Fundamental Problems in Statistical Relational AI

Tutorial - KR 2024

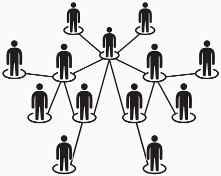
Sagar Malhotra
TU Wien, Austria
Machine Learning Research Unit

Table of Contents

StaRAI

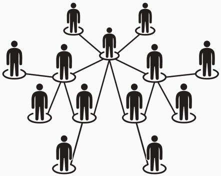
What are Relational Domains?

Social Networks

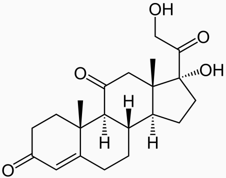


What are Relational Domains?

Social Networks

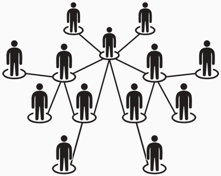


Molecules

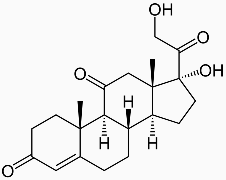


What are Relational Domains?

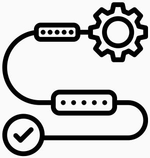
Social Networks



Molecules

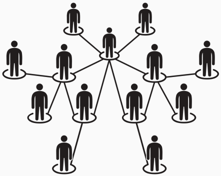


Business Processes



Modeling Relational Domains in First Order Logic

Social Networks



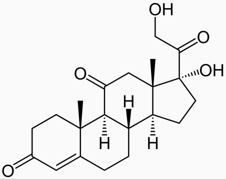
$\texttt{Freinds}(x, y)$

$\texttt{Smokes}(x)$

$\texttt{Cancer}(x)$

$\vdots$

Molecules



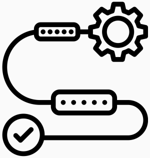
$\texttt{Bond}(x, y)$

$\texttt{Carbon}(x)$

$\texttt{Oxygen}(x)$

$\vdots$

Business Processes



$\texttt{Teaches}(x, y)$

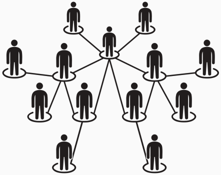
$\texttt{Professor}(x)$

$\texttt{Student}(x)$

$\vdots$

Modeling Knowledge in First Order Logic

Social Networks

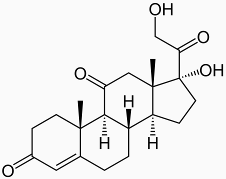


$\forall xy.\mathtt{Fr}(x,y) \rightarrow \mathtt{Fr}(y,x)$

$\forall xy.\mathtt{Sm}(x) \wedge \mathtt{Fr}(x,y) \rightarrow \mathtt{Sm}(y)$

$\vdots$

Molecules

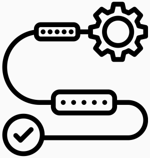


$\forall x.\mathtt{H}(x) \rightarrow \exists y.\mathtt{Bond}(x,y)$

$\forall x.\mathtt{O}(x) \rightarrow \exists^{\leq 2} y.\mathtt{Bond}(x,y)$

$\vdots$

Business Processes



$\forall xy.\mathtt{Prof}(x) \wedge \mathtt{Tch}(x,y) \rightarrow \mathtt{Stud}(y)$

$\vdots$

Modelling Relational Domains in First Order Logic

PROBLEMS![1]

- **Laziness**: Too much work to list out all rules!

[1]Russell and Norvig. Artificial Intelligence: A Modern Approach

Modelling Relational Domains in First Order Logic

PROBLEMS![1]

- **Laziness**: Too much work to list out all rules!
- **Theoretical Ignorance**: We don't have all the rules!

[1] Russell and Norvig. Artificial Intelligence: A Modern Approach

Modelling Relational Domains in First Order Logic

PROBLEMS![1]

- **Laziness**: Too much work to list out all rules!
- **Theoretical Ignorance**: We don't have all the rules!
- **Practical Ignorance**: Maybe there are no rules — inherent stochasticity!

[1] Russell and Norvig. Artificial Intelligence: A Modern Approach

Statistical Relational AI

Statistical Relational AI

=

Logic + Probability

Ingredients of Statistical Relational Learning

SRL ingredients:

- A set of Herbrand models $\Omega^{(n)}$ in function-free First Order Logic
- A parametric probability distribution $\mathbb{P}_\theta : \Omega \to [0, 1]$

Ingredients of Statistical Relational Learning

SRL ingredients:

- A set of Herbrand models $\Omega^{(n)}$ in function-free First Order Logic
- A parametric probability distribution $\mathbb{P}_\theta : \Omega \rightarrow [0, 1]$

The Inference Problem:

- $\mathbb{P}_\theta(q) = \sum_{\omega \models q} \mathbb{P}_\theta(\omega)$

Ingredients of Statistical Relational Learning

SRL ingredients:

- A set of Herbrand models $\Omega^{(n)}$ in function-free First Order Logic
- A parametric probability distribution $\mathbb{P}_\theta : \Omega \rightarrow [0, 1]$

The Inference Problem:

- $\mathbb{P}_\theta(q) = \sum_{\omega \models q} \mathbb{P}_\theta(\omega)$

The Learning Problem:

- $\theta^* = \text{argmax}_\theta \, \mathbb{P}_\theta(\omega^*)$

Fundamental Problems in SRL

"What are the fundamental problems in Statistical Relational Learning?"

Fundamental Problems in SRL

"What are the fundamental problems in Statistical Relational Learning?"

The Inference Problem:

- $\mathbb{P}_\theta(q) = \sum_{\omega \models q} \mathbb{P}_\theta(\omega)$ **Intractability**

Fundamental Problems in SRL

"What are the fundamental problems in Statistical Relational Learning?"

The Inference Problem:

- $\mathbb{P}_\theta(q) = \sum_{\omega \models q} \mathbb{P}_\theta(\omega)$ **Intractability**

The Learning Problem:

- $\theta^* = \mathrm{argmax}_\theta \, \mathbb{P}_\theta(\omega^*)$ **No Consistent Estimation**

Tractability

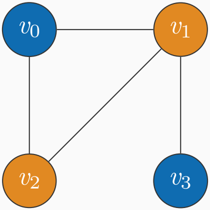
Markov Logic[2]: An Example

A contact-tracing model:

$$w_1 : \texttt{Covid}(\texttt{x})$$
$$w_2 : \texttt{Covid}(\texttt{x}) \wedge \texttt{Contact}(\texttt{x}, \texttt{y}) \rightarrow \texttt{Covid}(\texttt{y})$$

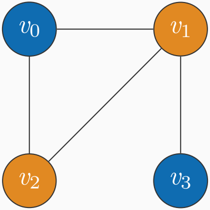
[2] Richardson and Domingos. Markov Logic Networks. 2006

MLNs: An Example



- $\texttt{Covid(x)}$
  $n_1(\omega) = |\{v_i : \omega \models \texttt{Covid}(\texttt{v}_\texttt{i})\}|$

MLNs — An Example



- $\mathtt{Covid}(\mathtt{x}) \land \mathtt{Contact}(\mathtt{x}, \mathtt{y}) \to \mathtt{Covid}(\mathtt{y})$
  $n_2(\omega) = |\{(v_i, v_j) : \omega \models \phi_2(\mathtt{v_i}, \mathtt{v_j})\}|$

Markov Logic Networks: Almost Formally

- Weighted quantifier-free first-order logic formulas:

$$\{w_i : \phi_i\}$$

Markov Logic Networks: Almost Formally

- Weighted quantifier-free first-order logic formulas:

$$\{w_i : \phi_i\}$$

- Probability distribution on all finite structures of size n:

$$\texttt{Pr}(\omega) := \frac{1}{Z} \exp\left(\sum_i w_i n_i(\omega)\right) \tag{1}$$

Markov Logic Networks: Almost Formally

- Weighted quantifier-free first-order logic formulas:

$$\{w_i : \phi_i\}$$

- Probability distribution on all finite structures of size n:

$$\mathtt{Pr}(\omega) := \frac{1}{Z} \exp\left(\sum_i w_i n_i(\omega)\right) \tag{1}$$

- The partition function — main source of intractability:

$$Z := \sum_{\omega \models \Phi_\infty} \exp\left(\sum_i w_i n_i(\omega)\right) \tag{2}$$

Weighted First Order Model Counting

Symmetric Weighted First Order Model Counting (WFOMC):

$$\text{WFOMC}(\Phi, n) := \sum_{\omega \models \Phi} \mathrm{w}(\omega)$$

Weighted First Order Model Counting

Symmetric Weighted First Order Model Counting (WFOMC):

$$\mathsf{WFOMC}(\Phi, n) := \sum_{\omega \models \Phi} \mathrm{w}(\omega)$$

$$\mathrm{w}(\omega) = \prod_{\substack{\omega \models g \\ g \in \mathcal{G}}} w(pred(g)) \prod_{\substack{\omega \models \neg g \\ g \in \mathcal{G}}} \bar{w}(pred(g)).$$

Weighted First Order Model Counting

Symmetric Weighted First Order Model Counting (WFOMC):

$$\mathsf{WFOMC}(\Phi, n) := \sum_{\omega \models \Phi} \mathrm{w}(\omega)$$

$$\mathrm{w}(\omega) = \prod_{\substack{\omega \models g \\ g \in \mathcal{G}}} w(pred(g)) \prod_{\substack{\omega \models \neg g \\ g \in \mathcal{G}}} \bar{w}(pred(g)).$$

WFOMC is **Tractable** if it can be computed in polynomial time w.r.t the **domain size (n)**.

WFOMC: An Example

$$\forall \mathsf{x}.\mathsf{KR}(\mathsf{x}) \rightarrow \mathtt{Intelligent}(\mathsf{x})$$

$$w(\mathsf{KR}) = 2.5 \qquad \bar{w}(\mathsf{KR}) = 0.5$$

$$w(\mathtt{Intelligent}) = 0.5 \qquad \bar{w}(\mathtt{Intelligent}) = 1.5$$

KR(a)	Intelligent(a)	w(ω)
1	1	$2.5 \times 0.5 = 1.25$
1	0	$2.5 \times 1.5 = 1.25$
0	1	$5 \times 0.5 = 1.5$
0	0	$5 \times 1.5 = 7.5$

$$\text{WFOMC}(\Phi, 1) = 1.25 + 1.5 + 7.5$$

WFOMC $\equiv$ Partition Function

$$Z := \sum_{\omega \models \Phi_\infty} \exp\left(\sum_i w_i n_i(\omega)\right)$$

$$\Phi_\infty \wedge \bigwedge_i \forall FV[\phi_i].\big(R_i(FV[\phi_i]) \leftrightarrow \phi_i\big) \tag{3}$$

$$w(R_i) = \exp(w_i) \qquad \bar{w}(R_i) = 1$$
$$w(*) = 1 \qquad \bar{w}(*) = 1$$

WFOMC $\equiv$ Partition Function: An Example

$$w : \mathsf{KR}(\mathsf{x}) \rightarrow \mathtt{Intelligent}(\mathsf{x})$$

WFOMC $\equiv$ Partition Function: An Example

$$w : \mathsf{KR}(\mathsf{x}) \rightarrow \mathtt{Intelligent}(\mathsf{x})$$

WFOMC encoding for the partition function:

$$\forall \mathsf{x}.\mathsf{R}(\mathsf{x}) \leftrightarrow \left(\mathsf{KR}(\mathsf{x}) \rightarrow \mathtt{Intelligent}(\mathsf{x})\right)$$

$$w(R) = \exp(w) \qquad w(*) = 1 \qquad \bar{w}(*) = 1$$

How Intractable (practically)?

WFOMC is a #P complete problem in general[3].

[3] Beame, Van den Broeck, Gribkoff, Suciu. PODS 2015.

How Intractable (practically)?

WFOMC is a #P complete problem in general[3].

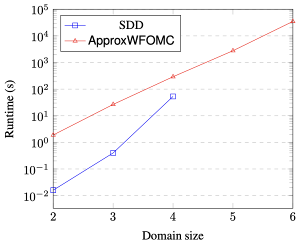
But is WMC not practically quite scalable?

[3] Beame, Van den Broeck, Gribkoff, Suciu. PODS 2015.

How Intractable?

$$w_1 : \texttt{Stress}(\texttt{x}) \rightarrow \texttt{Smokes}(\texttt{x})$$
$$w_2 : \texttt{Smokes}(\texttt{x}) \land \texttt{Fr}(\texttt{x}, \texttt{y}) \rightarrow \texttt{Smokes}(\texttt{y})$$
$$w_3 : \texttt{Fr}(\texttt{x}, \texttt{y}) \land \texttt{Fr}(\texttt{y}, \texttt{z}) \rightarrow \texttt{Fr}(\texttt{x}, \texttt{z})$$

HOW INTRACTABLE[4]?



(a) Runtime of various WFOMC methods for the `transitive-smokers-mln` problem for various domain sizes.

Figure 1: 10000 seconds is more than 2 hours

[4] Bremen and Kuzelka. IJCAI 2020

WFOMC: Fundamental Problems

What fragments of first-order logic admit tractable WFOMC?

[5] Beame, Van den Broeck, Gribkoff and Suciu. PODS 2015.
[6] den Broeck et al. KR 2014

WFOMC: Fundamental Problems

What fragments of first-order logic admit tractable WFOMC?

There is an intractable FO^3 formula[5]

[5] Beame, Van den Broeck, Gribkoff and Suciu. PODS 2015.
[6] den Broeck et al. KR 2014

WFOMC: Fundamental Problems

What fragments of first-order logic admit tractable WFOMC?

There is an intractable FO^3 formula[5]

FO^2 is tractable[6]

[5] Beame, Van den Broeck, Gribkoff and Suciu. PODS 2015.
[6] den Broeck et al. KR 2014

FO^2 Language: 1-Types

We have a language with **at most two variables** , with the following predicates:

- A unary predicate $\mathsf{KR}(x)$
- A binary predicate $\mathtt{Shaves}(x, y)$

FO^2 Language: 1-Types

We have a language with **at most two variables** , with the following predicates:

- A unary predicate $\mathsf{KR}(x)$
- A binary predicate $\mathsf{Shaves}(x, y)$

We have the following set of unary properties also called 1-types:

$$\neg\mathsf{KR}(c) \land \neg\mathsf{Shaves}(c, c)$$

$$\neg\mathsf{KR}(c) \land \mathsf{Shaves}(c, c)$$

$$\mathsf{KR}(c) \land \neg\mathsf{Shaves}(c, c)$$

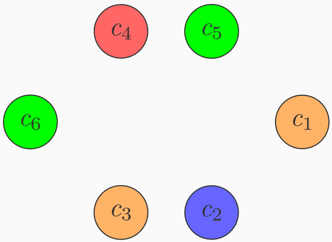
$$\mathsf{KR}(c) \land \mathsf{Shaves}(c, c)$$

1- Type Enumeration

An arrangement of 1-Types

$\neg\mathsf{KR}(\mathsf{c}) \land \neg\mathsf{Shaves}(\mathsf{c},\mathsf{c})$

$\neg\mathsf{KR}(\mathsf{c}) \land \mathsf{Shaves}(\mathsf{c},\mathsf{c})$



$\boldsymbol{k}:\quad k_1 = 1 \quad k_2 = 2 \quad k_3 = 1 \quad k_4 = 2$

$\mathsf{KR}(\mathsf{c}) \land \neg\mathsf{Shaves}(\mathsf{c},\mathsf{c})$

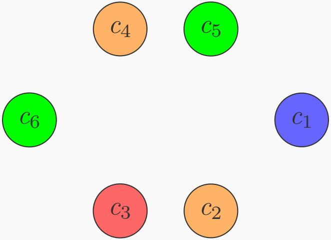
$\mathsf{KR}(\mathsf{c}) \land \mathsf{Shaves}(\mathsf{c},\mathsf{c})$

1- Type Enumeration

Another arrangement of 1-Types

$\neg$KR(c) $\wedge$ $\neg$Shaves(c, c)

$\neg$KR(c) $\wedge$ Shaves(c, c)



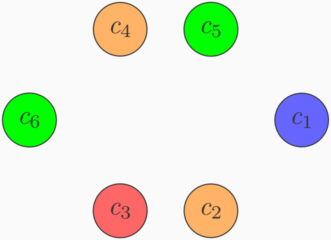
$\boldsymbol{k}:\quad k_1 = 1 \quad k_2 = 2 \quad k_3 = 1 \quad k_4 = 2$

KR(c) $\wedge$ $\neg$Shaves(c, c)

KR(c) $\wedge$ Shaves(c, c)

1-Type Enumeration

Counting for fixed 1-Type cardinalities



$$\boldsymbol{k}: \quad k_1 = 1 \quad k_2 = 2 \quad k_3 = 1 \quad k_4 = 2$$

$$\#\text{Similar Arrangements} = \binom{n}{k_1, k_2, k_3, k_4} = \frac{6!}{1!2!1!2!} = 180$$

FO^2 Language: 2-tables

We have an FO^2 language, with the following predicates:

- A unary predicate $\mathsf{KR}(x)$
- A binary predicate $\mathsf{Shaves}(x, y)$

We have the following set of binary properties also called 2-tables:

$$\mathsf{Shaves}(c, d) \land \mathsf{Shaves}(d, c)$$

$$\neg\mathsf{Shaves}(c, d) \land \mathsf{Shaves}(d, c)$$

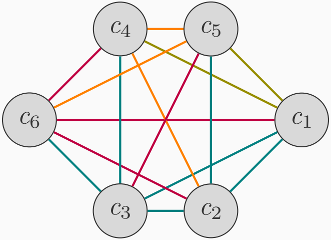
$$\mathsf{Shaves}(c, d) \land \neg\mathsf{Shaves}(d, c)$$

$$\neg\mathsf{Shaves}(c, d) \land \neg\mathsf{Shaves}(d, c)$$

2- Table Enumeration

An arrangement for 2-tables

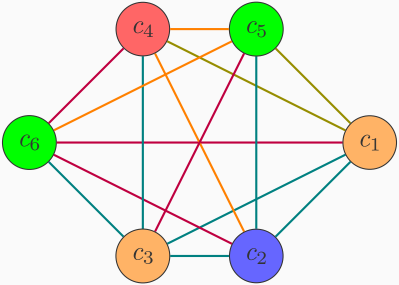
$\mathsf{Shaves}(c, d) \wedge \mathsf{Shaves}(d, c)$ $\neg\mathsf{Shaves}(c, d) \wedge \mathsf{Shaves}(d, c)$



$\mathsf{Shaves}(c, d) \wedge \neg\mathsf{Shaves}(d, c)$ $\neg\mathsf{Shaves}(c, d) \wedge \neg\mathsf{Shaves}(d, c)$

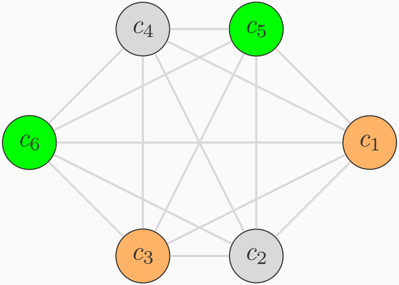
2- Table Enumeration given 1-Types

An arrangement for 2-tables given 1-types



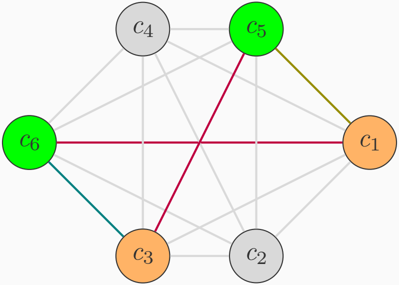
2- Table Enumeration given 1-Types

Pick a pair of 1-Types



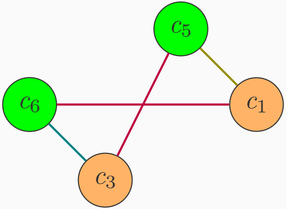
2- Table Enumeration given 1-Types

Picking a sub graph: Pick 2-Tables between them



2- Table Enumeration

Enumerating 2-tables given 1-types

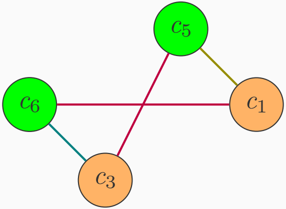


$k_2 = 2 \quad k_4 = 2$

$h_1 = 2 \quad h_2 = 1 \quad h_3 = 1 \quad h_4 = 0$

2- Table Enumeration

Enumerating 2-tables given 1-types



$$k_2 = 2 \quad k_4 = 2$$

$$h_1 = 2 \quad h_2 = 1 \quad h_3 = 1 \quad h_4 = 0$$

$$\binom{k_2 \times k_4}{h_1 \quad h_2 \quad h_3 \quad h_4} = \frac{(2 \times 2)!}{2!2!1!0!} = 6$$

Enumerating all models over 1-types and 2-tables

$$\sum_{\langle \vec{\boldsymbol{k}}, \vec{\boldsymbol{h}} \rangle} \binom{n}{k_1 \ldots k_u} \prod_{1 \leq i \leq j \leq u} \binom{k(i,j)}{h_1^{ij} \ldots h_b^{ij}}$$

$$\boldsymbol{k}(i,j) = \begin{cases} k_i k_j & \text{if } i \neq j \\ \frac{k_i(k_i-1)}{2} & \text{if } i = j \end{cases}$$

Adding Formulas : $\forall xy.\Phi(x, y)$

A formula $\forall xy.\Phi(x, y)$, allows some and disallows other 1-Type and 2-Table configuration. For Example:

$$\neg \mathsf{Shaves}(x, x)$$

$$\mathsf{Shaves}(x, y) \rightarrow \mathsf{Shaves}(y, x)$$

$$\mathsf{KR}(x) \land \mathsf{Shaves}(x, y) \rightarrow \mathsf{KR}(y)$$

Adding Formulas : $\forall xy.\Phi(x, y)$

$$\neg\mathsf{Shaves}(x, x)$$
$$\mathsf{Shaves}(x, y) \rightarrow \mathsf{Shaves}(y, x)$$
$$\mathsf{KR}(x) \wedge \mathsf{Shaves}(x, y) \rightarrow \mathsf{KR}(y)$$

Not allowed:

$$\neg\mathsf{KR}(c) \wedge \neg\mathsf{Sh}(c, c) \wedge \mathsf{KR}(d) \wedge \neg\mathsf{Sh}(d, d) \wedge \mathsf{Sh}(c, d) \wedge \mathsf{Sh}(d, c)$$

Allowed:

$$\mathsf{KR}(c) \wedge \neg\mathsf{Sh}(c, c) \wedge \mathsf{KR}(d) \wedge \neg\mathsf{Sh}(d, d) \wedge \mathsf{Sh}(c, d) \wedge \mathsf{Sh}(d, c)$$

FOMC in FO^2: $\forall x \forall y . \Phi(x, y)$

$$\mathrm{FOMC}(\Phi, n) =$$

$$\sum_{\langle \vec{\boldsymbol{k}}, \vec{\boldsymbol{h}} \rangle} \binom{n}{k_1, \dots, k_u} \prod_{1 \le i \le j \le u} \binom{\boldsymbol{k}(i,j)}{h_1^{ij}, \dots, h_b^{ij}} \prod_{1 \le v \le b} {n_{ijv}}^{h_v^{ij}}$$

Unary Properties

Constraints: Φ

Binary Properties

FOMC in FO^2: $\forall x \forall y. \Phi(x, y)$

$$\mathrm{FOMC}(\Phi, n) =$$

$$\sum_{\langle \vec{\boldsymbol{k}}, \vec{\boldsymbol{h}} \rangle} \binom{n}{k_1, \ldots, k_u} \prod_{1 \le i \le j \le u} \binom{\boldsymbol{k}(i, j)}{h_1^{ij}, \ldots, h_b^{ij}} \prod_{1 \le v \le b} {n_{ijv}}^{h_v^{ij}}$$

$$= \sum_{\langle \vec{\boldsymbol{k}}, \vec{\boldsymbol{h}} \rangle} \mathsf{FOMC}(\Phi, \langle \vec{\boldsymbol{k}}, \vec{\boldsymbol{h}} \rangle)$$

$$= \sum_{\vec{\boldsymbol{k}}} \mathsf{FOMC}(\Phi, \vec{\boldsymbol{k}})$$

Cardinality Constraints

$$\neg \mathsf{Shaves}(x, x)$$

$$\mathsf{Shaves}(x, y) \rightarrow \mathsf{Shaves}(y, x)$$

$$\mathsf{KR}(x) \wedge \mathsf{Shaves}(x, y) \rightarrow \mathsf{KR}(y)$$

$$|\mathsf{KR}(x)| = 2$$

Cardinality Constraints

$$\neg \mathsf{Shaves}(x, x)$$

$$\mathsf{Shaves}(x, y) \rightarrow \mathsf{Shaves}(y, x)$$

$$\mathsf{KR}(x) \wedge \mathsf{Shaves}(x, y) \rightarrow \mathsf{KR}(y)$$

$$|\mathsf{KR}(x)| = 2$$

k_1 : $\neg \mathsf{KR}(c) \wedge \neg \mathsf{Shaves}(c, c)$

k_2 : $\neg \mathsf{KR}(c) \wedge \mathsf{Shaves}(c, c)$

k_3 : $\mathsf{KR}(c) \wedge \neg \mathsf{Shaves}(c, c)$

k_4 : $\mathsf{KR}(c) \wedge \mathsf{Shaves}(c, c)$

Cardinality Constraints

$$\neg \mathsf{Shaves}(x, x)$$
$$\mathsf{Shaves}(x, y) \rightarrow \mathsf{Shaves}(y, x)$$
$$\mathsf{KR}(x) \wedge \mathsf{Shaves}(x, y) \rightarrow \mathsf{KR}(y)$$
$$|\mathsf{KR}(x)| = 2$$

$$k_1 : \neg \mathsf{KR}(c) \wedge \neg \mathsf{Shaves}(c, c)$$
$$k_2 : \neg \mathsf{KR}(c) \wedge \mathsf{Shaves}(c, c)$$
$$k_3 : \mathsf{KR}(c) \wedge \neg \mathsf{Shaves}(c, c)$$
$$k_4 : \mathsf{KR}(c) \wedge \mathsf{Shaves}(c, c)$$

$$\boldsymbol{k} : k_3 + k_4 = 2$$

Cardinality Constraints

Given an arbitrary cardinality constraint ρ:

$$\mathsf{FOMC}(\Phi \wedge \rho) := \sum_{\langle \boldsymbol{k}, \boldsymbol{h} \rangle \models \rho} \mathsf{FOMC}(\Phi, \langle \boldsymbol{k}, \boldsymbol{h} \rangle)$$

Existential Quantifiers [7]

Minimal Non-Trivial Example:

$$\Psi := \Phi \wedge \forall x \exists y. R(x, y)$$

[7] Guy Van den Broeck, Wannes Meert, Adnan Darwiche. KR 2014
Malhotra and Serrafini. AAAI 2022 (for this formulation)

Existential Quantifiers [7]

Minimal Non-Trivial Example:

$$\Psi := \Phi \land \forall x \exists y. R(x, y)$$

Key Proof ideas:

- $A_c = \{\omega | \omega \models \Phi \land \forall y. \neg R(c, y)\} \equiv$ Atleast c is isolated

[7] Guy Van den Broeck, Wannes Meert, Adnan Darwiche. KR 2014
Malhotra and Serrafini. AAAI 2022 (for this formulation)

Existential Quantifiers [7]

Minimal Non-Trivial Example:

$$\Psi := \Phi \land \forall x \exists y . R(x, y)$$

Key Proof ideas:

- $A_c = \{\omega | \omega \models \Phi \land \forall y . \neg R(c, y)\} \equiv$ Atleast c is isolated
- FOMC(Φ) - $|\bigcup_{i \in [n]} A_i|$

[7] Guy Van den Broeck, Wannes Meert, Adnan Darwiche. KR 2014
Malhotra and Serrafini. AAAI 2022 (for this formulation)

Existential Quantifiers [7]

Minimal Non-Trivial Example:

$$\Psi := \Phi \land \forall x \exists y . R(x, y)$$

Key Proof ideas:

- $A_c = \{\omega | \omega \models \Phi \land \forall y . \neg R(c, y)\} \equiv$ Atleast c is isolated
- FOMC(Φ) - $|\bigcup_{i \in [n]} A_i|$
- Use Principle of Inclusion Exclusion to calculate $|\bigcup_{i \in [n]} A_i|$

[7] Guy Van den Broeck, Wannes Meert, Adnan Darwiche. KR 2014
Malhotra and Serrafini. AAAI 2022 (for this formulation)

Principle of Inclusion-Exclusion

- $\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{\emptyset \neq J \subseteq \{1,\ldots,n\}} (-1)^{|J|+1} \left|\bigcap_{j \in J} A_j\right|$

Principle of Inclusion-Exclusion

- $\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{\emptyset \neq J \subseteq \{1,\ldots,n\}} (-1)^{|J|+1} \left|\bigcap_{j \in J} A_j\right|$
- The number of terms involved is $2^n - 1$
- Recall, $A_c = \{\omega | \omega \models \Phi \wedge \forall y. \neg R(c, y)\}$

Principle of Inclusion-Exclusion

- $\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{\emptyset \neq J \subseteq \{1,\ldots,n\}} (-1)^{|J|+1} \left|\bigcap_{j \in J} A_j\right|$
- The number of terms involved is $2^n - 1$
- Recall,$A_c = \{\omega | \omega \models \Phi \wedge \forall y. \neg R(c, y)\}$
- $\bigcap_{j \in J} A_j$ is dependent on $|J|$ and not on $\{J\}$

Principle of Inclusion-Exclusion

- $\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{\emptyset \neq J \subseteq \{1,\ldots,n\}} (-1)^{|J|+1} \left|\bigcap_{j \in J} A_j\right|$
- The number of terms involved is $2^n - 1$
- Recall,$A_c = \{\omega | \omega \models \Phi \wedge \forall y. \neg R(c, y)\}$
- $\bigcap_{j \in J} A_j$ is dependent on $|J|$ and not on $\{J\}$
- Let $\alpha_K = \left|\bigcap_{j \in J} A_j\right|$
- $\alpha_k = |\{\omega | \omega \models \Phi \wedge \forall xy. P(x) \rightarrow \neg R(x, y) \wedge |P| = k\}|$

Principle of Inclusion-Exclusion

Recall,

$$\alpha_k = |\{\omega | \omega \models \Phi \wedge \forall xy.P(x) \rightarrow \neg R(x, y) \wedge |P| = k\}|$$

Principle of Inclusion-Exclusion

Recall,

$$\alpha_k = |\{\omega | \omega \models \Phi \wedge \forall xy.P(x) \rightarrow \neg R(x,y) \wedge |P| = k\}|$$

We wanted:

$$\text{FOMC}(\Phi) - |\bigcup_{i \in [n]} A_i|$$

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} \alpha_k$$

$$\left|U \setminus \bigcup_{i=1}^{n} A_i\right| = \sum_{k=0}^{n} (-1)^{k+1} \binom{n}{k} \alpha_k$$

Existential Quantifier ≡PIE ≡WFOMC with negative weights

$$\text{WFOMC}(\Phi \wedge \forall x \exists y.R(x,y)) \equiv \text{WFOMC}(\Phi \wedge \forall xy.P(x) \rightarrow \neg R(x,y))$$

Existential Quantifier ≡PIE ≡WFOMC with negative weights

$$\text{WFOMC}(\Phi \wedge \forall x \exists y.R(x,y)) \equiv \text{WFOMC}(\Phi \wedge \forall xy.P(x) \rightarrow \neg R(x,y))$$

Scott's Normal Form is equi-satisfiable and WFOMC preserving:

$$\Phi \bigwedge_i \forall x \exists y R_i(x,y)$$

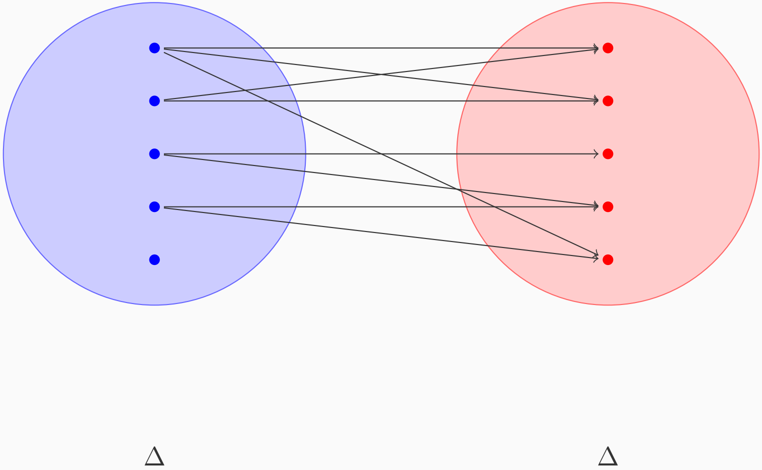
Counting Quantifiers [8]

Counting Quantifiers: $\forall x \exists^{=1} y . R(x, y)$

[8] Kuzelka. JAIR 2021. Functionality: Kuuisto, Lutz. LICS 2018

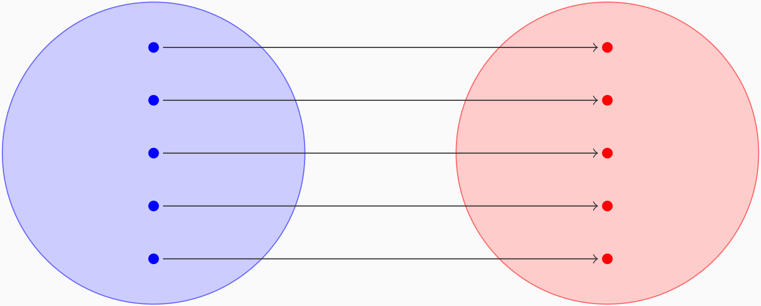
Counting Quantifiers: Expressing Functionality

$$R(x, y) : \Delta \to \Delta$$



Counting Quantifiers: Expressing Functionality

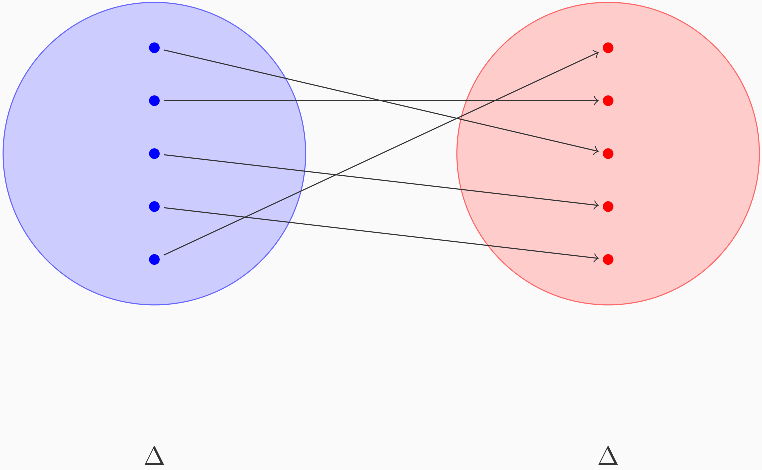
$$\forall x\ \exists^{=1} y.\ R(x, y)$$



Δ Δ

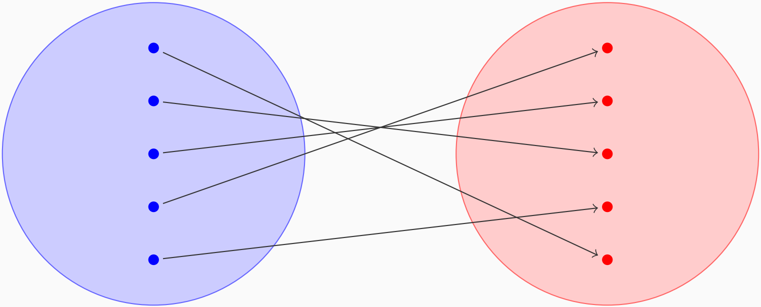
Counting Quantifiers: Expressing Functionality

$$\forall x\ \exists^{=1} y.\ R(x, y)$$



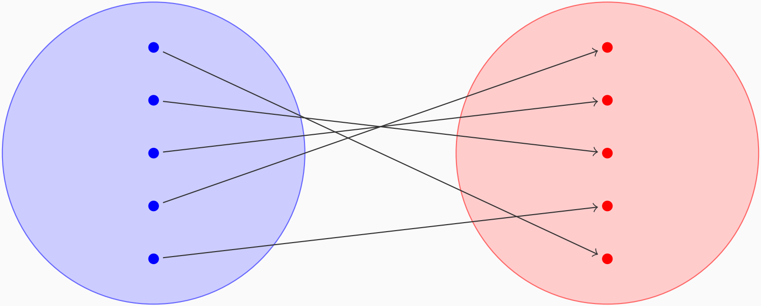
Counting Quantifiers: Expressing Functionality

$$\forall x \; \exists^{=1} y.\; R(x, y)$$



Counting Quantifiers: Expressing Functionality

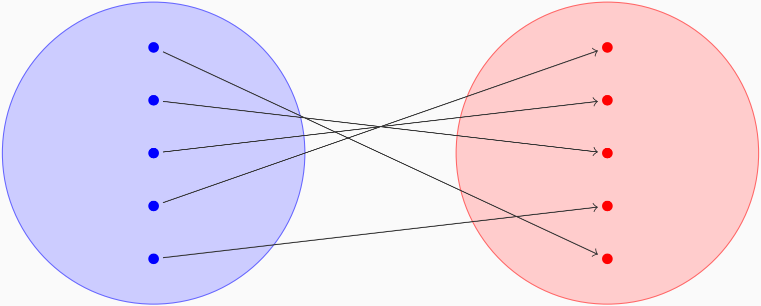
$$\forall x\ \exists^{=1} y.\ R(x, y)$$



What is the cardinality of R in all functions ??

Counting Quantifiers: Expressing Functionality

$$\forall x \; \exists^{=1} y. \; R(x, y)$$

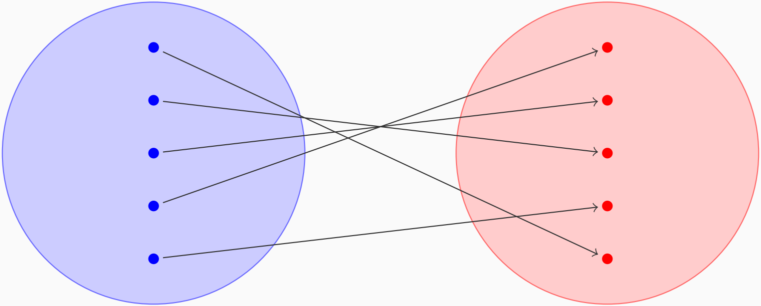


What is the cardinality of R in all functions ??

$$|\Delta|$$

Counting Quantifiers: Expressing Functionality

$$\forall x\ \exists^{=1} y.\ R(x, y)$$



What is the cardinality of R in all functions ??

$$|\Delta|$$

$$\forall x.\exists^{=1} y.R(x, y) \equiv \forall x.\exists y.R(x, y) \wedge (|R| = |\Delta|)$$

Counting Quantifiers[9]

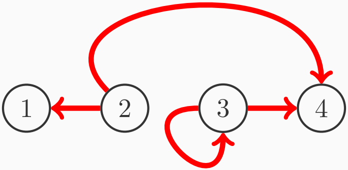
Let Φ be the following C^2 formula, then it can be written as:

$$\Phi_0 \wedge \bigwedge_{k=1}^{q} \forall x.(A_k(x) \leftrightarrow \exists^{=m_k} y.R_k(x, y))$$

where Φ_0 is a pure universal formula in FO^2.

[9] Kuzelka. JAIR 2021, Malhotra and Serafini. AAAI 2022 for this formulation

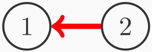
WFOMC beyond first-order logic [10]



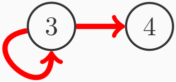
[10] Malhotra, Bizzaro and Serafini. 2024 (Under Review at Artificial Intelligence Journal)

Projections on sub-domain

Then, $\omega' = \omega \downarrow [2]$ and $\omega'' = \omega \downarrow [\bar{2}]$ are given respectively as

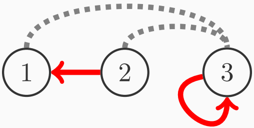


and



In how many ways can I merge them?

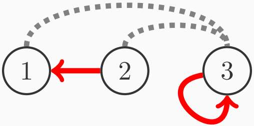
Then, $\omega' = \omega \downarrow [2]$ and $\omega'' = \omega \downarrow [\bar{2}]$ are given respectively as



What is the number of ways to merge these partial interpretations?

In how many ways can I merge them?

Then, $\omega' = \omega \downarrow [2]$ and $\omega'' = \omega \downarrow [\bar{2}]$ are given respectively as

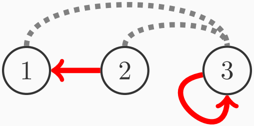


What is the number of ways to merge these partial interpretations?

$$2^4$$

In how many ways can I merge under constraints?

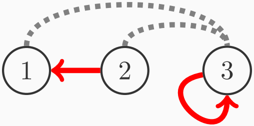
Then, $\omega' = \omega \downarrow [2]$ and $\omega'' = \omega \downarrow [\bar{2}]$ are given respectively as



What is the number of ways to merge these partial interpretations? Such that there are only directed edges from $[2]$ to $[\bar{2}]$?

In how many ways can I merge under constraints?

Then, $\omega' = \omega \downarrow [2]$ and $\omega'' = \omega \downarrow [\bar{2}]$ are given respectively as

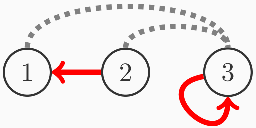


What is the number of ways to merge these partial interpretations?
Such that there are only directed edges from $[2]$ to $[\bar{2}]$?

$$2^2$$

In how many ways can I merge under constraints?

Then, $\omega' = \omega \downarrow [2]$ and $\omega'' = \omega \downarrow [\bar{2}]$ are given respectively as



What is the number of ways to merge these partial interpretations? Such that there are only directed edges from $[2]$ to $[\bar{2}]$?

$$2^2$$

This can be captured by restricting the allowed 2-types between Δ and Δ'

Counting by splitting

How can you compute WFOMC of all the models on $[m]$ and $[\bar{m}]$, such that:

C1: $\omega \downarrow [m] \models axiom'$

C2: $\omega \downarrow [\bar{m}] \models axiom''$

C3: $\omega \models \forall x \in [m]\ \forall y \in [\bar{m}].\Theta(x, y)$

C4: $\omega \models \forall xy.\Phi(x, y)$

Counting by splitting

How can you compute WFOMC of all the models on $[m]$ and $[\bar{m}]$, such that:

C1: $\omega \downarrow [m] \models axiom'$

C2: $\omega \downarrow [\bar{m}] \models axiom''$

C3: $\omega \models \forall x \in [m]\ \forall y \in [\bar{m}].\Theta(x, y)$

C4: $\omega \models \forall xy.\Phi(x, y)$

- $\mathsf{W}(\omega')\mathsf{W}(\omega'') \prod_{i,j \in [u]} r_{ij}^{k_i' k_j''}$, where r_{ij} captures the constrains across them.
- $$\mathsf{WFOMC}(\Psi_{[m]}, \boldsymbol{k}) = \sum_{\substack{\boldsymbol{k}' + \boldsymbol{k}'' = \boldsymbol{k} \\ |\boldsymbol{k}'| = m}} \mathsf{WFOMC}(\Psi', \boldsymbol{k}')\mathsf{WFOMC}(\Psi'', \boldsymbol{k}'') \prod_{i,j \in [u]} r_{ij}^{k_i' k_j''}$$

What can I do with this result?

	$axiom'$	$axiom''$	$\Theta(x, y)$
$DAG(R)$	$\forall xy.\neg R(x, y)$	$DAG(R)$	$\neg R(y, x)$
$Connected(R)$	$Connected(R)$	$\top$	$\neg R(x, y)$
$Forest(R)$	$Tree(R)$	$Forest(R)$	$\neg R(x, y)$

Table 1: A summary of results using counting by splitting

Open Problems

- Is there a some formal language that completely captures tractable WFOMC?

Open Problems

- Is there a some formal language that completely captures tractable WFOMC?
- Is there an FPRAS? (There is no PTAS (Jaeger 2014. TPLP))

Open Problems

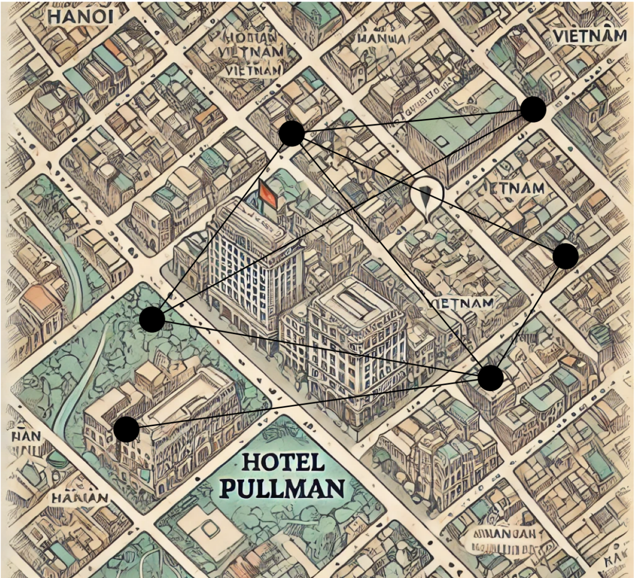
- Is there a some formal language that completely captures tractable WFOMC?
- Is there an FPRAS? (There is no PTAS (Jaeger 2014. TPLP))
- Evidence. Something is known about FPT (Broeck and Darwiche 2013. NeurIps). But not alot.

Open Problems

- Is there a some formal language that completely captures tractable WFOMC?
- Is there an FPRAS? (There is no PTAS (Jaeger 2014. TPLP))
- Evidence. Something is known about FPT (Broeck and Darwiche 2013. NeurIps). But not alot.
- Sampling – Amazing recent developments (Wang et al. LICS 2023)

Consistency

A Covid Outbreak in Pullman Hotel!



Back to my grim Example

A contact-tracing model:

$$w_1 : \texttt{Covid}(\texttt{x})$$
$$w_2 : \texttt{Covid}(\texttt{x}) \land \texttt{Contact}(\texttt{x}, \texttt{y}) \rightarrow \texttt{Covid}(\texttt{y})$$

$$\widehat{\mathbf{a}} = \underset{\mathbf{a}}{\operatorname{argmax}} \, P_{\Phi}^{(m)}(\omega) \tag{4}$$

A Covid Outbreak in Pullman Hotel!

What do we observe, and what do we learn?

How does one usually do parameter estimation?

$$\widehat{\mathbf{a}} = \operatorname*{argmax}_{\mathbf{a}} P_{\Phi}^{(m)}(\omega) \tag{5}$$

What do we observe, and what do we learn?

How does one usually do parameter estimation?

$$\widehat{\mathbf{a}} = \operatorname*{argmax}_{\mathbf{a}} P_{\Phi}^{(m)}(\omega) \tag{5}$$

What do we actually want to do?

$$\widehat{\mathbf{a}} = \operatorname*{argmax}_{\mathbf{a}} P_{\Phi}^{(n+m)} \downarrow [n](\omega)$$

where,

$$P^{(n)} \downarrow [m](\omega') = \sum_{\omega \in \Omega^{(n)} : \omega \downarrow [m] = \omega'} P^{(n)}(\omega)$$

Consistency of Learning and Inference

Consistency of Inference [Projectivity][11]: A model learnt over a domain of size n, basically says nothing quantitative about a domain of size $n+1$

$$P_{\boldsymbol{\theta}}^{(m)}(q) \neq P_{\boldsymbol{\theta}}^{(n)} \downarrow [m](q)$$

[11] Shalizi and Rinaldo. 2013. Jaeger and Shulte. StarAI Workshop 2018. Mittal, Bhardwaj, Gogate, Singla. AISTATS 2019.

Consistency of Learning and Inference

Consistency of Inference [Projectivity][11]: A model learnt over a domain of size n, basically says nothing quantitative about a domain of size $n+1$

$$P_{\boldsymbol{\theta}}^{(m)}(q) \neq P_{\boldsymbol{\theta}}^{(n)} \downarrow [m](q)$$

Consistency of Learning: Batching or stochastic gradient descent cannot work for MLNs ! as they donnot admit consistent parameter estimates

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, E_{\omega'}[\log P_{\boldsymbol{\theta}}^{(m)}(\omega')] \neq \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \log P_{\boldsymbol{\theta}}^{(n)}(\omega)$$

[11] Shalizi and Rinaldo. 2013. Jaeger and Shulte. StarAI Workshop 2018. Mittal, Bhardwaj, Gogate, Singla. AISTATS 2019.

Trying to get near Projectivity!

Can we improve the learning procedure to get near-projective models [12]

[12] Chen, Weitkamper, Malhotra. ECML 2024.
(For theory of RMP) Kuzelka, Wang, Davis and Schockaert. AAAI 2018.

Trying to get near Projectivity!

Can we improve the learning procedure to get near-projective models [12]

$$-\log P_{\Phi}^{(n+m)} \downarrow [n](\omega) \leq -\log P_{\Phi}^{(n)}(\omega) + \log \Delta \tag{6}$$

$$KL(P_{\Phi}^{(n+m)} \downarrow [n] || P_{\Phi}^{(n)}) \leq \log \Delta \tag{7}$$

[12] Chen, Weitkamper, Malhotra. ECML 2024.
(For theory of RMP) Kuzelka, Wang, Davis and Schockaert. AAAI 2018.

But what are Projective Models?

But what are Projective Models?

Erdős–Rényi random graph:

$$\Pr(G(n+m,p)=G)=p^{|E|}(1-p)^{\binom{n+m}{2}-|E|}$$

But what are Projective Models?

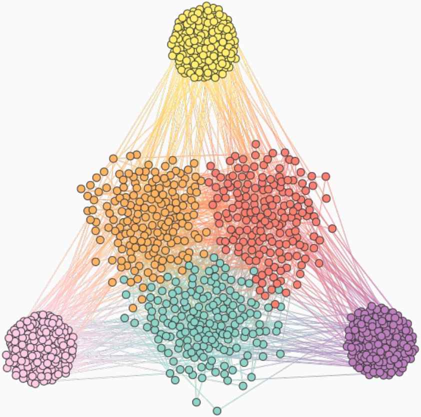
Erdős–Rényi random graph:

$$\Pr(G(n+m,p) = G) = p^{|E|}(1-p)^{\binom{n+m}{2}-|E|}$$

$$\begin{aligned} \Pr(G(n+m,p)\downarrow[n] = G') &= p^{|E'|}(1-p)^{\binom{n}{2}-|E'|} && (8)\\ &= \Pr(G(n,p) = G'). && (9)\end{aligned}$$

Can we express any projective models in MLNs?

"A Markov Logic Network in the Two Variable Fragment is projective if and only if it represents a stochastic block structure [13]"



[13] Malhotra and Serafini. ECML 2022

PROBABILITY PARAMETERS

- p_i is the probability of an arbitrary domain constant realising the ith 1-type

$$p_i := P\big(i(x)\big)$$

PROBABILITY PARAMETERS

- p_i is the probability of an arbitrary domain constant realising the ith 1-type

$$p_i := P\big(i(x)\big)$$

- w_{ijl} is the conditional probability of an arbitrary pair of domain constants to realise the lth 2-table, given they realise the i^{th} and the j^{th} 1-type.

$$w_{ijl} = P\big(l(x,y)|i(x) \land j(y)\big)$$

"Who Shaves Who in KR?"

Assuming **Shaves** to be irreflexive and symmetric, we have the following non-zero $\{p_i, w_{ijl}\}$

$$
\begin{aligned}
p_1 &= P(\mathsf{KR}(\mathsf{a})) \\
p_2 &= P(\neg\mathsf{KR}(\mathsf{a})) \\
w_{111} &= P(\mathsf{Shaves}(\mathsf{a},\mathsf{b})|\mathsf{KR}(\mathsf{a}),\mathsf{KR}(\mathsf{b})) \\
w_{112} &= P(\neg\mathsf{Shaves}(\mathsf{a},\mathsf{b})|\mathsf{KR}(\mathsf{a}),\mathsf{KR}(\mathsf{b})) \\
w_{121} &= P(\mathsf{Shaves}(\mathsf{a},\mathsf{b})|\mathsf{KR}(\mathsf{a}),\neg\mathsf{KR}(\mathsf{b})) \\
w_{122} &= P(\neg\mathsf{Shaves}(\mathsf{a},\mathsf{b})|\mathsf{KR}(\mathsf{a}),\neg\mathsf{KR}(\mathsf{b})) \\
w_{221} &= P(\mathsf{Shaves}(\mathsf{a},\mathsf{b})|\neg\mathsf{KR}(\mathsf{a}),\neg\mathsf{KR}(\mathsf{b})) \\
w_{222} &= P(\neg\mathsf{Shaves}(\mathsf{a},\mathsf{b})|\neg\mathsf{KR}(\mathsf{a}),\neg\mathsf{KR}(\mathsf{b}))
\end{aligned}
$$

The Relational Block Model

Relational Block Model :

$$P(\boldsymbol{X} = \boldsymbol{x}) := \prod_{q=1}^{n} p_{x_q} = \prod_{i=1}^{u} p_{x_i}^{k_i}$$

$$P(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{X} = \boldsymbol{x}) := \prod_{1 \leq q < r \leq n} w_{x_q x_r y_{qr}}$$

$$= \prod_{1 \leq i \leq j \leq u} \prod_{1 \leq l \leq b} (w_{ijl})^{h_l^{ij}}$$

Any projective MLN in the two variable fragment reduces to an RBM

Malhotra and Serafini. ECML PKDD 2022

What is the most general class of projective models on graphs?

[14] László Lovász and Aldous-Hoover-Kallenberg

What is the most general class of projective models on graphs?

A **graphon**[14] is a symmetric, measurable function $W : [0,1] \times [0,1] \to [0,1]$ such that:

- $W(x,y) = W(y,x)$ for all $x, y \in [0,1]$ (symmetry),
- $W(x,y)$ gives the probability of an edge between points x and y.

[14] László Lovász and Aldous-Hoover-Kallenberg

What is the most general class of projective models on graphs?

A **graphon**[14] is a symmetric, measurable function $W : [0,1] \times [0,1] \to [0,1]$ such that:

- $W(x,y) = W(y,x)$ for all $x, y \in [0,1]$ (symmetry),
- $W(x,y)$ gives the probability of an edge between points x and y.

Sampling a graph from a graphon:

1. Sample n points $u_1, u_2, \ldots, u_n$ uniformly from $[0,1]$.
2. For each pair (i,j) with $i \neq j$, place an edge with probability $W(u_i, u_j)$.

[14] László Lovász and Aldous-Hoover-Kallenberg

What do all these models have in common?

What do all these models have in common?

- Exchangeable Distributions:

$$P((X_{ij})) = P((X_{\sigma(i)\sigma(j)}))$$

What do all these models have in common?

- Exchangeable Distributions:

$$P((X_{ij})) = P((X_{\sigma(i)\sigma(j)}))$$

- X_{ij} are edges for a ER graph or a graphon
- X_{ij} are 2-types for FO^2
- Can be generalized to exchangeable random arrays like X_{ijl}

Aldous-Hoover-Kallenberg

Theorem 1 (Aldous-Hoover representation of jointly exchangeable matrices (Aldous, 1981; Hoover, 1979)). A random 2-array $(X_{ij})_{i,j \in \mathbb{N}}$ is jointly exchangeable if there exists a function $f : [0,1] \times [0,1]^2 \times [0,1] \to E$ such that

$$(X_{ij}) = (f(U, U_i, U_j, U_{ij})),$$

where $(U_i)_{i \in \mathbb{N}}$ and $(U_{ij})_{i,j>i \in \mathbb{N}}$ with $U_{ij} = U_{ji}$ are a sequence and matrix, respectively, of i.i.d. Uniform$[0,1]$ random variables.

Jaeger and Schulte. IJCAI 2020. For extension to Relational data.

Open Problems

- How to represent and learn expressive AHK models?

Open Problems

- How to represent and learn expressive AHK models?
- How to encode expert Knowledge into AHK models?

Thank You!