Intelligent Agents:
Web-mining Agents

Probabilistic Graphical Models

Probabilistic Relational Models

Tanya Braun

UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR INFORMATIONSSYSTEME

Probabilistic Graphical Models (PGMs)

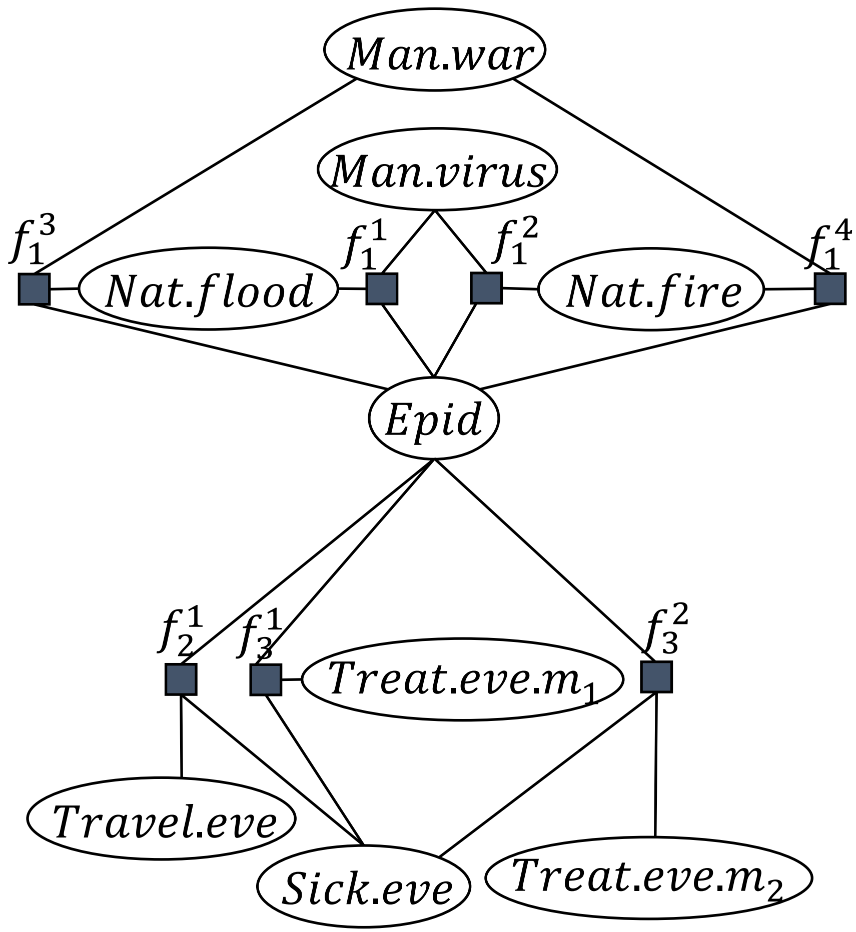
1. Recap: **Propositional** modelling
   - Factor model, Bayesian network, Markov network
   - Semantics, inference tasks + algorithms + complexity
2. **Probabilistic relational models** (PRMs)
   - Parameterised models, Markov logic networks
   - Semantics, inference tasks
3. **Lifted inference**
   - LVE, LJT, FOKC
   - Theoretical analysis
4. **Lifted learning**
   - Recap: propositional learning
   - From ground to lifted models
   - Direct lifted learning
5. **Approximate Inference: Sampling**
   - Importance sampling
   - MCMC methods
6. **Sequential models & inference**
   - Dynamic PRMs
   - Semantics, inference tasks + algorithms + complexity, learning
7. **Decision making**
   - (Dynamic) Decision PRMs
   - Semantics, inference tasks + algorithms, learning
8. **Continuous Space**
   - Gaussian distributions and Bayesian networks
   - Probabilistic soft logic

Outline: 2. Probabilistic relational models

A. ***Parameterised models (PMs)***
- Motivation: Symmetries and relations
- Syntax, semantics
- Graphical representation
- Inference tasks

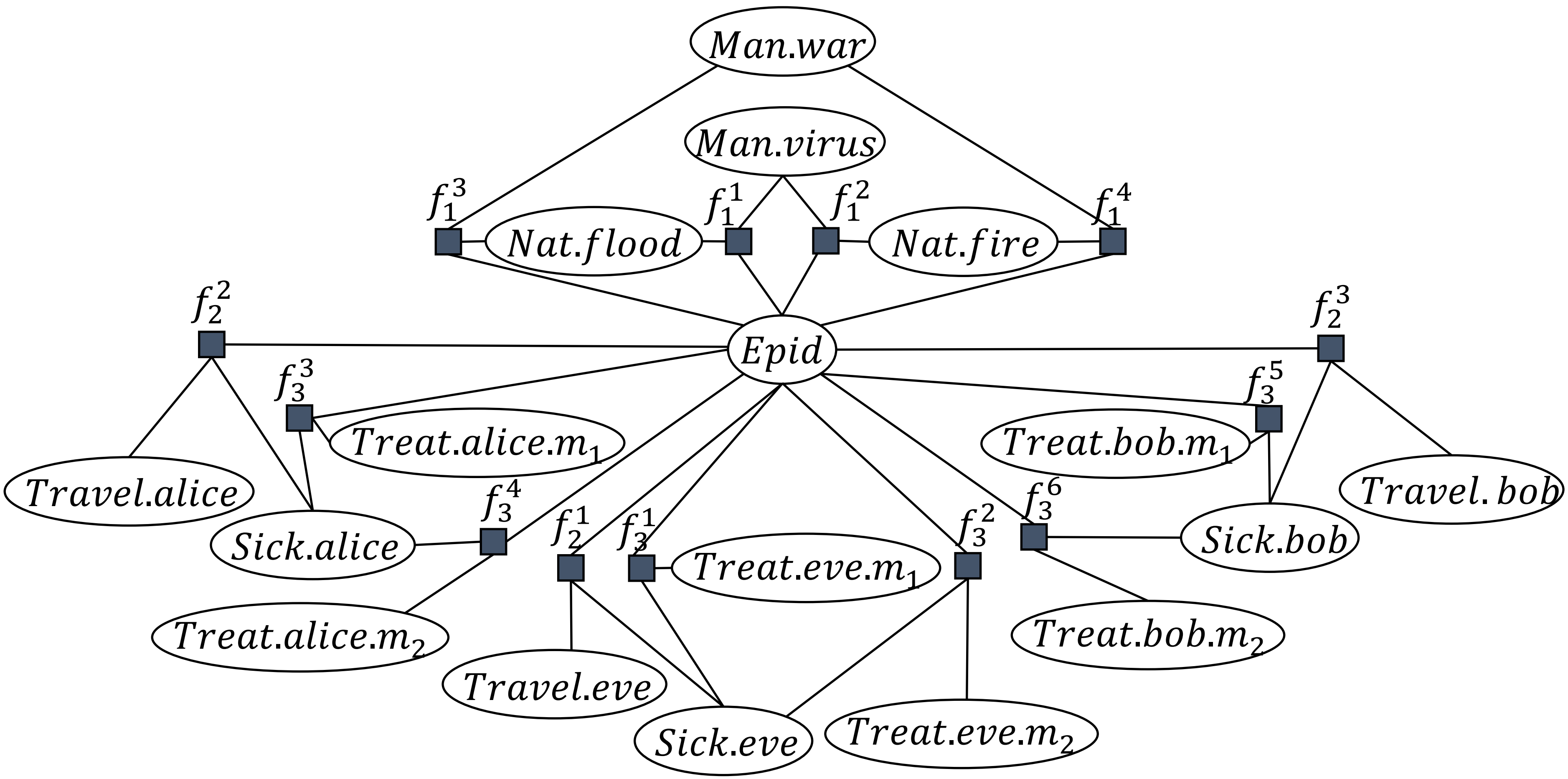
B. *Markov logic networks (MLNs)*
- Syntax of first-order logic
- Syntax, semantics of MLNs
- Graphical representation
- Turning MLNs into PMs and vice versa

Problem: Models Explode



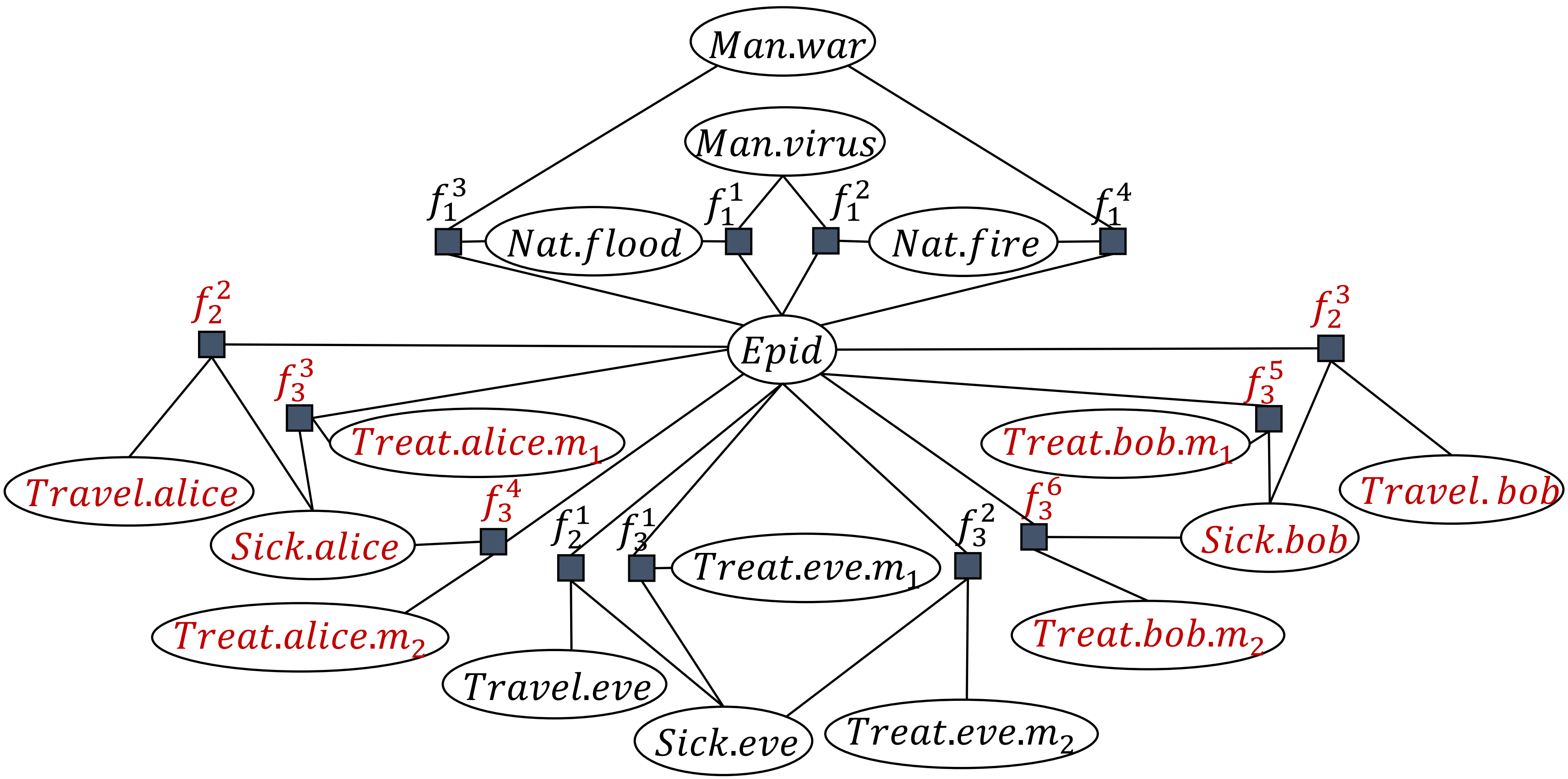
$7 \cdot 2^3 = 56$ entries in 7 factors, 9 variables

Problem: Models Explode



$13 \cdot 2^3 = 104$ entries in 13 factors, 17 variables

Propositional → First-order View



Symmetries in graph / relations in scenario

Encoding Relations and Symmetries

- Logical variables (logvars) to encode
  - Recurring structures in graphs
  - Relational structures in data
- By parameterising random variables with logvars
- And using those parameterised random variables (PRVs) as inputs to factors
  - Forming parametric factors (parfactors)
- Allowing to encode
  - Duplicate factors only once
  - That factors apply to multiple individuals (constants)

David Poole: First-order Probabilistic Inference. In *IJCAI-03 Proceedings of the 18th International Joint Conference on Artificial Intelligence*, 2003.
Brian Milch, Luke S. Zettelmoyer, Kristian Kersting, Michael Haimes, and Leslie Pack Kaelbling. Lifted Probabilistic Inference with Counting Formulas AAAI-08 Proceedings of the 23rd Conference on Artificial Intelligence, 2008.
Nima Taghipour: Lifted Probabilistic Inference by Variable Elimination. KU Leuven, 2013.
Tanya B: Rescued from a Sea of Queries: Exact Inference in Probabilistic Relational Models. UzL, 2020.

Logvars, Domains, Universe

- Logvar L
  - to encode patterns or relations (first-order constructs)
- Possible values (constants) of logical variables = domain $\mathcal{D}(L) = \{c_1, \dots, c_k\}$
- E.g.,
  - Logvars D, M, W, X with domains
    - $\mathcal{D}(D) = \{fire, flood\}$
    - $\mathcal{D}(M) = \{m_1, m_2\}$
    - $\mathcal{D}(W) = \{virus, war\}$
    - $\mathcal{D}(X) = \{alice, eve, bob\}$
- Universe $\boldsymbol{D} = \{c_1, \dots, c_K\}$
  - All constants occurring in a scenario, i.e., $\mathcal{D}(L) \subseteq \boldsymbol{D}$

Constraints

- To restrict applicability of domain constants:
  Constraint $C = (\mathcal{X}, C_{\mathcal{X}})$
  - $\mathcal{X} = (X_1, \dots, X_n)$ a sequence of logvars
  - $C_{\mathcal{X}} \subseteq \times_{i=1}^{n} \mathcal{D}(X_i)$ a subset of domain combinations
    - Set of sequences of constants in the order given by $\mathcal{X}$
  - If no restrictions apply, i.e, $C_{\mathcal{X}} = \times_{i=1}^{n} \mathcal{D}(X_i)$: $C = \top$
    - Can be omitted
- E.g.,
  - $(X, \{(alice), (eve)\})$
  - $((X, M),$
    $\{(alice, m_1), (alice, m_2),$
    $(eve, m_1), (eve, m_2),$
    $(bob, m_1), (bob, m_2)\}) = \top$

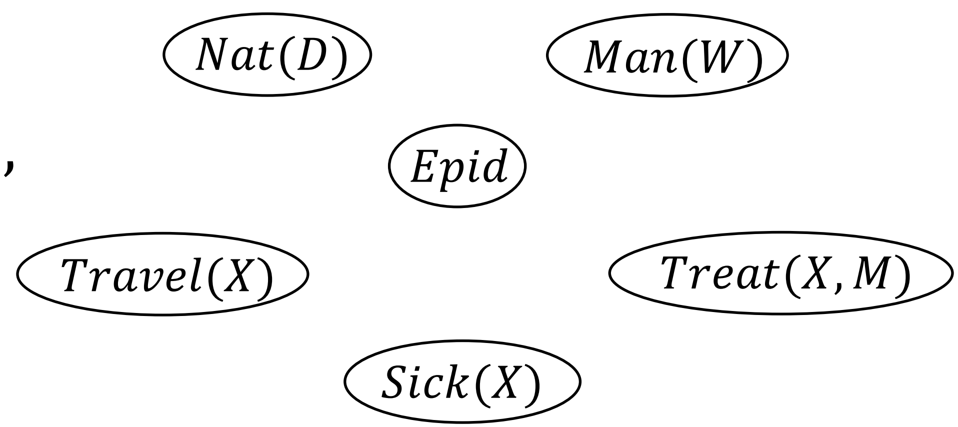
Constraints =
Abstraction of a database

PRVs

- Random variable R parameterised with logical variables $X_1, \dots, X_n, n \geq 0$
  = PRV

$$A = R(X_1, \dots, X_n)$$

  - if $n = 0$, $A = R$ a propositional random variable
- Range as before: $\mathcal{R}(A) = \{v_1, \dots, v_m\}$
- E.g.,
  - $Treat(X, M)$ with $n = 2$
  - $Epid$ with $n = 0$
  - $Treat(X, M), Travel(X), Sick(X)$ sharing the same logvar



Restricting PRVs

- PRV A under constraint C written as $A_{|C}$
  - $C = (\mathcal{X}, C_{\mathcal{X}}), \mathcal{X} = lv(A)$
    - $lv(.)$ referring to the logvars occurring in the given input
  - Represents a set of propositional random variables
- E.g.,
  - $Sick(X)_{|C}$
    - $C = (X, \{(alice), (eve)\})$
    - Represents $Sick(alice), Sick(eve)$
  - $Treat(X, M)_{|\top}$
    - $\top = \big((X, M), \mathcal{D}(X) \times \mathcal{D}(M)\big)$
    - Represents $Treat(alice, m_1), Treat(alice, m_2),$ $Treat(eve, m_1), Treat(eve, m_2),$ $Treat(bob, m_1), Treat(bob, m_2)$

Grounding of PRVs

- Grounding PRVs given constraints: $gr(A_{|(\mathcal{X},C_{\mathcal{X}})})$
  - Obtain set of propositional random variables represented by $A_{|(\mathcal{X},C_{\mathcal{X}})}$
  - Substitute logvars with constants named in constraint, i.e.,
    - Build substitutions

$$\boldsymbol{\theta} = \bigcup_{\boldsymbol{x} \in C_{\mathcal{X}}} \left\{ \bigcup_{x \in \boldsymbol{x}} \{X \to x\} \right\}$$

    - Apply the substitutions in $\boldsymbol{\theta}$ to A

$$gr(A_{|(\mathcal{X},C_{\mathcal{X}})}) = \bigcup_{\theta_{\boldsymbol{x}} \in \boldsymbol{\theta}} A\theta_{\boldsymbol{x}}$$

Grounding: Example

$$gr\left(A_{|(\mathcal{X},C_{\mathcal{X}})}\right) = \bigcup_{\theta_{\boldsymbol{x}} \in \boldsymbol{\theta}} A\theta_{\boldsymbol{x}}, \boldsymbol{\theta} = \bigcup_{\boldsymbol{x} \in C_{\mathcal{X}}} \left\{ \bigcup_{x \in \boldsymbol{x}} \{X \to x\} \right\}$$

- E.g.,
  - $gr\left(Sick(X)_{|C}\right) = \{Sick(alice), Sick(eve)\}$
    - $C = (X, \{(alice), (eve)\})$
    - $\{\theta_{alice}, \theta_{eve}\} = \left\{\{X \to alice\}, \{X \to eve\}\right\}$
  - $gr\left(Treat(X,M)_{|\top}\right) = \{Treat(alice, m_1), Treat(alice, m_2), Treat(eve, m_1), Treat(eve, m_2), Treat(bob, m_1), Treat(bob, m_2)\}$
    - $\top = \left((X,M), \mathcal{D}(X) \times \mathcal{D}(M)\right)$
    - $\left\{\theta_{alice,m_1}, \theta_{alice,m_2}, \theta_{eve,m_1}, \theta_{eve,m_2}, \theta_{bob,m_1}, \theta_{bob,m_2}\right\} = \left\{\{X \to alice, M \to m_1\}, \{X \to alice, M \to m_2\}, \{X \to eve, M \to m_1\}, \{X \to eve, M \to m_2\}, \{X \to bob, M \to m_1\}, \{X \to bob, M \to m_2\}\right\}$

Parfactors

- Factor with PRVs as arguments = parfactors
  $g = \forall \boldsymbol{x} \in C_{\mathcal{X}} : \phi(\mathcal{A}\theta_{\boldsymbol{x}})_{|(\mathcal{X}, C_{\mathcal{X}})}$
  - $\mathcal{A} = (A_1, \dots, A_k)$ sequence of PRVs containing logvars $\mathcal{X}$
  - ϕ as before:
    $$\phi: \times_{i=1}^{k} \mathcal{R}(A_i) \longrightarrow \mathbb{R}^{0,+}$$
    - At least one potential > 0
  - Shorthand $\phi(\mathcal{A})_{|C}$
    - If $C = \top$: $\phi(\mathcal{A})$
  - E.g., $g_2 =$
    $\forall \boldsymbol{x} \in C_{\mathcal{X}}: \phi\big(Travel(x), Epid, Sick(x)\big)_{|((X),\{(alice),(eve),(bob)\})}$
    $= \phi\big(Travel(X), Epid, Sick(X)\big)_{|\top}$
    $= \phi\big(Travel(X), Epid, Sick(X)\big)$

$Travel(X)$	$Epid$	$Sick(X)$	g_2
$false$	$false$	$false$	5
$false$	$false$	$true$	0
$false$	$true$	$false$	4
$false$	$true$	$true$	6
$true$	$false$	$false$	4
$true$	$false$	$true$	6
$true$	$true$	$false$	2
$true$	$true$	$true$	9

Grounding of Parfactors

- Grounding of parfactor $gr(g)$
  - Obtain set of factors $\{f_i\}_{i=1}^m$ that g represents
  - Replace logvars in g with constants in constraint of g:

$$gr\left(\phi(\mathcal{A})_{|(\mathcal{X},C_{\mathcal{X}})}\right) = \bigcup_{\theta_{\boldsymbol{x}} \in \boldsymbol{\theta}} \phi(\mathcal{A}\theta_{\boldsymbol{x}})$$

$$\boldsymbol{\theta} = \bigcup_{\boldsymbol{x} \in C_{\mathcal{X}}} \left\{ \bigcup_{x \in \boldsymbol{x}} \{X \to x\} \right\}$$

- E.g., $gr(g_2) = \{f_2^1, f_2^2, f_2^3\}$
  - $g_2 = \phi\left(Travel(X), Epid, Sick(X)\right)_{|\top}$

$Travel(X)$	$Epid$	$Sick(X)$	g_2
$false$	$false$	$false$	5
$false$	$false$	$true$	0
$false$	$true$	$false$	4
$false$	$true$	$true$	6
$true$	$false$	$false$	4
$true$	$false$	$true$	6
$true$	$true$	$false$	2
$true$	$true$	$true$	9

Grounding: Example

- E.g., $gr(g_2) = \{f_2^1, f_2^2, f_2^3\}$

$Travel(X)$	$Epid$	$Sick(X)$	g_2
$false$	$false$	$false$	5
$false$	$false$	$true$	0
[illegible]	[illegible]	[illegible]	[illegible]

$Travel(eve)$	$Epid$	$Sick(eve)$	f_2^1
$false$	$false$	$false$	5
$false$	$false$	$true$	0
$false$	$true$	$false$	4
$false$	$true$	$true$	6
$true$	$false$	$false$	4
$true$	$false$	$true$	6
$true$	$true$	$false$	2
$true$	$true$	$true$	9

…(bob)	$Epid$	$Sick(bob)$	f_2^3
…e	$false$	$false$	5
…e	$false$	$true$	0
…e	$true$	$false$	4
…e	$true$	$true$	6
…e	$false$	$false$	4
…e	$false$	$true$	6
…e	$true$	$false$	2
$true$	$true$	$true$	9

$Travel(alice)$	Epi…		
$false$	$fals$…		
$false$	$fals$…		
$false$	$true$…		
$false$	$true$…		
$true$	$fals$…		
$true$	$fals$…		
$true$	$true$	$false$	2
$true$	$true$	$true$	9

Symmetries within

- Assume four possible epidemics with identical characteristics
  - $Epid_1, Epid_2, Epid_3, Epid_4$
  - Reasonable to model the epidemics such that it does not matter which $Epid$ variables specifically are $true$ or $false$, i.e., they are interchangeable
    - All $false$ maps to 8
    - 1 $true$, 3 $false$ maps to 6
    - 2 $true$, 2 $false$ maps to 4
    - 3 $true$, 1 $false$ maps to 2
    - All $true$ maps to 0

    → Five lines enough to describe

$\#_{true}$ $\#_{false}$

[0,4]	8
[1,3]	6
[2,2]	4
[3,1]	2
[4,0]	0

$Epid_1$	$Epid_2$	$Epid_3$	$Epid_4$	g
$false$	$false$	$false$	$false$	8
$false$	$false$	$false$	$true$	6
$false$	$false$	$true$	$false$	6
$false$	$false$	$true$	$true$	4
$false$	$true$	$false$	$false$	6
$false$	$true$	$false$	$true$	4
$false$	$true$	$true$	$false$	4
$false$	$true$	$true$	$true$	2
$true$	$false$	$false$	$false$	6
$true$	$false$	$false$	$true$	4
$true$	$false$	$true$	$false$	4
$true$	$false$	$true$	$true$	2
$true$	$true$	$false$	$false$	4
$true$	$true$	$false$	$true$	2
$true$	$true$	$true$	$false$	2
$true$	$true$	$true$	$true$	0

Counting Random Variable

- New PRV type:
  Counting random variable (CRV) $\#_X\left[R(\boldsymbol{X})_{|C}\right]$
  - Let $R(\boldsymbol{X})_{|C}$ be a PRV under constraint C with $lv\big(R(\boldsymbol{X})\big) = \{X\}$
    - I.e., $\boldsymbol{X}$ is a singleton or other parameters of R are constant
  - With new range value: Histogram $h = \{(v_i, n_i)\}_{i=1}^{m}$
    - $m = \left|\mathcal{R}\big(R(\boldsymbol{X})\big)\right|, v_i \in \mathcal{R}\big(R(\boldsymbol{X})\big)$
    - $n_i \in \mathbb{N}, n = \sum_{i=1}^{m} n_i = \left|gr\big(R(\boldsymbol{X})_{|C}\big)\right|$
    - Shorthand: $[n_1, \dots, n_m]$
    - Encodes several interchangeable assignments at once
      - Given by multinomial coefficient $Mul(h)$

$$Mul(h) = \frac{n!}{\prod_{i=1}^{m} n_i!}$$

  - Range of a CRV = space of histograms fulfilling the conditions above

CRV: Example

- Counting random variable (CRV) $\#_X[R(\boldsymbol{X})_{|C}]$
  - With new range value: Histogram $h = \{(v_i, n_i)\}_{i=1}^m$
    - $m = |\mathcal{R}(R(\boldsymbol{X}))|, v_i \in \mathcal{R}(R(\boldsymbol{X})), n_i \in \mathbb{N}, n = \sum_{i=1}^m n_i = |gr(R(\boldsymbol{X})_{|C})|$
    - Shorthand: $[n_1, \ldots, n_m]$
    - Encodes several interchangeable assignments $Mul(h) = \frac{n!}{\prod_{i=1}^m n_i!}$

- E.g.,
  - CRV: $\#_E[Epid(E)]$
    - $m = |\mathcal{R}(Epid(E))| = 2, v_i \in \mathcal{R}(Epid(E)) = \{true, false\}$
    - $n_i \in \mathbb{N}, n = \sum_{i=1}^m n_i = |gr(Epid(E)_{|\top})| = 4, \mathcal{D}(E) = \{e_1, e_2, e_3, e_4\}$
    - Range values and multiplicities:

$\{(true, 0), (false, 4)\} = [0,4]$ $\quad$ $Mul([0,4]) = \frac{4!}{0! \cdot 4!} = 1$

$\{(true, 1), (false, 3)\} = [1,3]$ $\quad$ $Mul([1,3]) = \frac{4!}{1! \cdot 3!} = 4$

$\{(true, 2), (false, 2)\} = [2,2]$ $\quad$ $Mul([2,2]) = \frac{4!}{2! \cdot 2!} = 6$

$\{(true, 3), (false, 1)\} = [3,1]$ $\quad$ $Mul([3,1]) = \frac{4!}{3! \cdot 1!} = 4$

$\{(true, 4), (false, 0)\} = [4,0]$ $\quad$ $Mul([4,0]) = \frac{4!}{4! \cdot 0!} = 1$

CRV: Example

- E.g., (continued)
  - CRV: $\#_E[Epid(E)]$
    - Range values
      $[0,4], [1,3], [2,2], [3,1], [4,0]$
      $1 \quad 4 \quad 6 \quad 4 \quad 1$
      how many assignments encoded
  - $g' = \phi(\#_E[Epid(E)])$

$\#_E[Epid(E)]$	g'
$[0,4]$	8
$[1,3]$	6
$[2,2]$	4
$[3,1]$	2
$[4,0]$	0

$Epid_1$	$Epid_2$	$Epid_3$	$Epid_4$	g
$false$	$false$	$false$	$false$	8
$false$	$false$	$false$	$true$	6
$false$	$false$	$true$	$false$	6
$false$	$false$	$true$	$true$	4
$false$	$true$	$false$	$false$	6
$false$	$true$	$false$	$true$	4
$false$	$true$	$true$	$false$	4
$false$	$true$	$true$	$true$	2
$true$	$false$	$false$	$false$	6
$true$	$false$	$false$	$true$	4
$true$	$false$	$true$	$false$	4
$true$	$false$	$true$	$true$	2
$true$	$true$	$false$	$false$	4
$true$	$true$	$false$	$true$	2
$true$	$true$	$true$	$false$	2
$true$	$true$	$true$	$true$	0

CRVs Continued

- With
  - $m = \left|\mathcal{R}\left(R(\boldsymbol{X})\right)\right|$
  - $n = \sum_{i=1}^{m} n_i = \left|gr\left(R(\boldsymbol{X})_{|C}\right)\right|$
- Instead of m^n mappings for the ground factor, the counted factor has

$$\binom{n+m-1}{n-1}$$

mappings
  - Upper bound of range size of a CRV:

$$\binom{n+m-1}{n-1} \leq n^m$$

CRVs Continued

- If $\{X\} \subset lv\big(R(\boldsymbol{X})\big)$ in $\#_X\big[R(\boldsymbol{X})_{|C}\big]$, CRV is a parameterised CRV (PCRV)
  - Represents a set of CRVs
- Counting binds a logvar, i.e.,

$$lv(\#_X[R(\boldsymbol{X})]) = lv\big(R(\boldsymbol{X})\big) \setminus \{\boldsymbol{X}\}$$

  - Constants that X represents made explicit through counts in histograms
- Restriction:
  Only one logvar can be counted at a time!
  - In the current formalisation
    (it might be possible to define more complex counting operations)

Parfactors Revisited

- Factor with PRVs as arguments = parfactors
  $$g = \forall \boldsymbol{x} \in C_{\mathcal{X}} : \phi(\mathcal{A}\theta_{\boldsymbol{x}})_{|(\mathcal{X}, C_{\mathcal{X}})}$$
  - $\mathcal{A} = (A_1, \dots, A_k)$ sequence of P(C)RVs containing logvars $\mathcal{X}$
  - ϕ as before:
    $$\phi: \times_{i=1}^{k} \mathcal{R}(A_i) \longrightarrow \mathbb{R}^{0,+}$$
    - At least one potential > 0
  - Shorthand $\phi(\mathcal{A})_{|C}$
    - If $C = \top$: $\phi(\mathcal{A})$
  - E.g.,
    $$g_2 = \phi\big(Travel(X), Epid, Sick(X)\big)$$
    $$g' = \phi(\#_X[Epid(E)])$$
    $$g'' = \phi(Sick(X), \#_X[Epid(E)])$$

$\mathcal{A}$ may contain ground random variables, PRVs, CRVs, PCRVs

$Sick(X)$	$\#_X[Epid(E)]$	g''
$false$	$[0,4]$	9
$false$	$[1,3]$	7
$false$	$[2,2]$	5
$false$	$[3,1]$	3
$false$	$[4,0]$	4
$true$	$[0,4]$	5
$true$	$[1,3]$	4
$true$	$[2,2]$	3
$true$	$[3,1]$	2
$true$	$[4,0]$	6

Parameterised Models

- Set of parfactors = Parameterised Model

$$G = \{g_i\}_{i=1}^{n}$$

  - Given the definitions of PRVs, it is possible to form a parameterised model that is completely propositional
  - E.g., $G = \{g_i\}_{i=1}^{3}$
    - $g_1 = \phi\big(Epid, Nat(D), Man(W)\big)$
    - $g_2 = \phi\big(Travel(X), Epid, Sick(X)\big)$
    - $g_3 = \phi\big(Epid, Sick(X), Treat(X, M)\big)$
    - Plus full specifications of potential functions
    - $\mathbf{3 \cdot 2^3 = 24}$ **entries** independent of domain sizes!
    - Instead of $\mathbf{13 \cdot 2^3 = 104}$ **entries** with domains

$\mathcal{D}(D) = \{fire, flood\}, |\mathcal{D}(D)| = 2$
$\mathcal{D}(M) = \{m_1, m_2\}, |\mathcal{D}(M)| = 2$
$\mathcal{D}(W) = \{virus, war\}, |\mathcal{D}(W)| = 2$
$\mathcal{D}(X) = \{alice, eve, bob\}, |\mathcal{D}(D)| = 3$

Parfactor Graphs

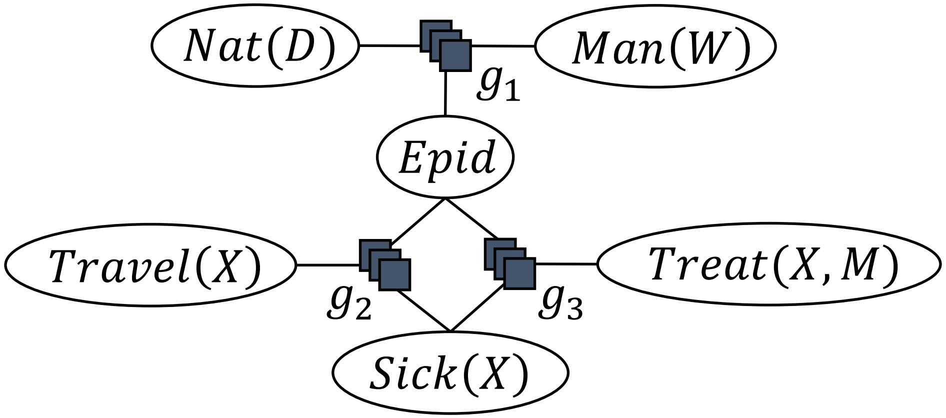
- Graphical representation of $G = \{g_i\}_{i=1}^{n}$: Parfactor graph
  - Analogous to factor graph
  - Each $A \in rv(G)$: variable node in graph (ellipse)
    - $rv(.)$ refers to the PRVs in its input
  - Each $g \in G$: factor node in graph (box)
  - For each argument A in $g \in G$: edge between variable node for A and factor node for g
  - Constraints are not depicted

- E.g.,

  $g_1 = \phi(Epid, Nat(D), Man(W))$

  $g_2 = \phi(Travel(X), Epid, Sick(X))$

  $g_3 = \phi(Epid, Sick(X), Treat(X, M))$

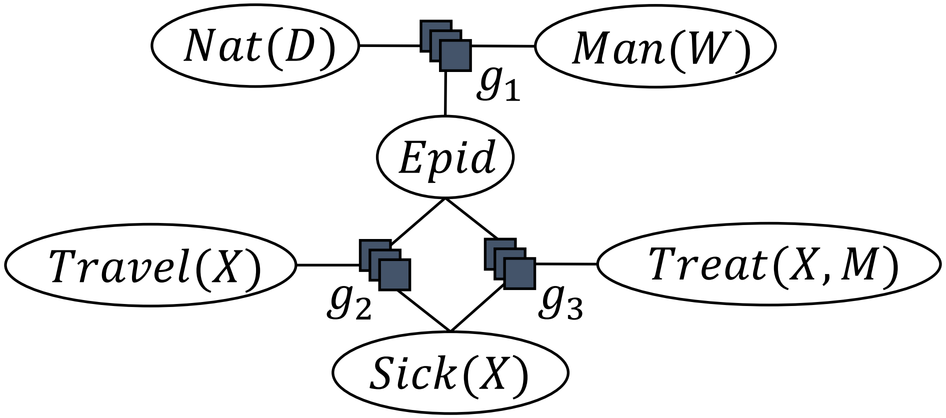


Grounding of Models

- Grounding of model G

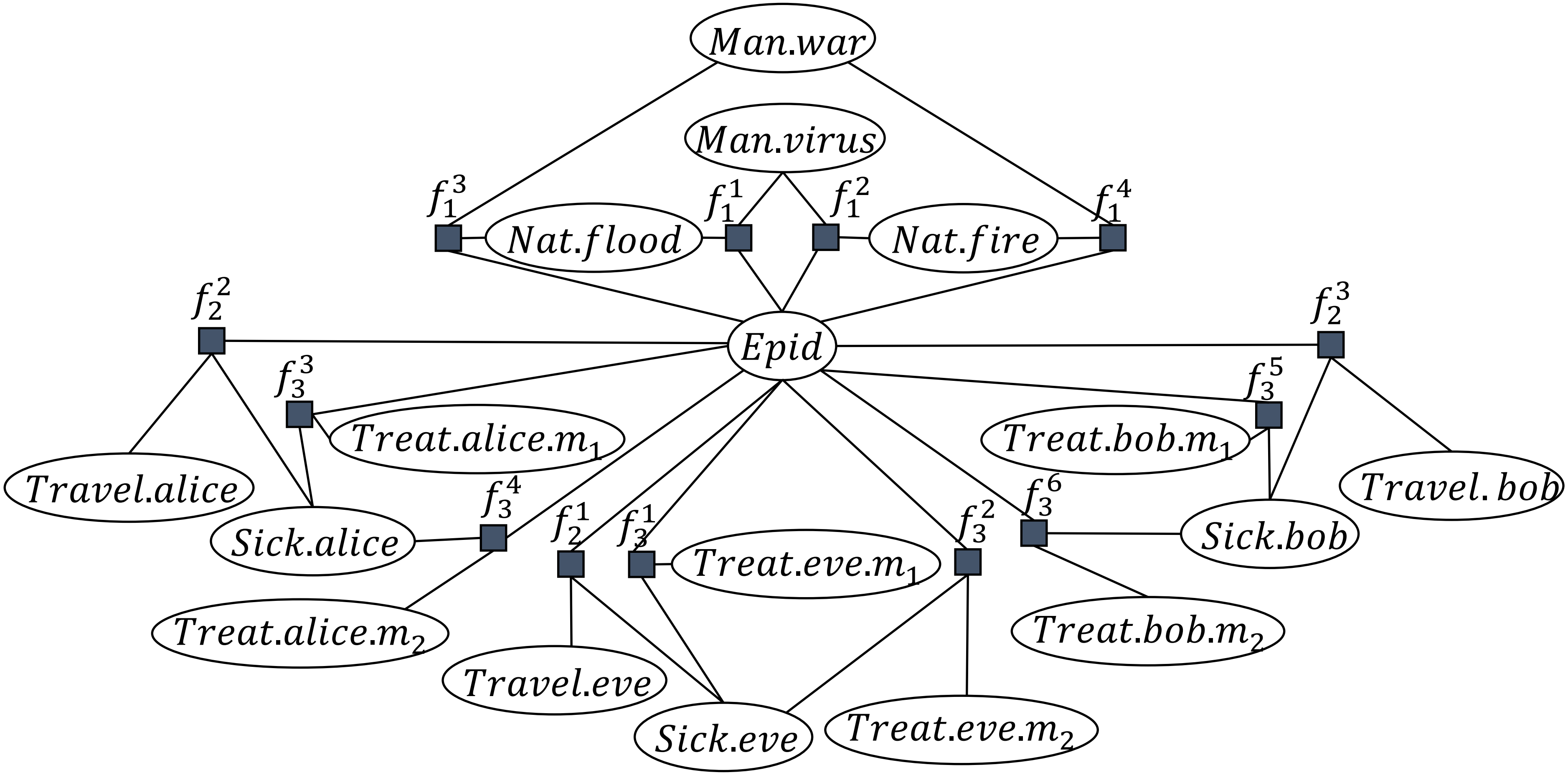
$$gr(G) = \bigcup_{g \in G} gr(g)$$

  - Depends on constraints and thus, domains
  - E.g., example model with $\top$ constraints and domains
    - $\mathcal{D}(D) = \{fire, flood\}$
    - $\mathcal{D}(M) = \{virus, war\}$
    - $\mathcal{D}(W) = \{m_1, m_2\}$
    - $\mathcal{D}(X) = \{alice, eve, bob\}$



Grounding of Models

- Graphical representation of grounding

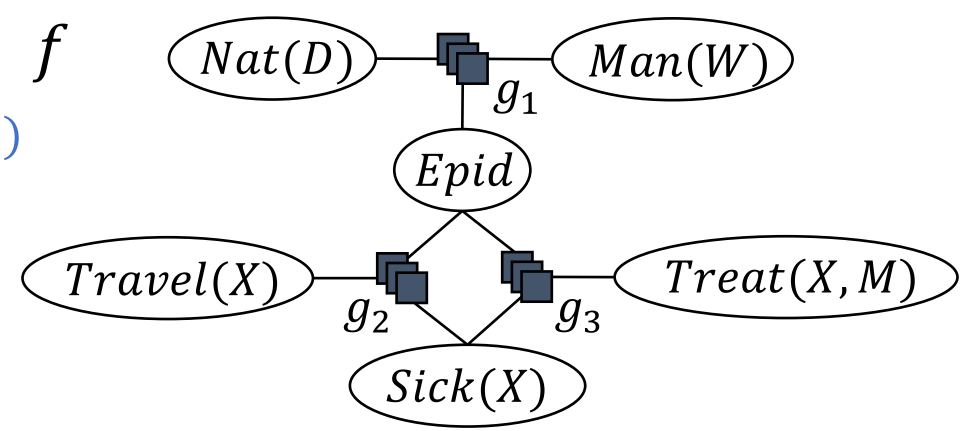


Semantics

- Given model $G = \{g_i\}_{i=1}^{n}$
  - Over PRVs $rv(G)$
  - Equivalent propositional model over random variables $F = gr\big(rv(G)\big) = \{R_1, \dots, R_N\}$ with semantics over P_F
- Semantics: Build full joint probability distribution P_G

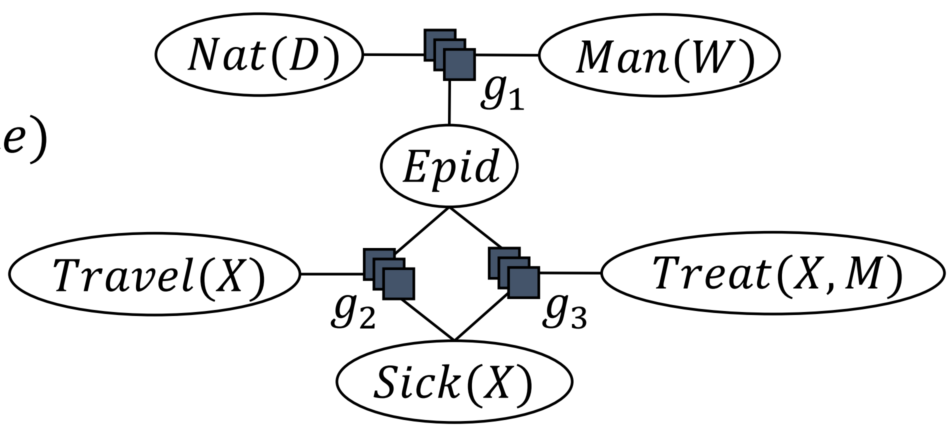
$$P_G = \frac{1}{Z} \prod_{f \in gr(G)} f$$

$$Z = \sum_{r_1 \in \mathcal{R}(R_1)} \dots \sum_{r_N \in \mathcal{R}(R_N)} \prod_{f \in gr(G)} f$$



Inference Tasks

- Again as before:
- Query Answering Problem
  - Compute an answer to a query $P(\boldsymbol{S}|\boldsymbol{T})$ given a model G representing the full joint probability distribution P_G
    - Query for a marginal (conditional) probability (distribution)
    - Avoid grounding (parts of) G
  - E.g.,
    - $P(Treat(eve, m_1))$
    - $P(Travel(eve), Epid)$
    - $P(Sick(eve)|Epid)$
    - $P(Epid|Sick(eve) = true)$
    - If a CRV occurs:
      $P(\#_E[Epid(E)])$
      $P(\#_E[Epid(E)] = [2,2])$



Evidence in Parameterised Models

- Observations for groundings of a PRV
  - Can be
    - One of the range values
    - Not available
  - E.g., for $Sick(X)$ with $\mathcal{D}(X) = \{x_1, \dots, x_n\}$
    - $Sick(x_1) = Sick(x_2) = \cdots = Sick(x_{10}) = true$
    - $Sick(x_{11}) = Sick(x_{12}) = \cdots = Sick(x_{20}) = false$
    - Observations for $Sick(x_{21})$ … $Sick(x_n)$ not available
- Compactly encode evidence with PRVs and parfactors

Evidence Parfactors

- Let $\boldsymbol{E}$ be the set of observations (evidence)
- For each PRV $A \in rv(\boldsymbol{E})$
  - For each range value r observed for a grounding of A
    - Set up a parfactor g_e with
      - A as argument
      - Map range value r to 1 and all range values $r' \neq r$ to 0
      - $C_{\mathcal{X}}$ in constraint $(\mathcal{X}, C_{\mathcal{X}})$ restricting the constants of $\mathcal{X}$ to the constants observed in A with r
        - Alternative: Use new logvars $\mathcal{X}'$ in A with domains corresponding to observed constants and $\top$ constraints in g_e

Evidence Parfactors: Example

- E.g.,
  - $\boldsymbol{E} =$
    $\{Sick(x_1) = Sick(x_2) = \cdots = Sick(x_{10}) = true,$
    $Sick(x_{11}) = Sick(x_{12}) = \cdots = Sick(x_{20}) = false\}$
  - For observations of $Sick(X)$ with range value $true$
    - Parfactor g_e^T
      - Argument: $Sick(X)$
      - $C_X = \{x_1, \ldots, x_{10}\}$ in constraint of g_e^T

$Sick(X)$	g_e^T
$false$	0
$true$	1

  - For observations of $Sick(X)$ with range value $false$
    - Parfactor g_e^F
      - Argument: $Sick(X)$
      - $C_X = \{x_{11}, \ldots, x_{20}\}$ in constraint of g_e^F

$Sick(X)$	g_e^F
$false$	1
$true$	0

Interim Summary

- First-order view on probabilistic modelling
  - Relations in data
  - Symmetries in graph
- Parameterised models
  - Logvars with domains of constants
    - Universe
  - Constraints to restrict domains to certain constants
  - PRVs to encode sets of propositional random variables
  - CRVs to encode symmetries within factors
  - Parfactors to build a model with recurring patterns
    - Semantics over grounding and full joint distribution
  - Inference tasks
    - Evidence parfactors

Outline: 2. Probabilistic relational models

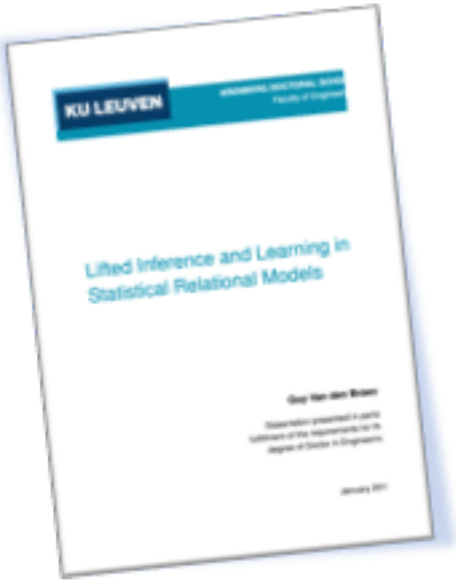
A. *Parameterised models (PMs)*
   - Motivation: Symmetries and relations
   - Syntax, semantics
   - Graphical representation
   - Inference tasks

B. ***Markov logic networks (MLNs)***
   - Syntax of first-order logic
   - Syntax, semantics of MLNs
   - Graphical representation
   - Turning MLNs into PMs and vice versa

Literature: Other than Books

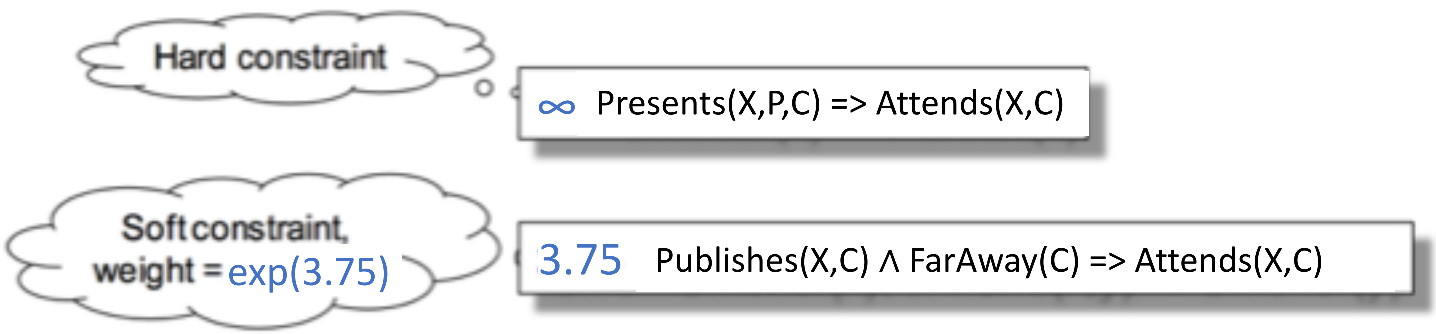
MLNs
FOKC (Section: Lifted Inference)

- ~~Two~~ Three PhD theses (especially for Sections 1-3):
  - Guy Van den Broeck: Lifted Inference and Learning in Statistical Relational Models. KU Leuven, 2013.
    - http://web.cs.ucla.edu/~guyvdb/phd/guyvdb-phdthesis.pdf
  - Nima Taghipour: Lifted Probabilistic Inference by Variable Elimination. KU Leuven, 2013.
    - https://lirias.kuleuven.be/1656026?limo=0
  - Tanya Braun: Rescued from a Sea of Queries: Exact Inference in Probabilistic Relational Models. UzL, 2020.
    - https://www.ifis.uni-luebeck.de/~braun/Diss/Braun_diss.pdf
- Further research papers referenced in slides



Markov Logic Networks (MLNs)

- Again: First-order view on probabilistic modelling
- Use logical formulas to specify potential functions
  - Weights for each formula induce a full joint again



- Next slides
  - Syntax of first-order logic with domain constraints
  - Syntax of MLNs
  - Semantics of MLNs

Matthew Richardson and Pedro Domingos: Markov Logic Networks. In: *Machine Learning*, 2006.

Alphabet

- Function-free first-order logic (FOL)
  - Logical symbols
    - Logical connectives $\neg, \wedge, \vee$, etc.
    - Quantifiers $\forall, \exists$
      - Mostly, we will be dealing with $\forall$
    - $true$ and $false$
    - Set of logical variables $\boldsymbol{X} = \{X, Y, \dots\}$
    - (Parentheses and punctuation)
  - Non-logical symbols (signature)
    - Set of predicate symbols p/n of arity n
      - Including propositional variables with arity $n = 0$
    - A set of constant symbols $\{a, b, c, \dots\}$
      - With a natural order on the constants

Alphabet

- FOL with domain constraints (FOL-DC)
  - Function-free FOL extended with
  - Logical symbols
    - Less-than $<$
    - Set membership and inclusion $\in$, $\subseteq$
    - Set of domain variables $\mathbf{D} = \{\mathrm{D}, \mathrm{F}, \dots\}$
    - Set operations $\cup$, $\cap$, $\setminus$
  - Non-logical symbols (signature)
    - Set of "sets of constants" symbols $\{\{a, b, c\}, \{a, d\}, \emptyset, \dots\}$

FOL-DC: Grammar

- Logical term
  - Logical variable or constant
  - Refers to objects in the domain of discourse (scenario)
- Domain term
  - Domain variable, set of constants or combination of two other domain terms t_d, t'_d in the form of $t_d \cup t'_d$, $t_d \cap t'_d$, or $t_d \setminus t'_d$
  - Refers to sets of objects in the domain of discourse
- Logical atom
  - $p(t_1, \ldots, t_n)$
  - Apply predicate p/n to n-tuple of logical term arguments t_i
- Domain atom
  - $t_l = t'_l$ or $t_l < t'_l$ between logical terms t_l, t'_l
  - $t_d = t'_d$ or $t_d \subseteq t'_d$ between domains terms t_d, t'_d
  - $t_l \in t_d$ between logical terms t_l, domain term t_d

FOL-DC: Grammar

- Domain constraint
  - Domain atom or its negation ($\neq, \notin, \nsubseteq$)
- Constraint set
  - Conjunction of domain constraints
    - (restricted to conjunction for the sake of simplicity)
    - (more expressive constraint languages would have the purpose to express certain constraint sets much more precisely)
  - E.g., $X \in \mathrm{Bird} \wedge X \neq kiwi$

FOL-DC: Grammar

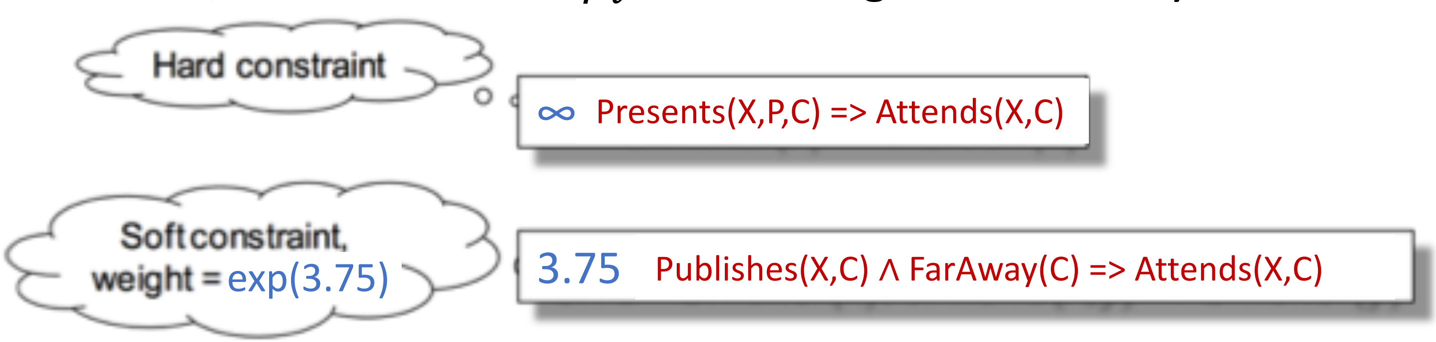
- Well-formed formula
  - Logical atom
  - If φ, ψ (well-formed) formulas, then the following are
    - $\neg\varphi$
    - *Extensional conjunction* $\varphi \land \psi$
    - *Extensional disjunction* $\varphi \lor \psi$
  - If φ formula, $\boldsymbol{V}$ a (potentially empty) set of (logical or domain) variables, and cs a constraint set that contains at least one domain atom of the form $V = t, V \in t$, or $V \subseteq t$ for every $V \in \boldsymbol{V}$, then the following are
    - Intensional conjunction $\forall \boldsymbol{V}, cs : \varphi$
    - Intensional disjunction $\exists \boldsymbol{V}, cs : \varphi$

Further Terminology

- Literal l
  - Logical atom a or its negation $\neg a$
- Clause
  - Disjunction of literals
- Theory in conjunctive normal form (CNF)
  - Conjunction of clauses
- Term
  - Conjunction of literals
- Theory in disjunctive normal form (DNF)
  - Disjunction of terms
- Theory or knowledge base
  - Conjunction of formulas
- A variable is bound
  - If it is quantified by an enclosing intensional conjunction or disjunction
- A variable is free
  - If it is not bound
- Sentence
  - Formula without free variables
- Formula is ground
  - If it does not contain any variables

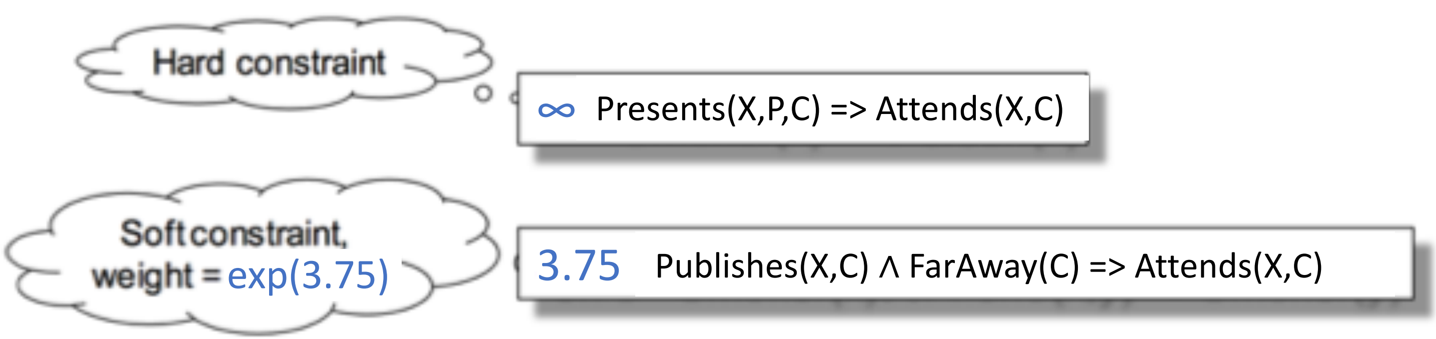
Markov Logic Networks (MLNs)

- Weighted formulas for modelling constraints ($\neq$ domain constraints)
- An MLN is a set of constraints $\Psi = \{(w_i, \psi_i)\}_{i=1}^{n}$
  - $w_i \in \mathbb{R}^+$ weight
  - ψ_i FOL formula
    - Originally without a constraint language for domains
    - Mostly used without quantifiers or implicit all-quantifiers
  - Implicitly connected via conjunction
    - I.e., set of formulas ψ_i = knowledge base/theory



Intuition behind Weights

- Soften logic using weights
  - Worlds that violate constraint become less likely but not impossible
    - As w_i increases, so does the strength of ψ_i
    - Infinite weight: Hard constraint = pure logic formula
      - Probabilities of worlds that do not satisfy hard constraint set to 0
    - Standard MLNs only use weights in $\mathbb{R}$; otherwise semantics break (special handling of hard constraints required)
      - E.g., use hard constraints to filter worlds, probabilities of remaining worlds based on soft constraints



Standard Weights

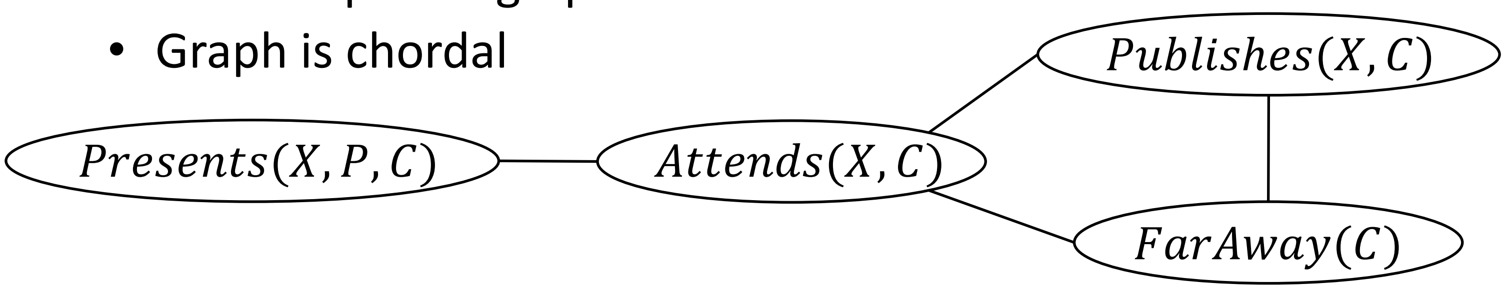
- Example of a standard MLN without infinite weights and not quantifiers
  - All formulas as soft constraints

10 Presents(X,P,C) => Attends(X,C)

3.75 Publishes(X,C) ∧ FarAway(C) => Attends(X,C)

MLN: Graphical Representation?

- Usually not depicted by a graph
- One approach
  - Logical atoms as nodes
  - Edges between nodes whenever atoms occur together in a formula
    - Form cliques in graph
    - Graph is chordal

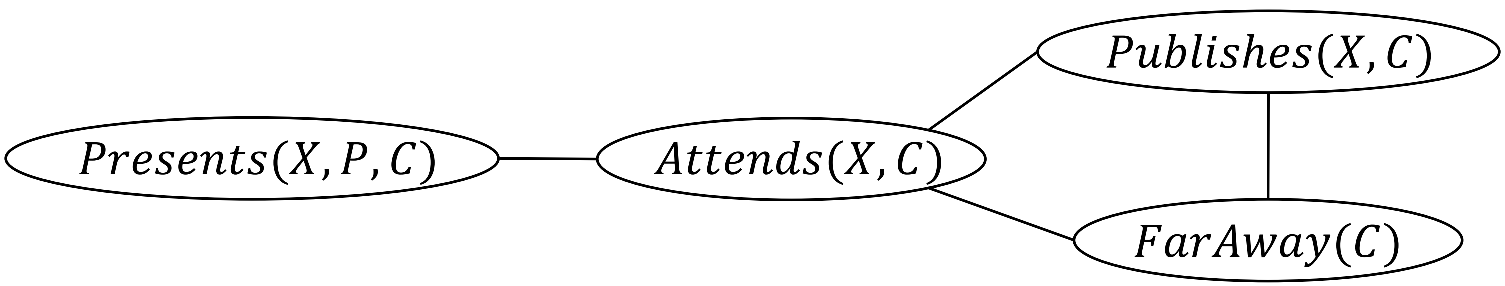


10 Presents(X,P,C) => Attends(X,C)

3.75 Publishes(X,C) ∧ FarAway(C) => Attends(X,C)

MLN: Graphical Representation?

- Potential functions
  - Map to $\exp(w_i)$ if assignment to atoms makes ψ_i true
  - Otherwise map to $\exp(0) = 1$
  - If indeed $w_i = \infty$: choose large number
    - In implementation: maximum number possible in encoding



10 Presents(X,P,C) => Attends(X,C)

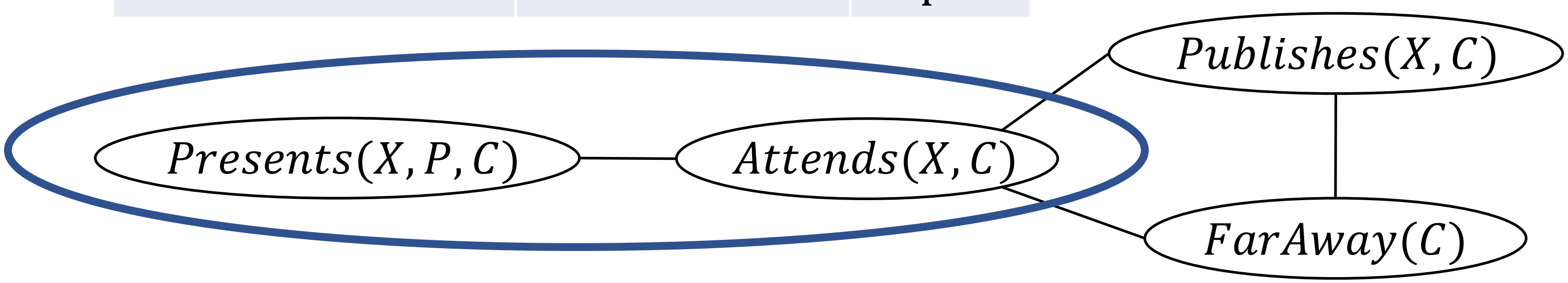
3.75 Publishes(X,C) ∧ FarAway(C) => Attends(X,C)

UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR INFORMATIONSSYSTEME

MLN: Graphical Representation?

- Potential functions

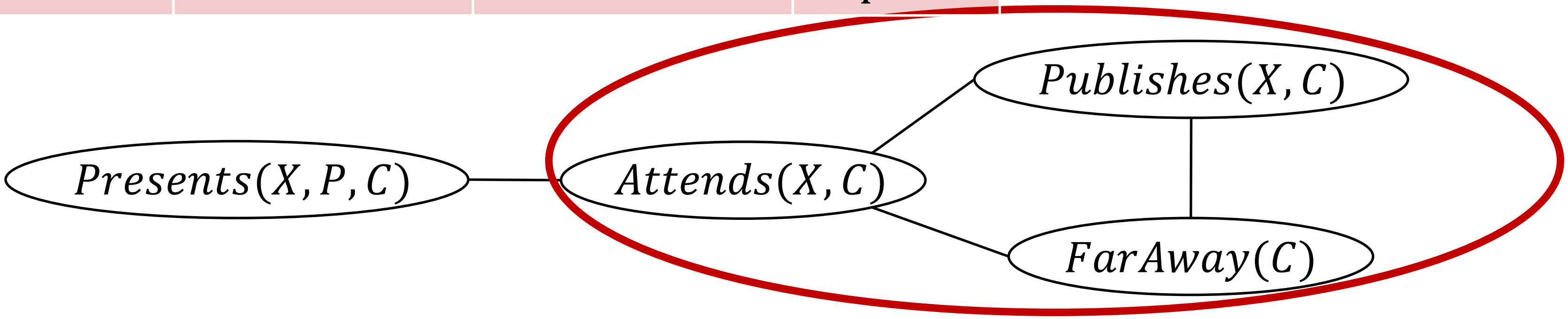
$Presents(X,P,C)$	$Attends(X,C)$	ϕ
$false$	$false$	$\exp 10$
$false$	$true$	$\exp 10$
$true$	$false$	$\exp 0$
$true$	$true$	$\exp 10$



10 Presents(X,P,C) => Attends(X,C)

3.75 Publishes(X,C) ∧ FarAway(C) => Attends(X,C)

$Publishes(X,C)$	$FarAway(C)$	$Attends(X,C)$	ϕ
$false$	$false$	$false$	exp 3.75
$false$	$false$	$true$	exp 3.75
$false$	$true$	$false$	exp 3.75
$false$	$true$	$true$	exp 3.75
$true$	$false$	$false$	exp 3.75
$true$	$false$	$true$	exp 3.75
$true$	$true$	$false$	exp 0
$true$	$true$	$true$	exp 3.75



10 Presents(X,P,C) => Attends(X,C)

3.75 Publishes(X,C) ∧ FarAway(C) => Attends(X,C)

Instances

- Each (w_i, ψ_i) represents a set of sentences
  - One sentence for each possible substitution of the free variables $free(\psi_i)$ in ψ_i given a finite domain (or a constraint set) D over $free(\psi_i)$
    - $\theta_D = \bigcup_{\boldsymbol{d} \in D} \{\bigcup_{t \in \boldsymbol{d}} \{X_d \rightarrow t\}\}$
    - Sentences that a domain yields called instances
  - E.g., $D = \mathcal{D}(X) \times \mathcal{D}(P) \times \mathcal{D}(C)$
    - $\mathcal{D}(X) = \{alice\}, \mathcal{D}(P) = \{p_1, p_2\}, \mathcal{D}(C) = \{ijcai, ki\}$
    - Represents instances
      - $\left(10, Presents(alice, p_1, ijcai) \Rightarrow Attends(alice, ijcai)\right)$
      - $\left(10, Presents(alice, p_1, ki) \Rightarrow Attends(alice, ki)\right)$
      - $\left(10, Presents(alice, p_2, ijcai) \Rightarrow Attends(alice, ijcai)\right)$
      - $\left(10, Presents(alice, p_2, ki) \Rightarrow Attends(alice, ki)\right)$

10 Presents(X,P,C) => Attends(X,C)

MLN to MN

- MLN $\Psi = \{(w_i, \psi_i)\}_{i=1}^{n}$ induces an MN
  - Given finite domain (set of constants)
  - Nodes = random variables = ground atoms
  - Edges connect literals that appear in the same instance of a formula
  - Potential functions (as before, now with instances)
    - Map to $\exp(w_i)$ if assignment to ground atoms makes ψ_i true
    - Otherwise map to $\exp(0) = 1$
    - If indeed $w_i = \infty$: choose large number
      - In implementation: maximum number possible in number format

10 Presents(X,P,C) => Attends(X,C)

3.75 Publishes(X,C) ∧ FarAway(C) => Attends(X,C)

MLN to MN: Example

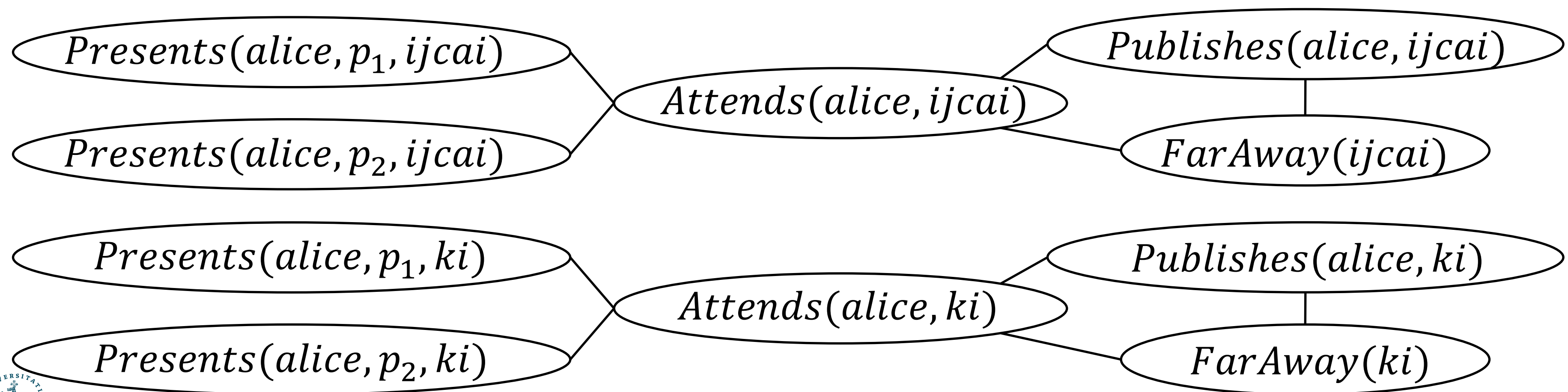
- MLN $\Psi = \{(w_i, \psi_i)\}_{i=1}^{2}$
  - $\mathcal{D}(X) = \{alice\}, \mathcal{D}(P) = \{p_1, p_2\}, \mathcal{D}(C) = \{ijcai, ki\}$
  - Instances
    - $(10, Presents(alice, p_1, ijcai) \Rightarrow Attends(alice, ijcai))$
    - $(10, Presents(alice, p_1, ki) \Rightarrow Attends(alice, ki))$
    - $(10, Presents(alice, p_2, ijcai) \Rightarrow Attends(alice, ijcai))$
    - $(10, Presents(alice, p_2, ki) \Rightarrow Attends(alice, ki))$
    - $(3.75, Publishes(alice, ijcai) \wedge FarAway(ijcai) \Rightarrow Attends(alice, ijcai))$
    - $(3.75, Publishes(alice, ki) \wedge FarAway(ki) \Rightarrow Attends(alice, ki))$

10 Presents(X,P,C) => Attends(X,C)

3.75 Publishes(X,C) ∧ FarAway(C) => Attends(X,C)

MLN to MN: Example

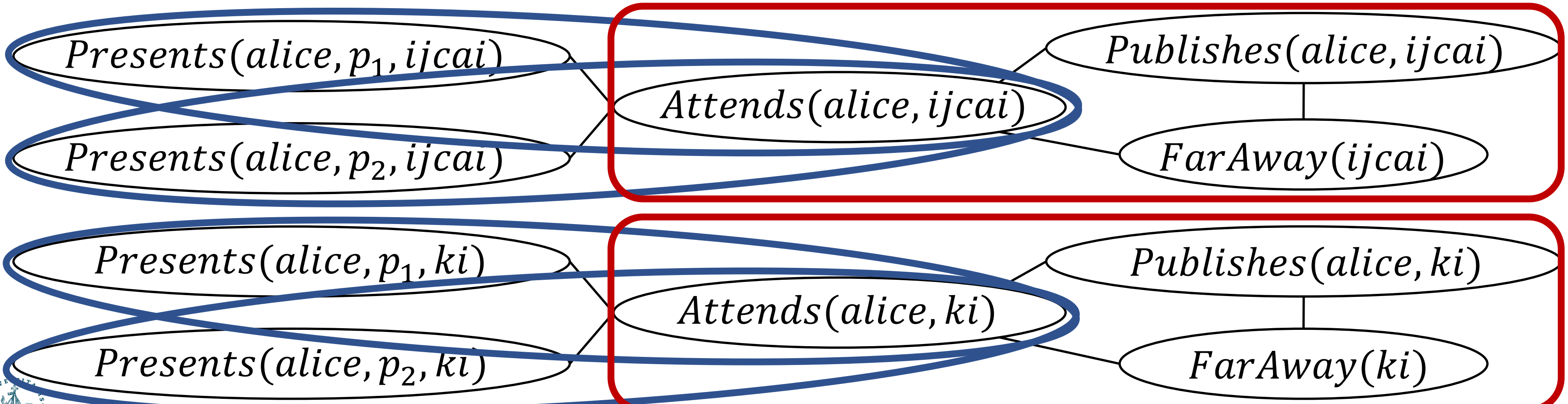
- MLN $\Psi = \{(w_i, \psi_i)\}_{i=1}^{2}$
  - $\mathcal{D}(X) = \{alice\}, \mathcal{D}(P) = \{p_1, p_2\}, \mathcal{D}(C) = \{ijcai, ki\}$
  - Instances
    - $\big(10, Presents(alice, p_1, ijcai) \Rightarrow Attends(alice, ijcai)\big)$
    - $\big(10, Presents(alice, p_1, ki) \Rightarrow Attends(alice, ki)\big)$
    - $\big(10, Presents(alice, p_2, ijcai) \Rightarrow Attends(alice, ijcai)\big)$
    - $\big(10, Presents(alice, p_2, ki) \Rightarrow Attends(alice, ki)\big)$
    - $\big(3.75, Publishes(alice, ijcai) \wedge FarAway(ijcai) \Rightarrow Attends(alice, ijcai)\big)$
    - $\big(3.75, Publishes(alice, ki) \wedge FarAway(ki) \Rightarrow Attends(alice, ki)\big)$



- Potential functions as defined before per (ground) clique
  - (*Atoms abbreviated for space reasons*)

$Pres(x,p,c)$	$Att(x,c)$	ϕ
$false$	$false$	$\exp 10$
$false$	$true$	$\exp 10$
$true$	$false$	$\exp 0$
$true$	$true$	$\exp 10$

$Pub(x,c)$	$FarA(c)$	$Att(x,c)$	ϕ
$false$	$false$	$false$	$\exp 3.75$
$false$	$false$	$true$	$\exp 3.75$
$false$	$true$	$false$	$\exp 3.75$
$false$	$true$	$true$	$\exp 3.75$
$true$	$false$	$false$	$\exp 3.75$
$true$	$false$	$true$	$\exp 3.75$
$true$	$true$	$false$	$\exp 0$
$true$	$true$	$true$	$\exp 3.75$



MLNs: Semantics

- MLN $\Psi = \{(w_i, \psi_i)\}_{i=1}^{n}$, with $w_i \in \mathbb{R}$, induces a probability distribution over possible worlds
$$\omega \in \{true, false\}^N$$
  - N = the number of ground atoms in the grounded Ψ

$$P(\omega) = \frac{1}{Z} \exp\left(\sum_{i=1}^{n} w_i n_i(\omega)\right)$$

  - $n_i(\omega)$ = number of true instances of ψ_i in ω

MLNs: Semantics – Derivation

- Let an MLN $\Psi = \{(w_i, \psi_i)\}_{i=1}^{n}$ and a domain D be given
- The grounded MLN is the following

$$\Psi' = \bigcup_{i=1}^{n} \bigcup_{\theta \in \theta_D} \{(w_i, \psi_i\theta)\}$$

- The semantics of Ψ' induces a probability distribution over possible worlds ω
  - ω assigns a truth value to each (ground) logical atom in Ψ'
  - I.e., a normalised product over all formulas (cliques) with $\exp w_j$ if ω makes $\psi_j\theta$ true and $\exp 0$ otherwise, i.e.,

$$P(\omega) = \frac{1}{Z} \prod_{j=1}^{|\Psi'|} \exp(w_j) = \frac{1}{Z} weight(\omega)$$

    - product called weight of a world

MLNs: Semantics – Derivation

- Consider the weight of ω

$$weight(\omega) = \prod_{j=1}^{|\Psi'|} \exp(w_j)$$

- To simplify, use the following:
  - Formulas in Ψ' are instances of n formulas where each set of instances per formula carries the same weight
  - If an instance is false, its contribution to the product is $\exp(0) = 1$.
- We can rewrite the expression as

$$weight(\omega) = \prod_{i=1}^{n} \exp(w_i)^{n_i(\omega)} = \prod_{i=1}^{n} \exp\left(w_i \cdot n_i(\omega)\right)$$

  - $n_i(\omega)$ = number of true instances of ψ_i in ω

MLNs: Semantics – Derivation

- Therefore, MLN $\Psi = \{(w_i, \psi_i)\}_{i=1}^{n}$ induces a probability distribution over possible worlds ω

$$P(\omega) = \frac{1}{Z}\, weight(\omega)$$
$$= \frac{1}{Z} \prod_{i=1}^{n} \exp\big(w_i \cdot n_i(\omega)\big)$$
$$= \frac{1}{Z} \exp\left(\sum_{i=1}^{n} w_i n_i(\omega)\right)$$

- $Z = \sum_{\omega \in \{true, false\}^N} weight(\omega)$
  - N = the number of ground atoms in Ψ'

Why exp?

- Weight of a world ω:

$$weight(\omega) = \exp\left(\sum_{i=1}^{n} w_i n_i(\omega)\right)$$

- Taking the logarithm yields

$$lweight(\omega) = \ln \exp\left(\sum_{i=1}^{n} w_i n_i(\omega)\right) = \sum_{i=1}^{n} w_i n_i(\omega)$$

  - Sum allows for component-wise optimisation during weight learning
  - Referred to as log-linear models
- Semantics:

$$P(\omega) = \frac{1}{Z'} \sum_{i=1}^{n} w_i n_i(\omega)$$

  - $Z' = \sum_{\omega \in \{true, false\}^N} lweight(\omega) = \sum_{\omega \in \{true, false\}^N} \sum_{i=1}^{n} w_i n_i(\omega)$

Transforming MLNs into PMs …

- Follows the same idea of generating a graphical representation
  - Logical atoms = PRVs $\mathcal{A}$
  - Potential function
    - Map a to $\exp(w_i)$ if truth values in a as assignments to atoms makes ψ_i true
    - Otherwise map to $\exp(0) = 1$
    - If indeed $w_i = \infty$: choose large number
      - In implementation: maximum number possible in encoding

10 Presents(X,P,C) => Attends(X,C)

3.75 Publishes(X,C) ∧ FarAway(C) => Attends(X,C)

Transforming MLNs into PMs …

- Follows the same idea of generating a graphical representation
  - Constraints
    - If no domain constraints given: $(\mathcal{X}, C_{\mathcal{X}}) = \top$, i.e.,

$$\mathcal{X} = lv(\mathcal{A}), C_{\mathcal{X}} = \times_{X \in \mathcal{X}} \mathcal{D}(X)$$

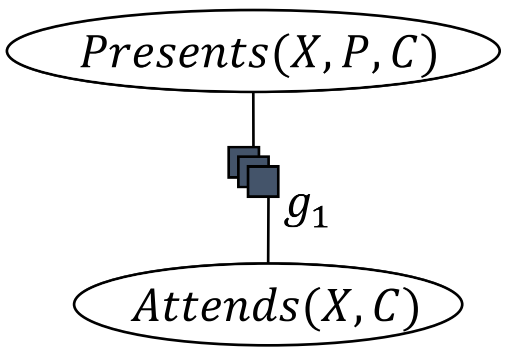
    - Otherwise: build constraint according to domain constraints

10 Presents(X,P,C) => Attends(X,C)

3.75 Publishes(X,C) ∧ FarAway(C) => Attends(X,C)

Transformation Example

- E.g.,
  - $g_1 = \phi\big(Presents(X,P,C), Attends(X,C)\big)_{|C_1}$
    - $C_1 = \big((X,P,C), \mathcal{D}(X) \times \mathcal{D}(P) \times \mathcal{D}(C)\big)$

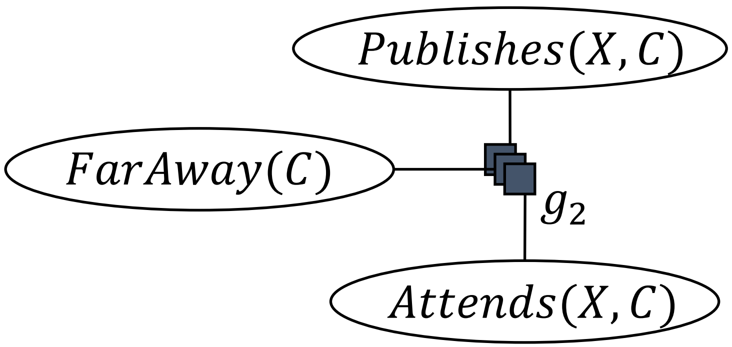


$Presents(X,P,C)$	$Attends(X,C)$	ϕ
$false$	$false$	$\exp 10$
$false$	$true$	$\exp 10$
$true$	$false$	1
$true$	$true$	$\exp 10$

10 Presents(X,P,C) => Attends(X,C)

3.75 Publishes(X,C) ∧ FarAway(C) => Attends(X,C)

- $g_2 = \phi(Publishes(X,C), FarAway(C), Attends(X,C))_{|C_2}$
  - $C_2 = ((X,C), \mathcal{D}(X) \times \mathcal{D}(C))$



$Pub(X,C)$	$FarA(C)$	$Att(X,C)$	ϕ
$false$	$false$	$false$	exp 3.75
$false$	$false$	$true$	exp 3.75
$false$	$true$	$false$	exp 3.75
$false$	$true$	$true$	exp 3.75
$true$	$false$	$false$	exp 3.75
$true$	$false$	$true$	exp 3.75
$true$	$true$	$false$	1
$true$	$true$	$true$	exp 3.75

10 Presents(X,P,C) => Attends(X,C)

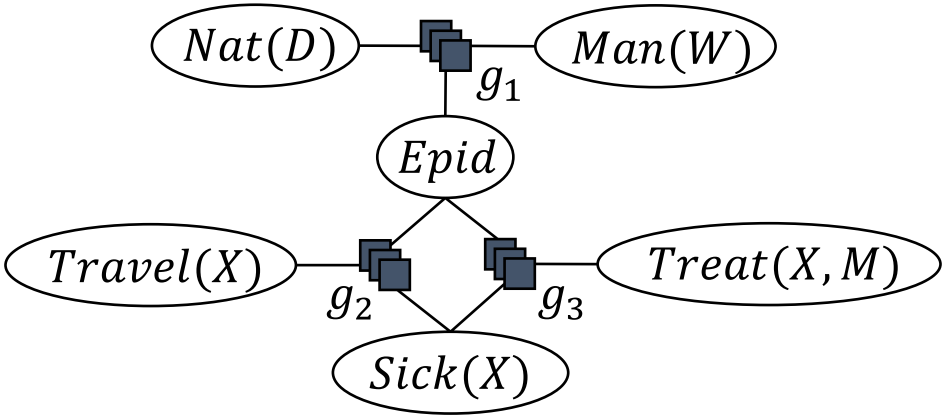
3.75 Publishes(X,C) ∧ FarAway(C) => Attends(X,C)

Transforming PMs into MLNs

- Inverse of transforming MLNs into PMs
- Given $G = \{\phi_i(\mathcal{A})_{|C_i}\}_{i=1}^{n}$
  - All $A \in rv(G)$ need to have Boolean ranges
    - Otherwise: build multiple models where $true$ takes on one range value and $false$ combines all the remaining range values (add up potentials in parfactors)
  - PRVs $\mathcal{A}$ = logical atoms
  - Potential function ϕ_i
    - For each input-output mapping $a \to p$
      - Build formula $\psi = \bigwedge_{A \in \mathcal{A}} t(A, a)$ with weight $w = \ln p$
        - $t(A,a) = \begin{cases} a & \text{Assignment of } A \text{ in } a \text{ is } true \\ \neg a & \text{Assignment of } A \text{ in } a \text{ is } false \end{cases}$
        - If the potentials are in $[0,1]$, then w is negative
          - If potential is 0, choose large negative number
            - Compare to the inverse with a weight of infinity
        - Best: ensure that potentials are in either $(0,1]$ or $\mathbb{R}_{\geq 1}$
    - If some potentials appear multiple times, use an algorithm to identify formulas needed to encode the same information
      - E.g., Quine-McCluskey algorithm
  - Constraint C_i
    - If $C_i = \top$: original MLN without further restrictions
    - Otherwise: encode in constraint set per formula

Inverse Example

- E.g., $G = \{g_i\}_{i=1}^{3}$
  - $g_1 = \phi\big(Epid, Nat(D), Man(W)\big)$
  - $g_2 = \phi\big(Travel(X), Epid, Sick(X)\big)$
  - $g_3 = \phi\big(Epid, Sick(X), Treat(X, M)\big)$
  - Logical atoms:
    $Epid, Nat(D), Man(W),$
    $Travel(X), Sick(X),$
    $Treat(X, M)$
  - Domains as given for G



Inverse Example

- Consider
  $g_2 = \phi\big(Travel(X), Epid, Sick(X)\big)$
  as given by the table on the right
- Weighted formulas (w, ψ)
  - $\big(\ln 5\,, \neg travel(X) \land \neg epid \land \neg sick(X)\big)$
  - $\big(\ln 1\,, \neg travel(X) \land \neg epid \land sick(X)\big)$
  - $\big(\ln 4\,, \neg travel(X) \land epid \land \neg sick(X)\big)$
  - $\big(\ln 6\,, \neg travel(X) \land epid \land sick(X)\big)$
  - $\big(\ln 4\,, travel(X) \land \neg epid \land \neg sick(X)\big)$
  - $\big(\ln 6\,, travel(X) \land \neg epid \land sick(X)\big)$
  - $\big(\ln 2\,, travel(X) \land epid \land \neg sick(X)\big)$
  - $\big(\ln 9\,, travel(X) \land epid \land sick(X)\big)$
- Same has to be done for g_1, g_3

$Travel(X)$	$Epid$	$Sick(X)$	g_2
$false$	$false$	$false$	5
$false$	$false$	$true$	1
$false$	$true$	$false$	4
$false$	$true$	$true$	6
$true$	$false$	$false$	4
$true$	$false$	$true$	6
$true$	$true$	$false$	2
$true$	$true$	$true$	9

Inverse Example

- Consider another potential function
  $$\phi(Travel(X), Epid, Sick(X))$$
  as given by the table on the right
- Only two weighted formulas (w, ψ) necessary
  - $(\ln 2, \neg travel(X) \vee \neg epid \vee \neg sick(X))$
  - $(\ln 7, travel(X) \wedge epid \wedge sick(X))$
  - If potential of 1 instead of 2, would reduce to
    - $(\ln 7, travel(X) \wedge epid \wedge sick(X))$
    - assignments that do not make the formula true automatically get weight of $0 = \ln 1$

$Travel(X)$	$Epid$	$Sick(X)$	ϕ
$false$	$false$	$false$	2
$false$	$false$	$true$	2
$false$	$true$	$false$	2
$false$	$true$	$true$	2
$true$	$false$	$false$	2
$true$	$false$	$true$	2
$true$	$true$	$false$	2
$true$	$true$	$true$	7

Size of resulting MLN depends on the *local symmetries* in the potentials!

MLNs vs. PMs

- MLNs form rules expressed in FOL-DC
  - Requires knowing the interaction to build the formulas
  - Allows humans to interpret a knowledge base more easily
  - Space-efficient encoding if different truth value assignments have the same weight *and* one rule allows for encoding the assignments
- PMs are more blurred
  - Interaction described via distributions
    - Does not enforce connectives between variables
    - But: Interpretation not that easy
- Possible to start with PMs
  - Extract rules if necessary

Interim Summary

- MLNs
  - Weighted formulas
  - Semantics again over groundings and full joint distribution
  - Log-linear version allows for local optimisation during learning
- Transformations between MLNs and PMs
  - One weighted rule → one potential function
  - One potential function → possibly # of input-output mappings different rules
  - If local symmetries exist (in potentials or weights), then MLNs offer compact encoding

Other Formalisms*

* Relations not necessarily used for efficient inference (point of the next section)

- Relational BNs
  - Nodes in BNs are n-ary relations
    - Labelled with so-called probability formulas, which define the semantics as a probability measure over interpretations of the relations
  - Implementation: http://people.cs.aau.dk/~jaeger/Primula/
- Hinge-loss MNs
  - Continuous variables in the $[0,1]$ unit interval combined in constraints of first-order logic syntax
    - Distance of each constraint to satisfaction = hinge loss
  - Implemented with Probabilistic Soft Logic https://psl.linqs.org
- ProbLog: Probabilistic Prolog
  - Defines a probability distribution over logic programs
  - Typically Horn clauses annotated with probabilities
  - Implementation: https://dtai.cs.kuleuven.be/problog/

Manfred Jaeger: Relational Bayesian Networks. In: *UAI-97 Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence*, 1997.
Stephen H. Bach, Matthias Broecheler, Bert Huang, and Lise Getoor: Hinge-loss Markov Random Fields and Probabilistic Soft Logic. In: *Journal of Machine Learning Research*, 2017
Luc De Raedt, Angelika Kimmig, and Hannu Toivonen: ProbLog: A Probabilistic Prolog and its Application in Link Discovery. In: IJCAI-07 Proceedings of the 20th International Joint Conference on Artificial Intelligence, 2007.

Outline: 2. Probabilistic relational models

A. Parameterised models (PMs)
- Motivation: Symmetries and relations
- Syntax, semantics
- Graphical representation
- Inference tasks

B. Markov logic networks (MLNs)
- Syntax of first-order logic
- Syntax, semantics of MLNs
- Graphical representation
- Turning MLNs into PMs and vice versa

$\Longrightarrow$ Next: Lifted inference