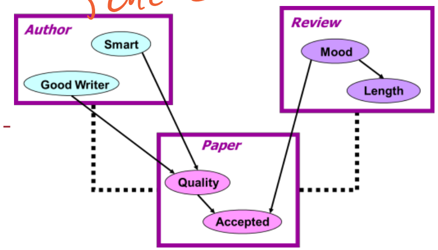


Fixed relational skeleton σ:

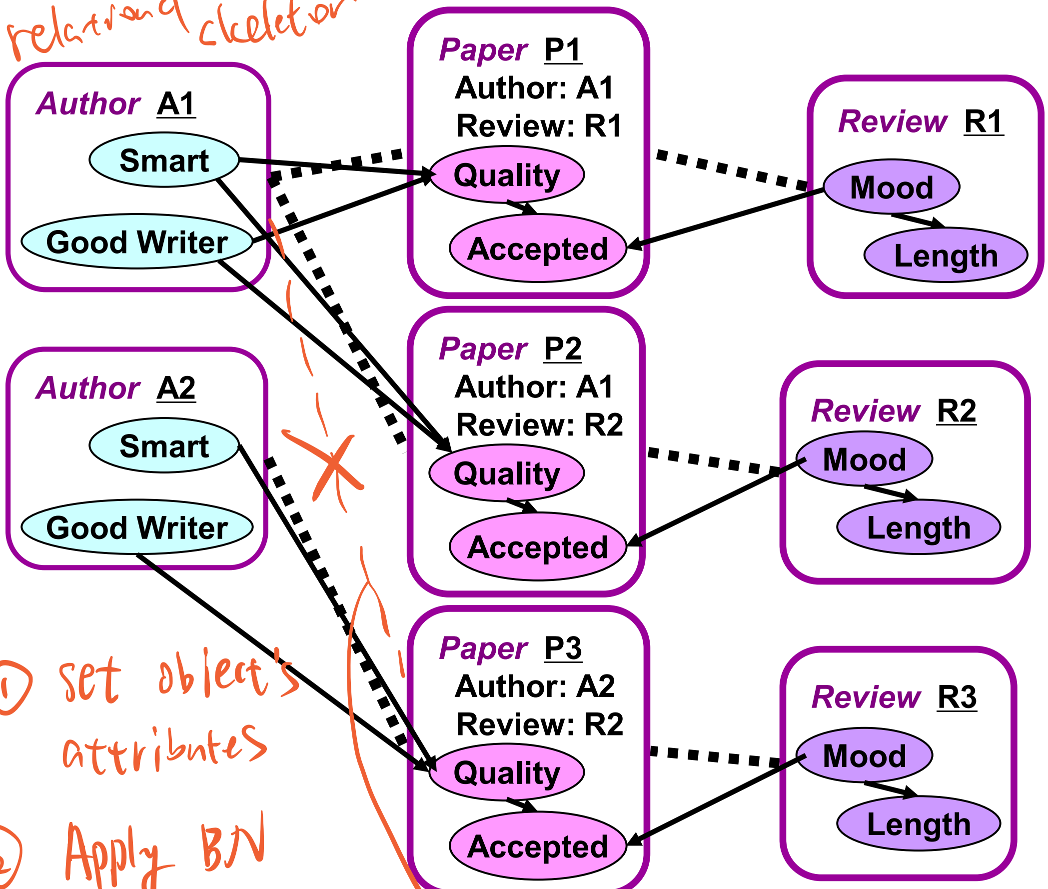
- set of objects in each class
- relations between them
- attribute values unknown

PRM with Attribute Uncertainty

scheme



relational skeleton



PRM unrolled wrt. the relational skeleton produces a BN that models the distribution over instantiations of attributes.

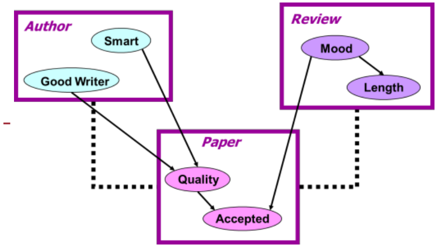
① set object's attributes

② Apply BN in scheme

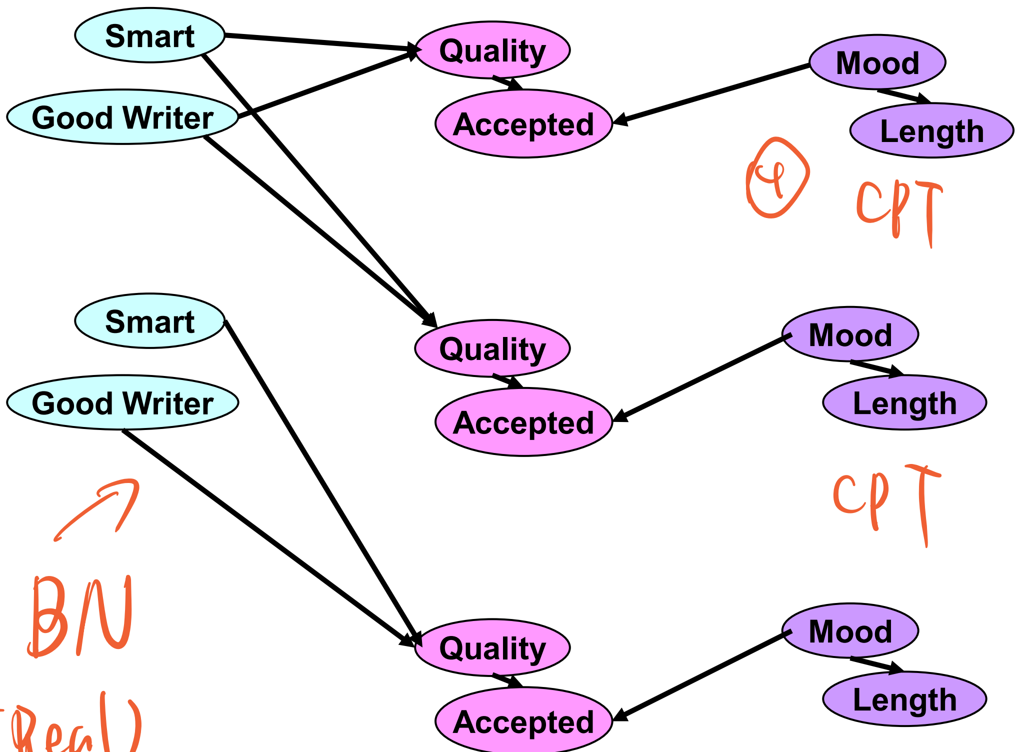
No relation

No BN

PRM with Attribute Uncertainty



③ remove objects and relations



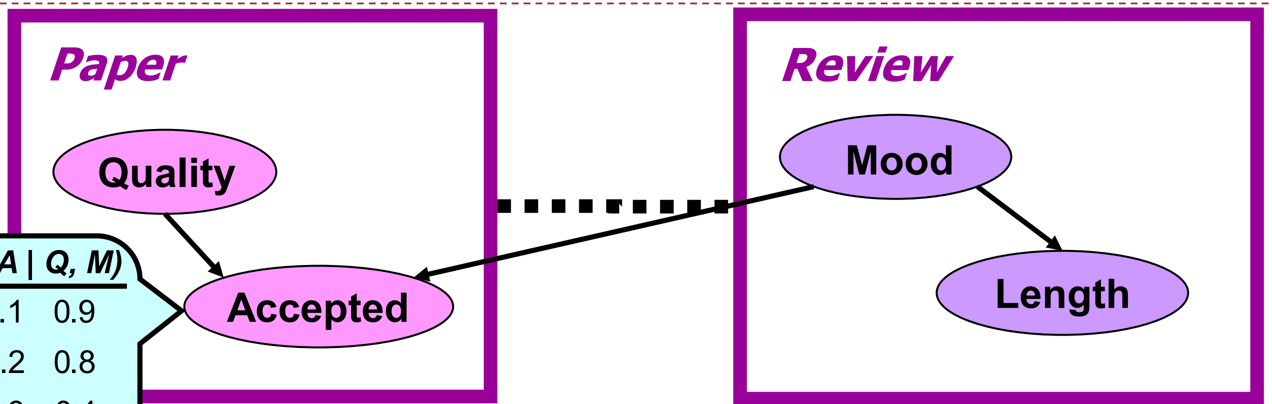
④ CPT

CPT

BN (Real)

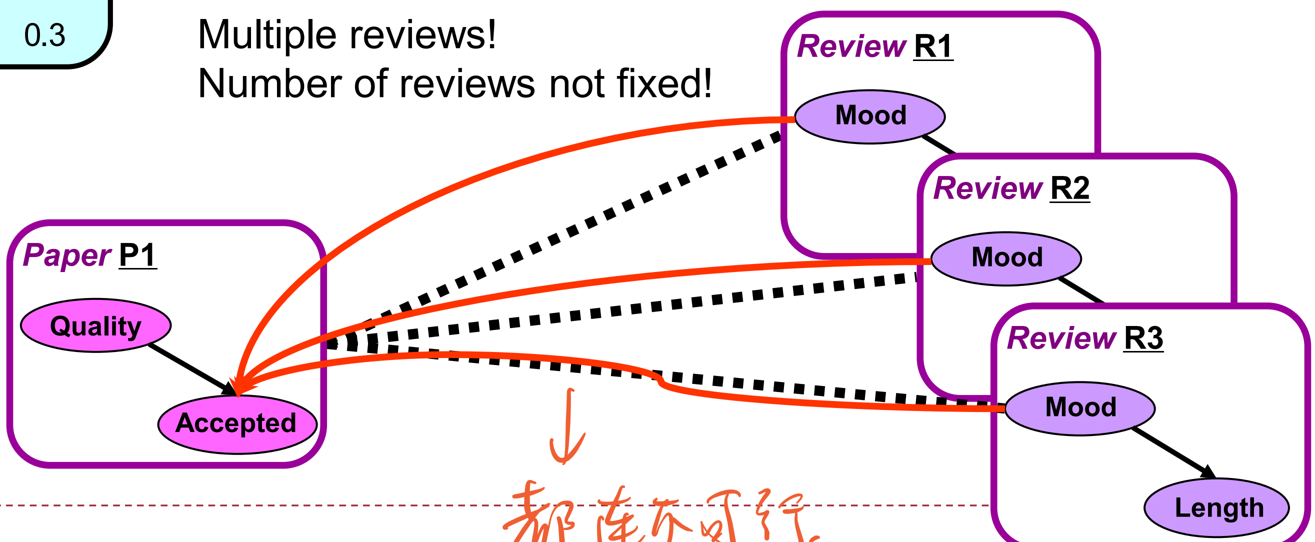
PRM unrolled wrt. the relational skeleton produces a BN that models the distribution over instantiations of attributes.

PRM: Aggregate Dependencies



Q, M	P(A | Q, M)	
f, f	0.1	0.9
f, t	0.2	0.8
t, f	0.6	0.4
t, t	0.7	0.3

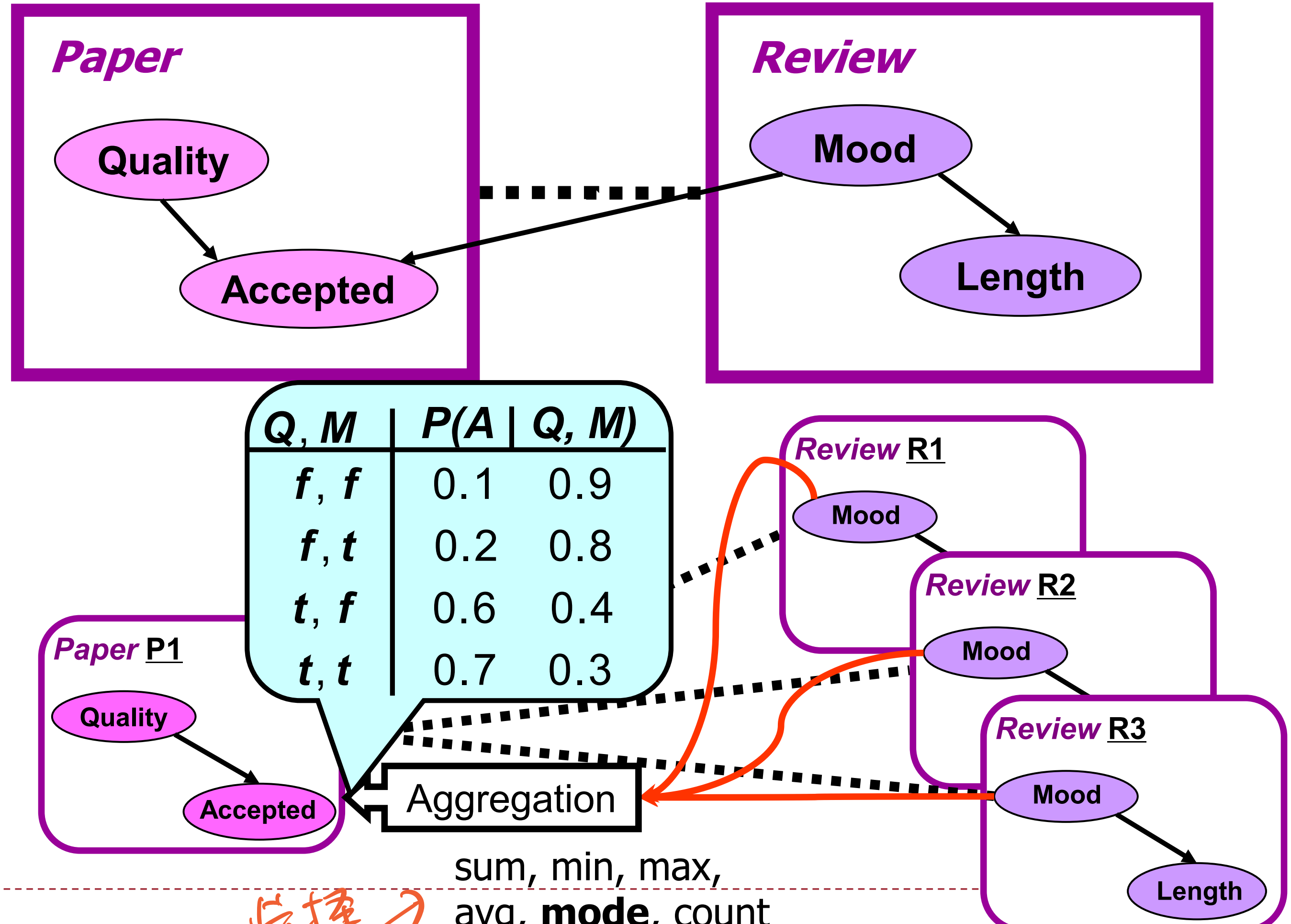
Multiple reviews!
Number of reviews not fixed!



都连不可行.

only 1 CPT, only assume 2 parent

PRM: Aggregate Dependencies



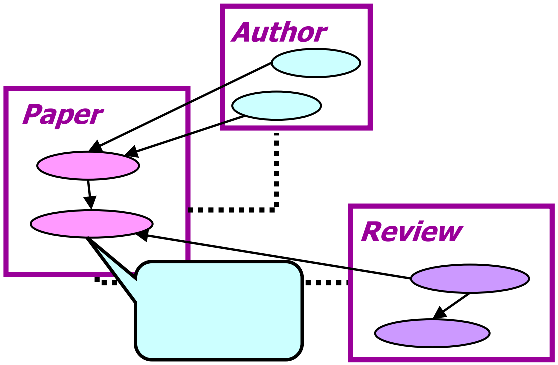
Q, M	P(A | Q, M)	
f, f	0.1	0.9
f, t	0.2	0.8
t, f	0.6	0.4
t, t	0.7	0.3

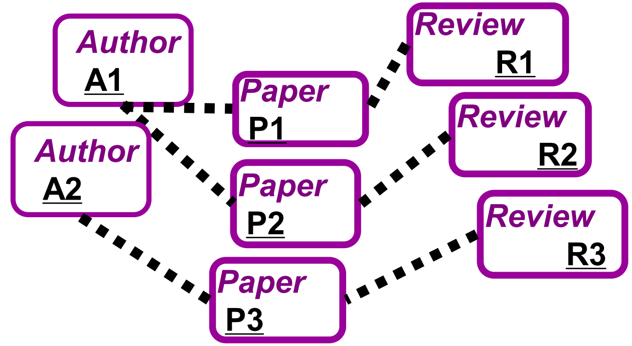
投票, majority voting

keep accepted | parent

eg. ≥2 in 3 → win

PRM with Attribute Uncertainty





结构 scheme -

PRM (S,Θ) + relational skeleton (σ) =

prob relational mod

probability distribution over instantiations of attributes I:

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

BN

Objects Attributes

条件

type

if object equal

Structural Uncertainty

not known objects relations

- PRM with AU only well-defined when the relational skeleton is known
- What if we are uncertain about the relational structure?
  - How many objects does an object relate to?
  - Which object is an object related to?
  - Does an object actually exist?
  - Are two objects identical?

equal

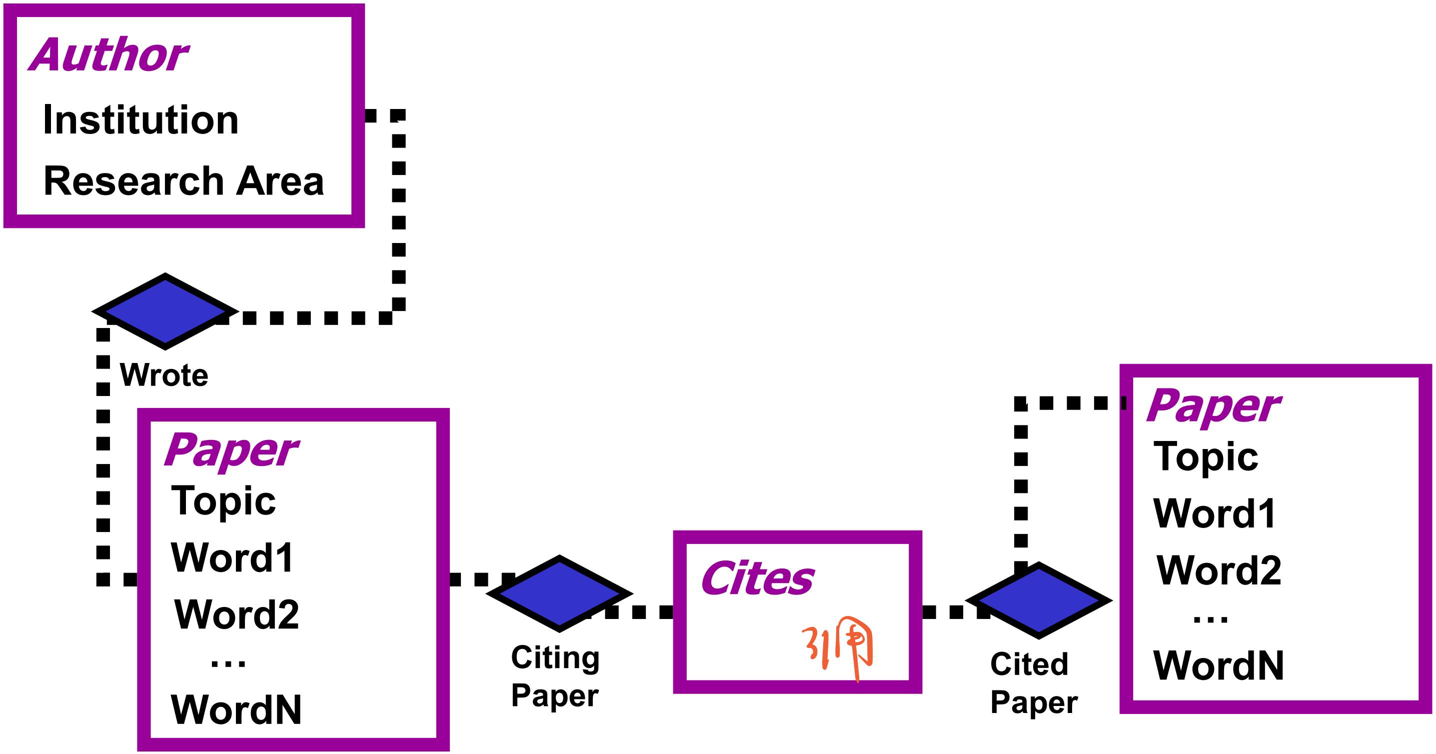
Structural Uncertainty

- Need probabilistic models that capture structural uncertainty
- Types of SU:
  - Existence uncertainty
  - Reference uncertainty
  - Number uncertainty
  - Type uncertainty
  - Identity uncertainty

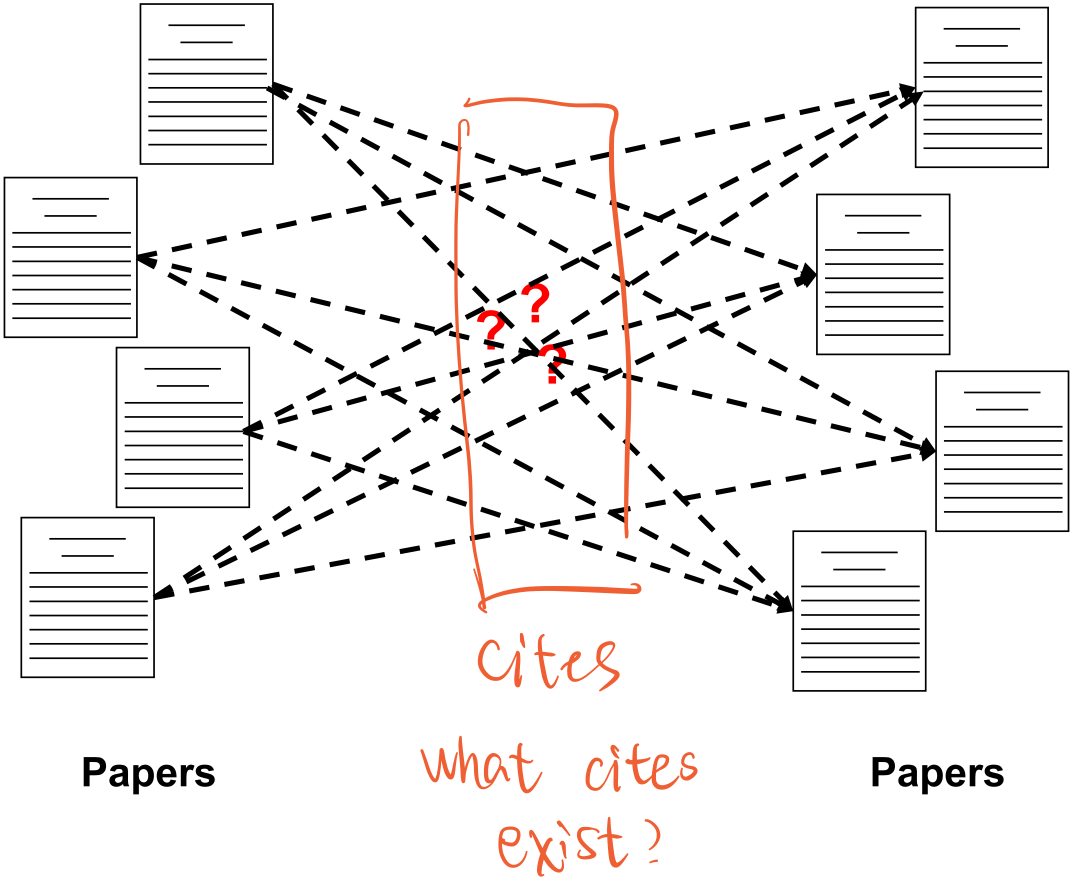
Today

If object exist?

Citation Schema

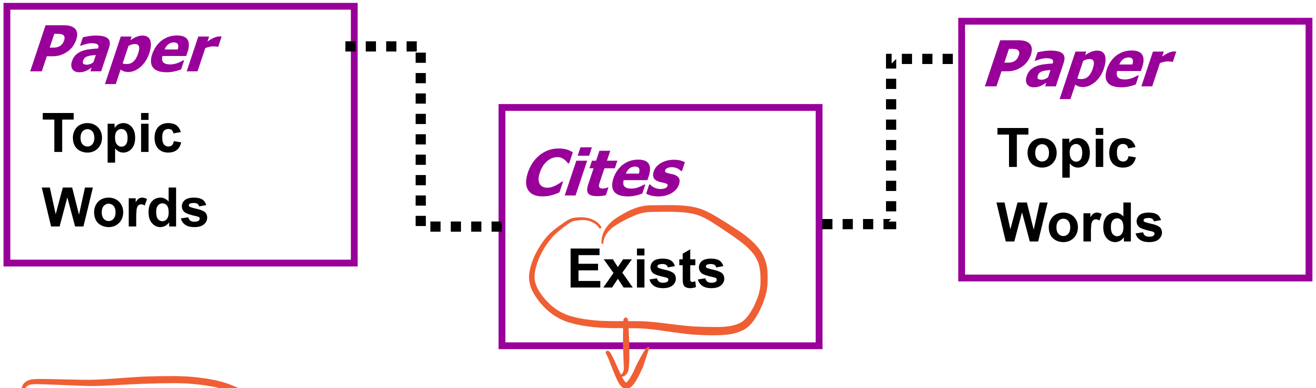


Existence Uncertainty



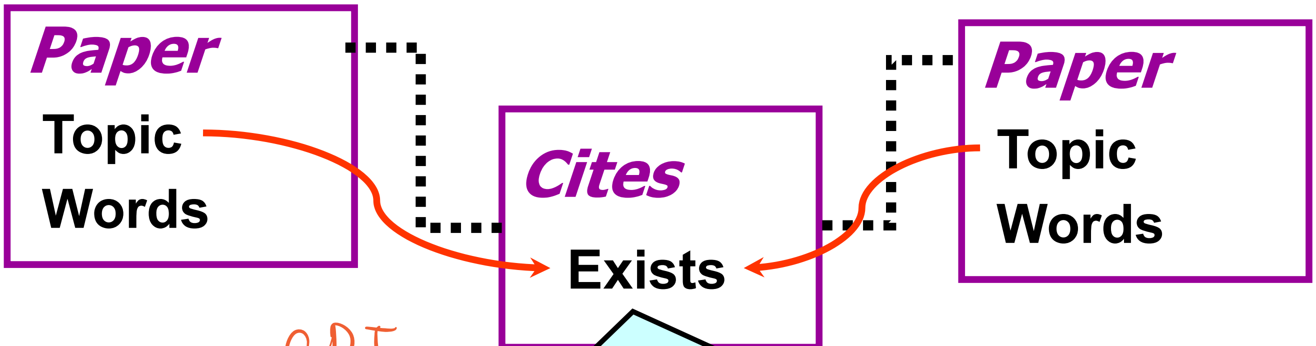
PRM with Existence Uncertainty

Scheme :



Introduce the **Exists** attribute for ***Cites***

PRM with Existence Uncertainty

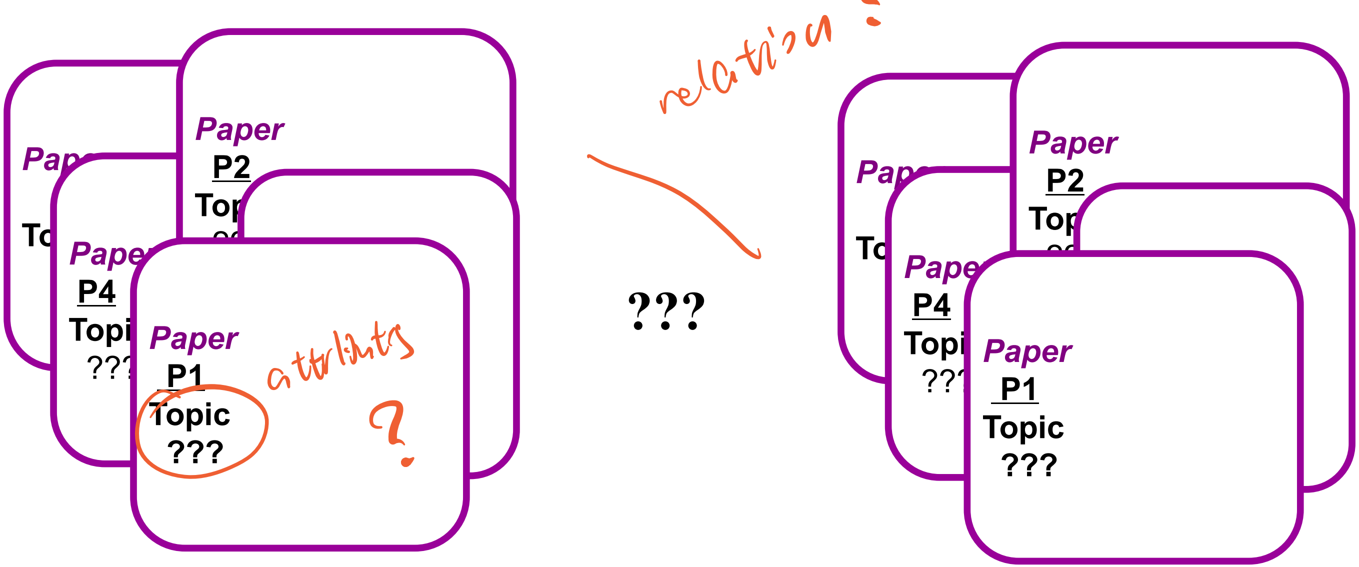


CPT

Citer.Topic	Cited.Topic	*False*	*True*
Theory	*Theory*	0.995	0.005
Theory	*AI*	0.999	0.001
AI	*Theory*	0.997	0.003
AI	*AI*	0.992	0.008

Since no relation

Object skeleton

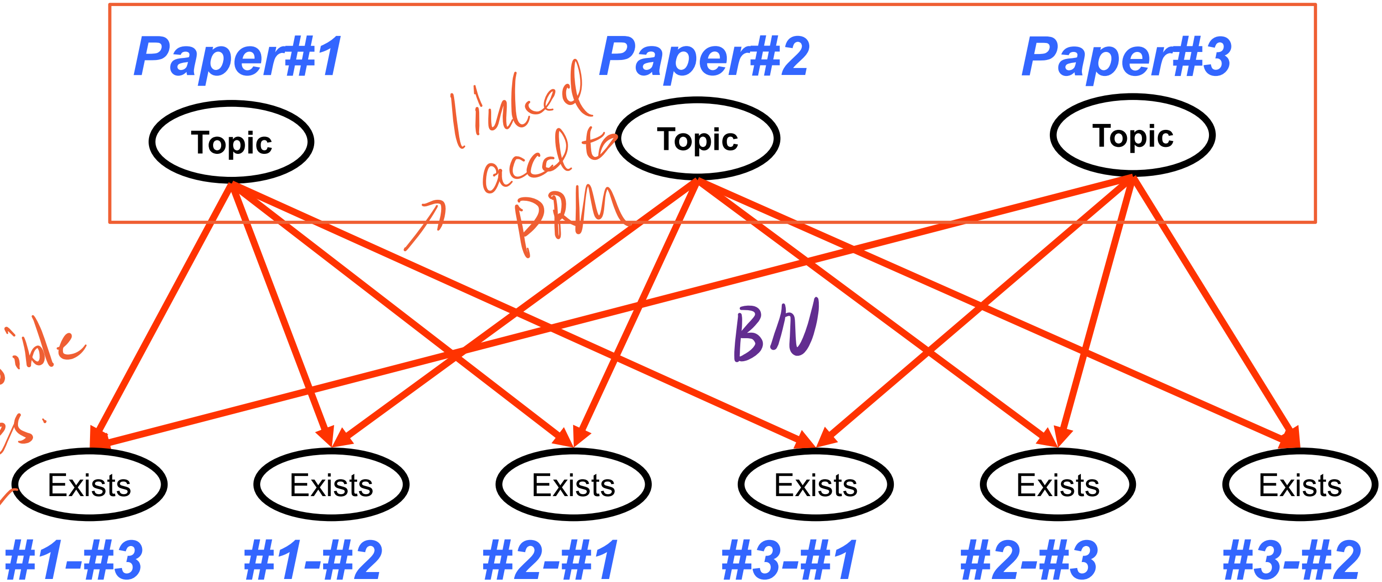


Fixed object skeleton σ:

- set of objects in each class
- unknown relations between them
- unknown attribute values

PRM with Existence Uncertainty

object skeleton.



linked accd to PRM

BN

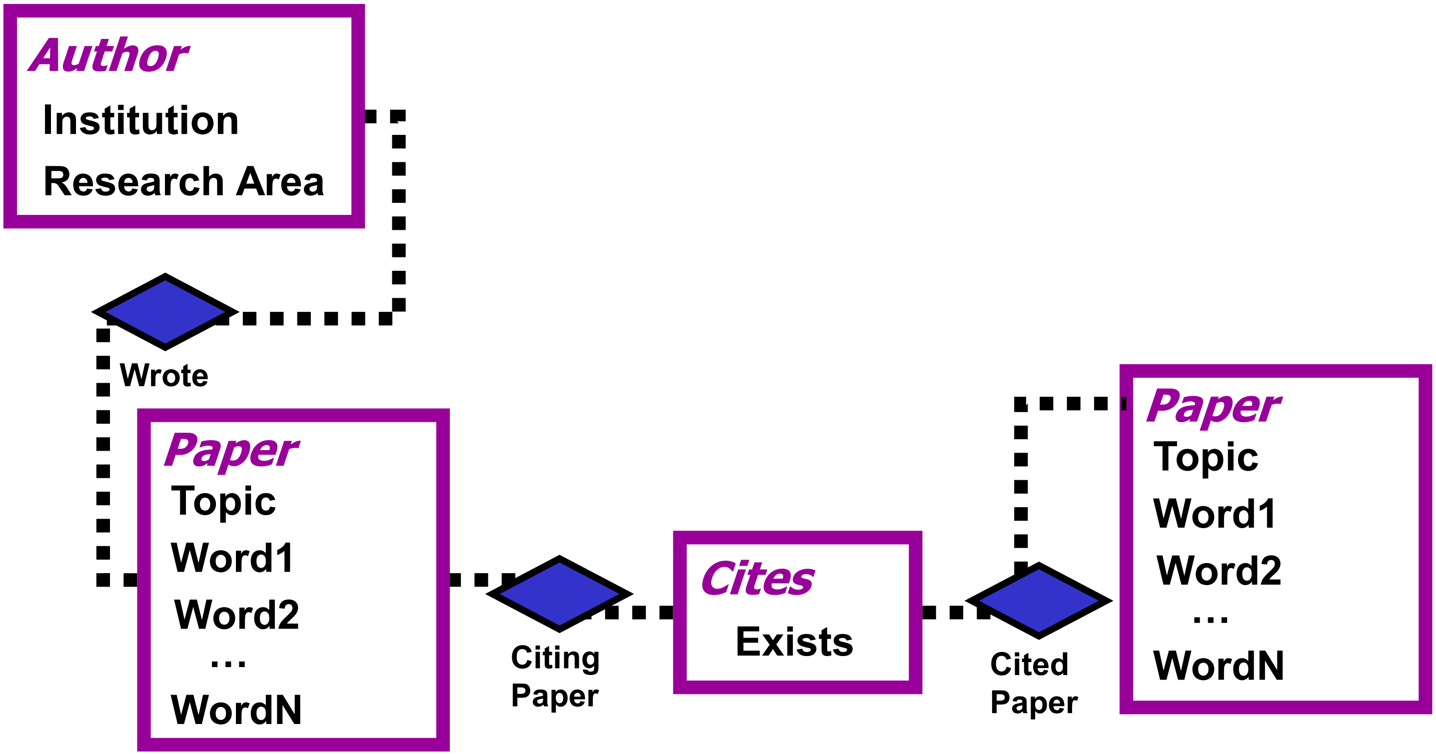
possible cites.

attributes

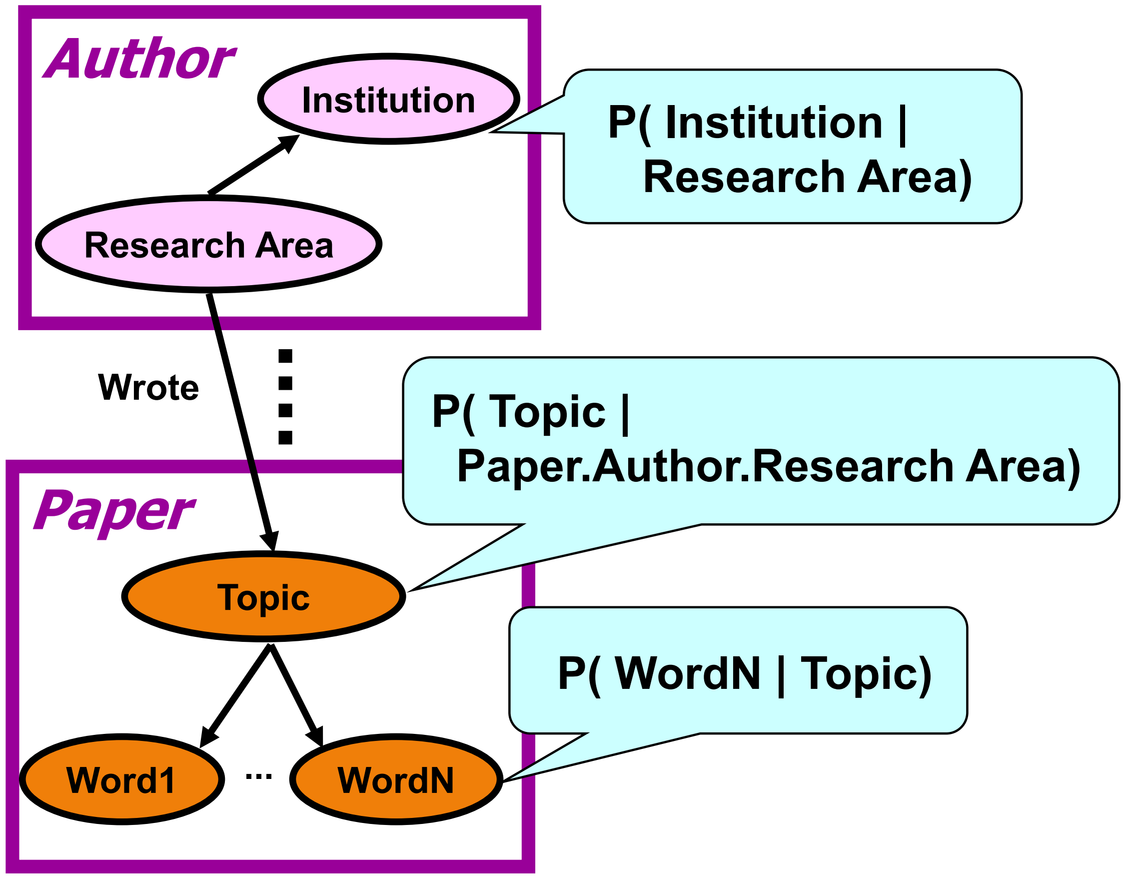
PRM w/ EU unrolled wrt. the object skeleton produces a BN

A more complicated example

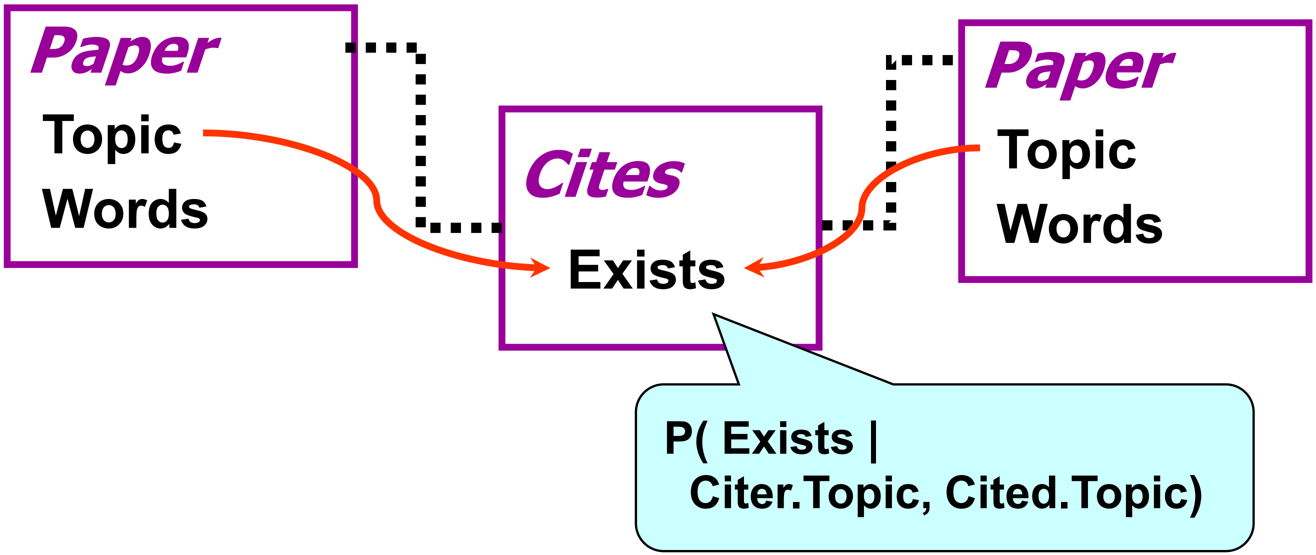
scheme:



PRM with Attribute Uncertainty

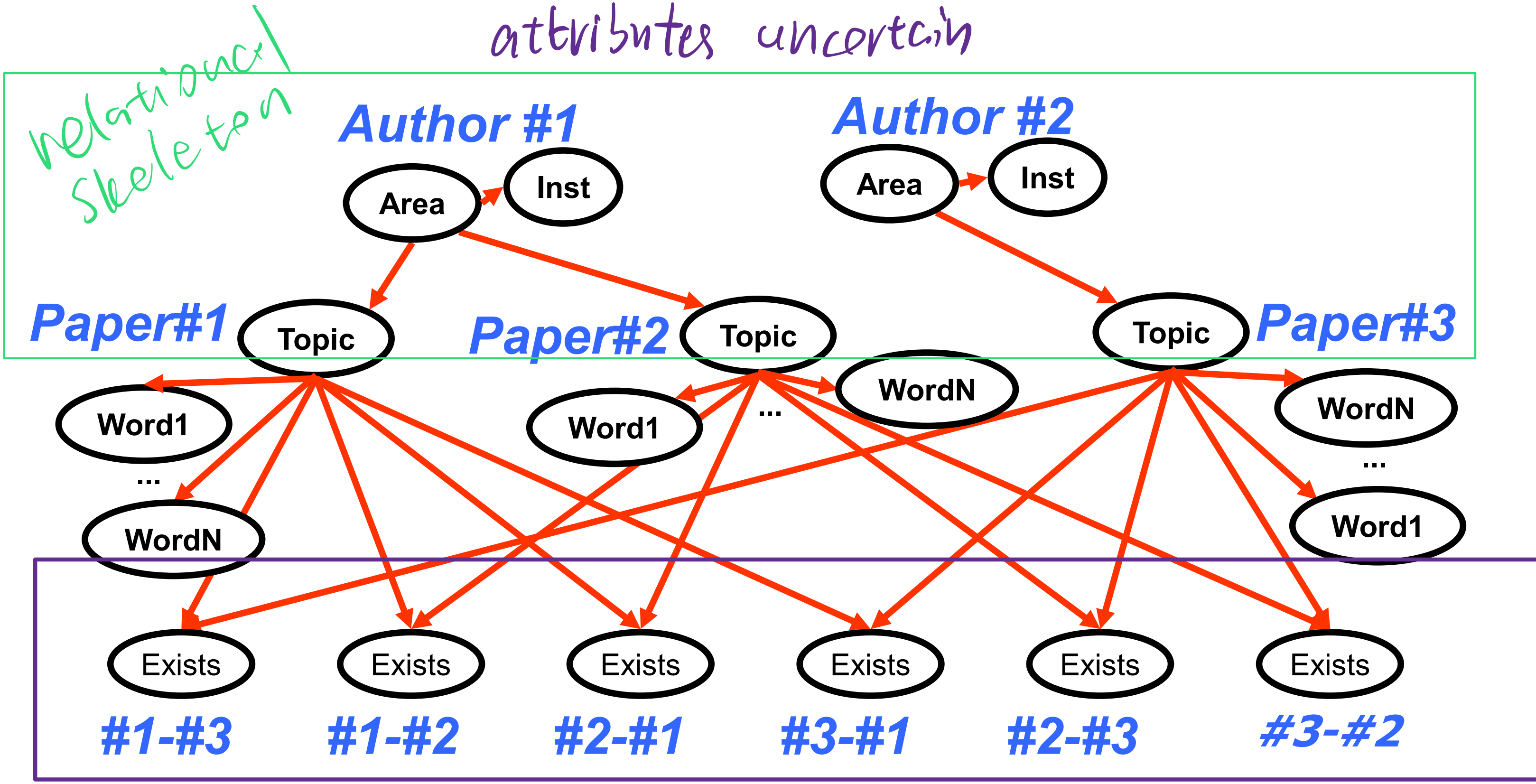


PRM with Existence Uncertainty



PRM with Existence Uncertainty

attributes uncertain



object skeleton

existence uncertain

Markov Logic

Markov Logic

- Logical language
  - First-order logic
- Probabilistic language
  - Markov networks

Review: Markov networks

- A Markov network (or Markov random field) encodes a joint distribution with an undirected graph

无向!
no direction

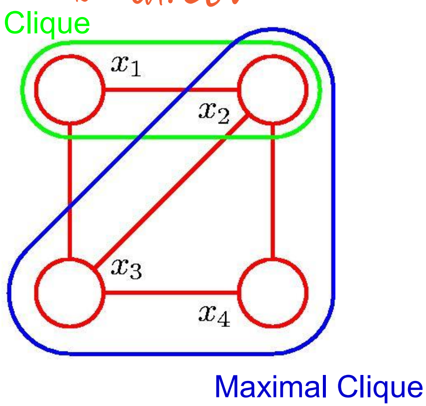
$$p(\mathbf{x}) = \frac{1}{Z}\prod_C \psi_C(\mathbf{x}_C)$$

potential function

where $\psi_C(\mathbf{x}_C)$ is the potential over clique C and

$$Z = \sum_{\mathbf{x}}\prod_C \psi_C(\mathbf{x}_C)$$

is the normalization coefficient.



knowledge base?

Markov Logic: Intuition

- A logical KB is a set of **hard constraints** on the set of possible worlds
  - If a world violates a formula, it becomes impossible
- Let's make them **soft constraints**: When a world violates a formula, it becomes less probable, not impossible
- Give each formula a **weight** (Higher weight $\Rightarrow$ Stronger constraint)

$0 \longleftrightarrow \infty$

$$P(\text{world}) \propto \exp\left(\sum \text{weights of formulas it satisfies}\right)$$

$\left(\frac{1}{Z}\right)$

Markov Logic: Definition

- A Markov Logic Network (MLN) is a set of pairs (F, w) where
  - F is a formula in first-order logic
  - w is a real number
- Together with a set of constants, it defines a Markov network with

  = objects

  把FOL中的var换成 ground term

  NODE: grounding
  - One node for each grounding of each predicate in the MLN

    命题化
    - This is *propositionalization* (remember?)

  Edge:
  - One clique for each grounding of each formula F in the MLN, with the potential being:
    - if exp(w) for node assignments that satisfy F
    - else: 1 otherwise

potential clique

for

通用实例化

Universal instantiation (UI)

result of substituting a

object

(Term without variables)

- For any sentence α, variable *v* and <u>ground term</u> *g*:

$\forall x\ A$

with free x in A

$$\frac{\forall v\ \alpha}{\mathrm{Subst}(\{v/g\}, \alpha)}$$ ← Substitute *v* with *g* in *α*

substitute

$\Rightarrow A: \{x \mapsto \alpha\}$.

- Every instantiation of a universally quantified sentence is entailed by it
- UI can be applied multiple times to add new sentences
- E.g., $\forall x\ King(x) \wedge Greedy(x) \Rightarrow Evil(x)$ yields:
  - $King(John) \wedge Greedy(John) \Rightarrow Evil(John)$
  - $King(Richard) \wedge Greedy(Richard) \Rightarrow Evil(Richard)$
  - $King(Father(John)) \wedge Greedy(Father(John)) \Rightarrow Evil(Father(John))$

"可代入"

Ground expression

From Wikipedia, the free encyclopedia

In mathematical logic, a **ground term** of a formal system is a term that does not contain any variables. Similarly, a **ground formula** is a formula that does not contain any variables.

In first-order logic with identity, the sentence $Q(a) \lor P(b)$ is a ground formula, with a and b being constant symbols. A **ground expression** is a ground term or ground formula.

Contents [hide]

Examples [edit]

Consider the following expressions in first order logic over a signature containing a constant symbol 0 for the number 0, a unary function symbol s for the successor function and a binary function symbol $+$ for addition.

- $s(0), s(s(0)), s(s(s(0))), \ldots$ are ground terms,
- $0+1,\ 0+1+1, \ldots$ are ground terms,
- $x+s(1)$ and $s(x)$ are terms, but not ground terms,
- $s(0)=1$ and $0+0=0$ are ground formulae,

Formal definition [edit]

What follows is a formal definition for first-order languages. Let a first-order language be given, with C the set of constant symbols, V the set of (individual) variables, F the set of functional operators, and P the set of predicate symbols.

Ground terms [edit]

A **ground term** is a terms that contain no variables. They may be defined by logical recursion (formula-recursion):

1. Elements of C are ground terms;
2. If $f \in F$ is an n-ary function symbol and $\alpha_1, \alpha_2, \ldots, \alpha_n$ are ground terms, then $f(\alpha_1, \alpha_2, \ldots, \alpha_n)$ is a ground term.
3. Every ground term can be given by a finite application of the above two rules (there are no other ground terms; in particular, predicates cannot be ground terms).

Roughly speaking, the Herbrand universe is the set of all ground terms.

Ground atom [edit]

A **ground predicate**, **ground atom** or **ground literal** is an atomic formula all of whose argument terms are ground terms.

If $p \in P$ is an n-ary predicate symbol and $\alpha_1, \alpha_2, \ldots, \alpha_n$ are ground terms, then $p(\alpha_1, \alpha_2, \ldots, \alpha_n)$ is a ground predicate or ground atom.

Roughly speaking, the Herbrand base is the set of all ground atoms, while a Herbrand interpretation assigns a truth value to each ground atom in the base.

Ground formula [edit]

A **ground formula** or **ground clause** is a formula without variables.

Formulas with free variables may be defined by syntactic recursion as follows:

1. The free variables of an unground atom are all variables occurring in it.
2. The free variables of $\neg p$ are the same as those of p. The free variables of $p \lor q, p \land q, p \to q$ are those free variables of p or free variables of q.
3. The free variables of $\forall x\ p$ and $\exists x\ p$ are the free variables of p except x.

Reduction contd.

- Every FOL KB can be propositionalized so as to preserve entailment 保留
  - i.e., a ground sentence is entailed by new KB iff entailed by original KB
- A naïve idea for FOL inference:
  - propositionalize KB and query, apply resolution, return result
- Problem: with function symbols, there are infinitely many ground terms,
  - e.g., *Father*(*Father*(*Father*(*John*)))

Example: Friends & Smokers

Smoking causes cancer.

Friends have similar smoking habits.

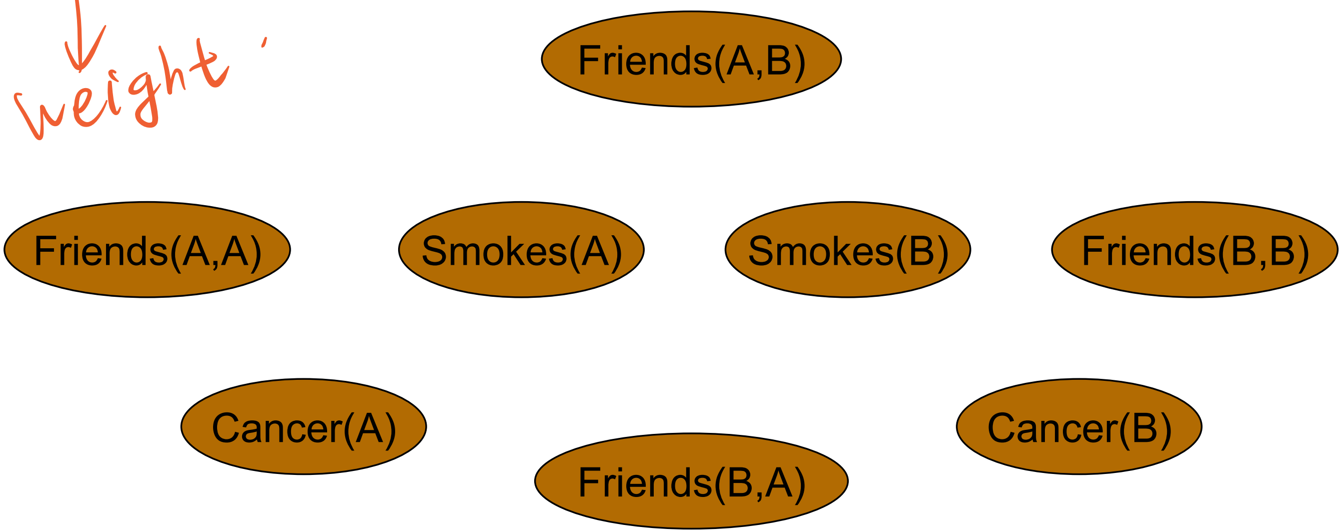
FOL

Example: Friends & Smokers

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

weight

Two constants: **Anna** (A) and **Bob** (B)

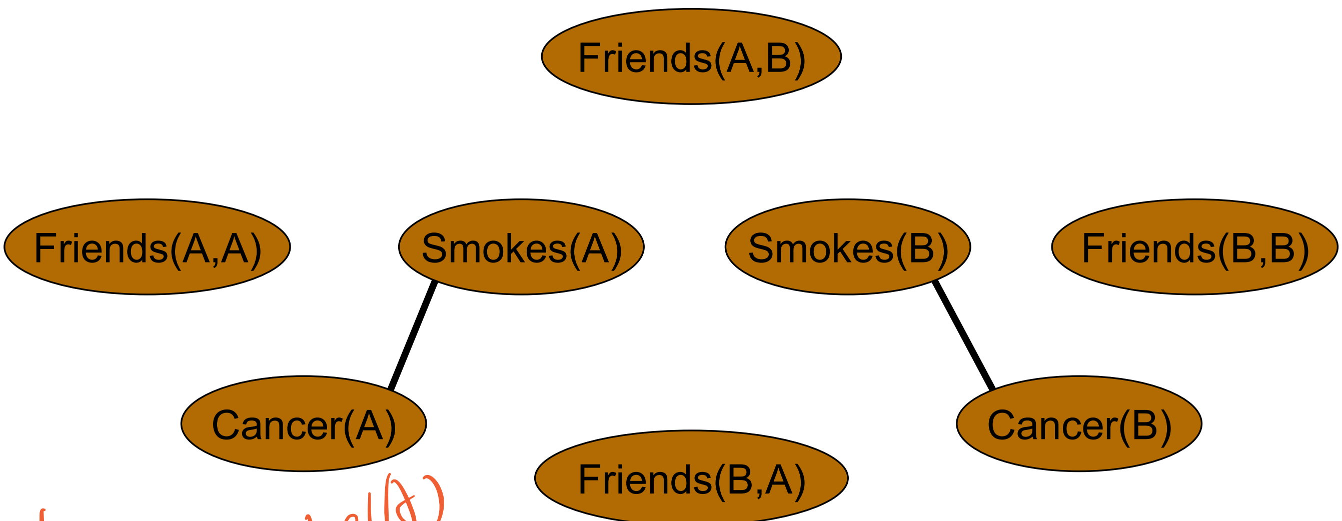


Example: Friends & Smokers

Predict: smoke cancer friends.

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x,y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

Two constants: **Anna** (A) and **Bob** (B)

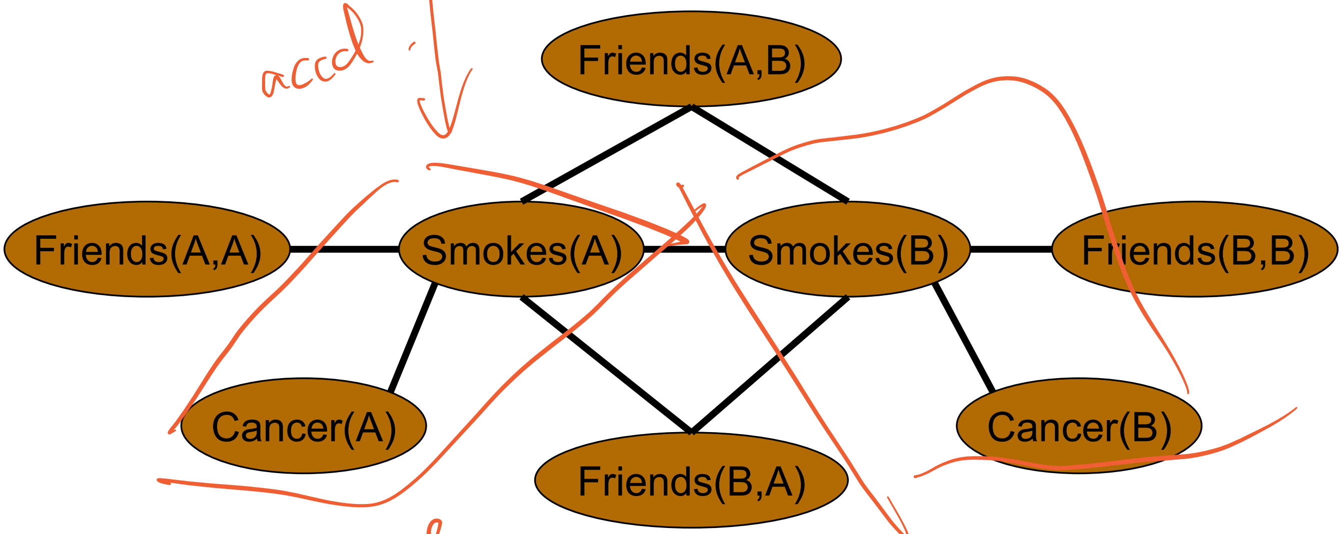


smokes $\rightarrow \left\{ \begin{array}{l} \text{smoke(A)} \\ \text{smoke(B)} \end{array} \right.$

Example: Friends & Smokers

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

Two constants: **Anna** (A) and **Bob** (B)



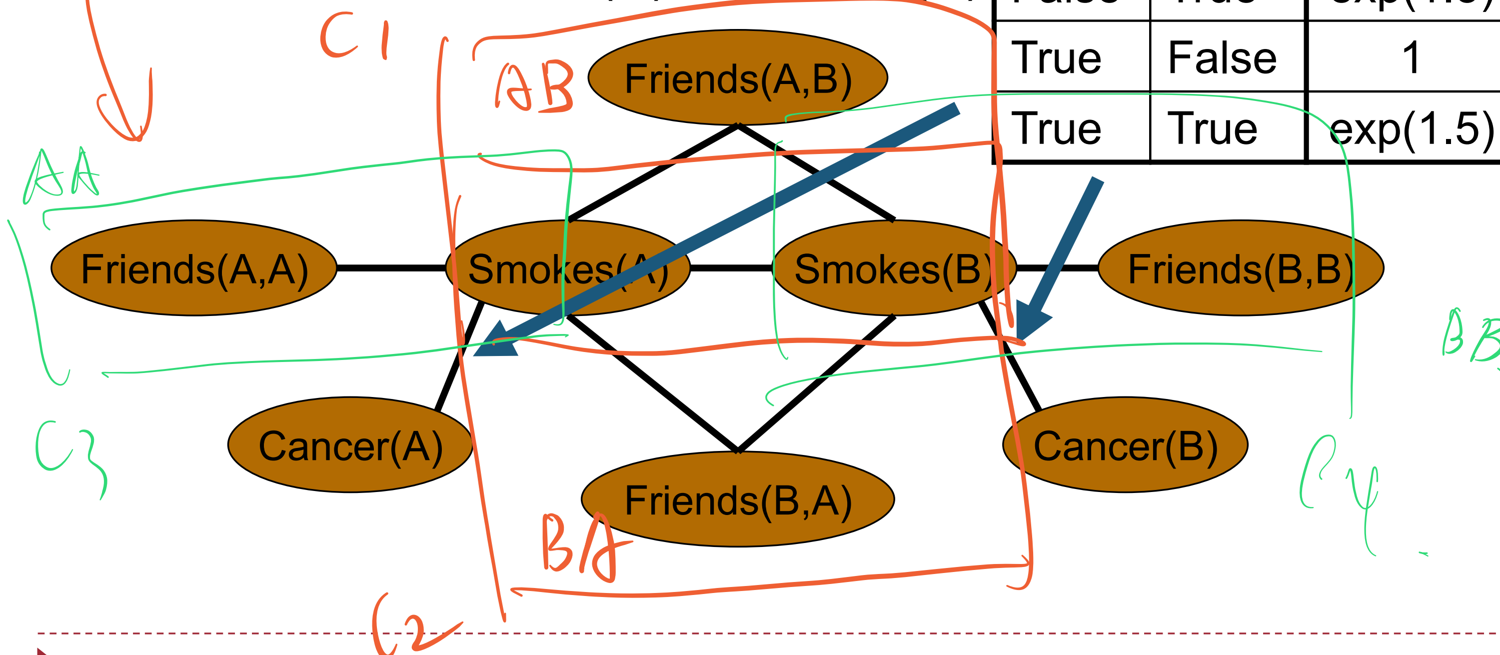
clique

Example: Friends & Smokers

1.5	$\forall x \; Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y \; Friends(x, y) \Rightarrow \big(Smokes(x) \Leftarrow$

S	C	Φ(S,C)
False	False	exp(1.5)
False	True	exp(1.5)
True	False	1
True	True	exp(1.5)

Two constants: **Anna** (A) and **Bob** (B)



Example: Friends & Smokers

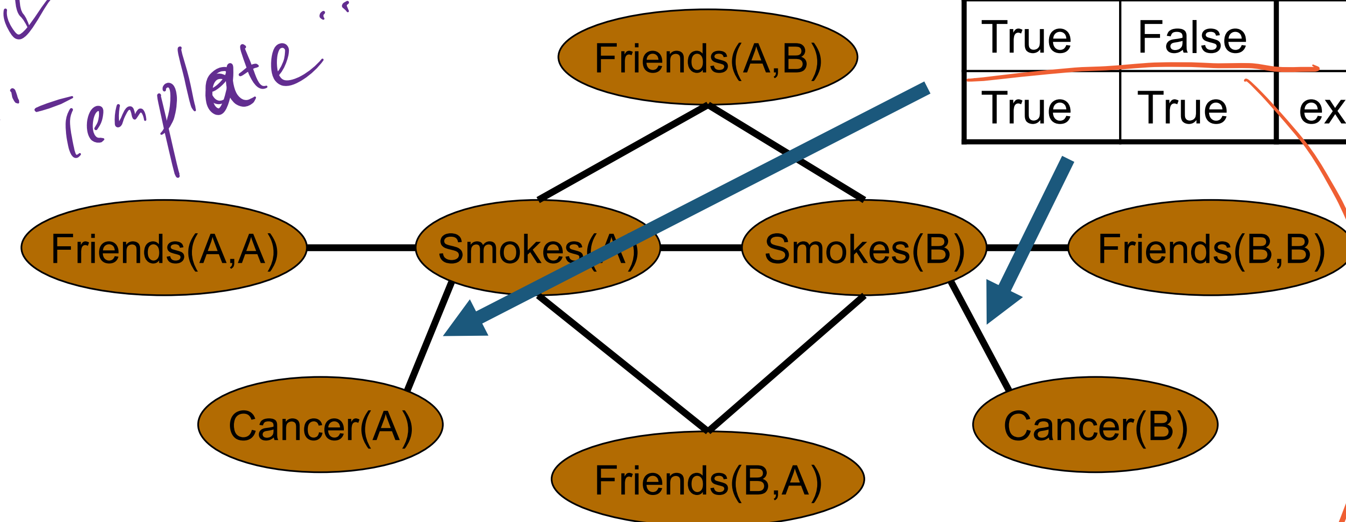
constraints: potential func:

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow (Smokes(x) \Leftarrow$

S	C	Φ(S,C)
False	False	exp(1.5)
False	True	exp(1.5)
True	False	1
True	True	exp(1.5)

Two constants: **Anna** (A) and **Bob** (B)

"Template"



not satisfy

constraint

so, /

Markov Logic Networks

- MLN is **template** for ground Markov nets
- Probability of a world x:

possible world

$$P(x) = \frac{1}{Z}\exp\left(\sum_i w_i n_i(x)\right)$$

world

Weight of formula i

No. of true groundings of formula i in x

个数.

Example: Friends & Smokers

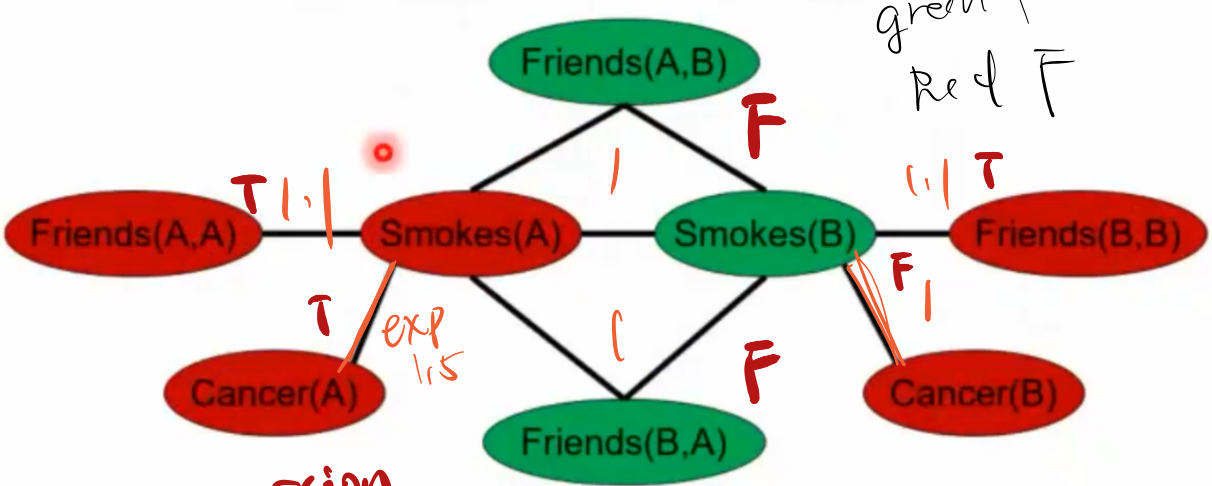
1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x,y) \Rightarrow (Smokes(x) \Leftrightarrow Smokes(y))$

2个 clique

4个

Two constants: **Anna** (A) and **Bob** (B)

green T
Red F



m: expression (clique)

$\Rightarrow p(x) \propto \exp(1.1 + 1.1 + 1.5)$

only T!

Remember: 对每个表达式在每个 group 构建 clique.

Relation to First-Order Logic

- Infinite weights $\Rightarrow$ First-order logic
  - P(x)>0 iff. x satisfies KB

- Markov logic allows contradictions between formulas

eg.

$F \quad w_1$

$\neg F \quad w_2$

$F, \neg F$

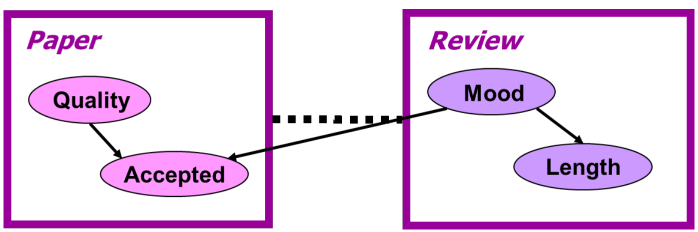
contain 2 contradict

$F_{all} = w_1 - w_2$

soft. constraints.

Relation to PRM

- MLN is More general and flexible than PRM
- In principle, a PRM can be converted into a MLN by writing a formula for each entry of each CPT and setting the weight to be the logarithm of the conditional probability



Q, M	$P(A \mid Q, M)$	
f, f	0.1	0.9
f, t	0.2	0.8
t, f	0.6	0.4
t, t	0.7	0.3

paper review

Trans

log0.1: $\forall$x,y hasReview(x,y) $\wedge$ Q(x,f) $\wedge$ M(y,f) $\wedge$ A(x,t)

log0.9: $\forall$x,y hasReview(x,y) $\wedge$ Q(x,f) $\wedge$ M(y,f) $\wedge$ A(x,f)

log0.2: $\forall$x,y hasReview(x,y) $\wedge$ Q(x,f) $\wedge$ M(y,t) $\wedge$ A(x,t)

……

Software of MLN

- Alchemy
  - https://alchemy.cs.washington.edu/

Inference

Inference

- A naive approach
  - Unroll the model to a BN or MN and run inference algorithms (such as VE)
  - Problem: the BN/MN may be very large and highly interconnected

repeat alot

- Lifted inference

similar structure

  - Lots of repeated structures in the unrolled model ⇒ repeated computation in inference
  - Group similar random variables at the FOL level and handle them at the same time

scheme layer

lift

compute

instance lage

Summary

- Probabilistic Relational Models
  - Logical language: Frame
  - Probabilistic language: Bayes nets
  - Bayes net “template” for object classes
  - Object’s attrs. can depend on attrs. of related objs.
- Markov Logic
  - Logical language: First-order logic
  - Probabilistic language: Markov networks
  - Syntax: First-order formulas with weights
  - Semantics: Templates for Markov net cliques