UNIVERSITY OF TRENTO - Italy
Department of Information Engineering and Computer Science

Weighted Model Counting in the Two Variable Fragments

AAAI 2022 @ FBK

Sagar Malhotra[1,2] and Luciano Serafini [1]

[1]Fondazione Bruno Kessler
[2]University of Trento

Model Counting

Given a logical formula Φ. e.g.

$$\text{Trentino} \rightarrow \text{Climber}$$

MC ?

Model Counting

Given a logical formula Φ. e.g.

$$\text{Trentino} \to \text{Climber}$$

MC ?

Trentino	Climber	Trentino $\to$ Climber
1	1	1
1	0	0
0	1	1
0	0	1

$$\text{MC} = 3$$

Weighted Model Counting

Given a logical formula Φ. e.g.

$$\text{Trentino} \rightarrow \text{Climber}$$

And a weight function e.g.

$$w(\text{Trentino}) = 2.5 \qquad w(\neg\text{Trentino}) = 5$$
$$w(\text{Climber}) = 0.5 \qquad w(\neg\text{Climber}) = 1.5$$

WMC ?

Weighted Model Counting

Given a logical formula Φ. e.g.

$$\text{Trentino} \rightarrow \text{Climber}$$

And a weight function e.g.

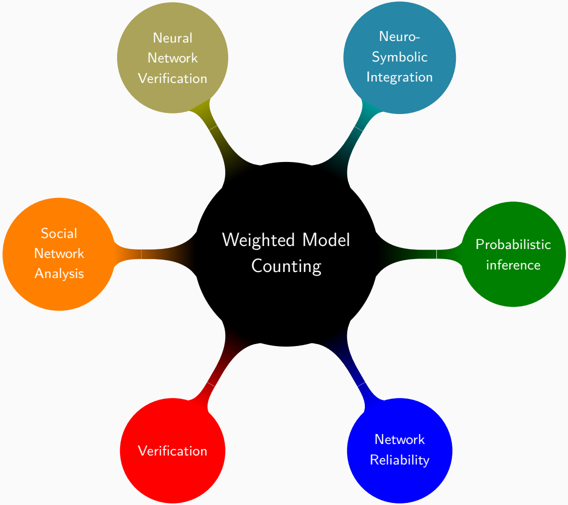
$$w(\text{Trentino}) = 2.5 \qquad w(\neg\text{Trentino}) = 5$$
$$w(\text{Climber}) = 0.5 \qquad w(\neg\text{Climber}) = 1.5$$

WMC ?

Trentino	Climber	Trentino $\rightarrow$ Climber
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	$5 \times 1.5 = 7.5$

WMC = 11.25

WMC Applications



Weighted Model Counting: Approaches

- Devising heuristic/approximation algorithms
  WMC is intractable !

Weighted Model Counting: Approaches

- Devising heuristic/approximation algorithms
  WMC is intractable !
- Identifying logical languages that admit efficient WMC
  Today's presentation

Weighted First Order Model Counting

$$\forall x.\mathsf{Trentino}(x) \rightarrow \mathsf{Climber}(x)$$

A weight function associated to predicates:

$$w(\mathsf{Trentino}) = 2.5 \qquad w(\neg\mathsf{Trentino}) = 5$$
$$w(\mathsf{Climber}) = 0.5 \qquad w(\neg\mathsf{Climber}) = 1.5$$

What is the WFOMC for n people ?

Weighted First Order Model Counting

$$\forall x.\text{Trentino}(x) \rightarrow \text{Climber}(x)$$

A weight function associated to predicates:

$$w(\text{Trentino}) = 2.5 \qquad w(\neg\text{Trentino}) = 5$$
$$w(\text{Climber}) = 0.5 \qquad w(\neg\text{Climber}) = 1.5$$

What is the WFOMC for n people ?

Can we compute it in PTIME?

Weighted First Order Model Counting

$$\forall x. \text{Trentino}(x) \rightarrow \text{Climber}(x)$$

A weight function associated to predicates:

$$w(\text{Trentino}) = 2.5 \qquad w(\neg\text{Trentino}) = 5$$

$$w(\text{Climber}) = 0.5 \qquad w(\neg\text{Climber}) = 1.5$$

What is the WFOMC for n people ?

Can we compute it in PTIME?

Can we get a closed form formula ?

FOMC in the two variable fragment

FO^2 Language: 1-Types

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

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

FO^2 Language: 1-Types

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

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

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

$$\neg\mathtt{FBK}(\mathtt{c}) \land \neg\mathtt{Shaves}(\mathtt{c}, \mathtt{c})$$

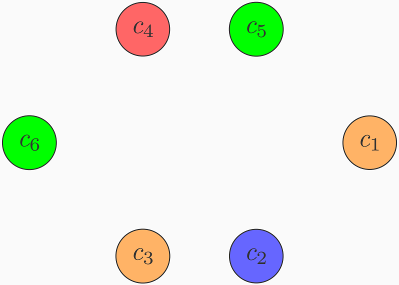
$$\neg\mathtt{FBK}(\mathtt{c}) \land \mathtt{Shaves}(\mathtt{c}, \mathtt{c})$$

$$\mathtt{FBK}(\mathtt{c}) \land \neg\mathtt{Shaves}(\mathtt{c}, \mathtt{c})$$

$$\mathtt{FBK}(\mathtt{c}) \land \mathtt{Shaves}(\mathtt{c}, \mathtt{c})$$

1- Type Enumeration

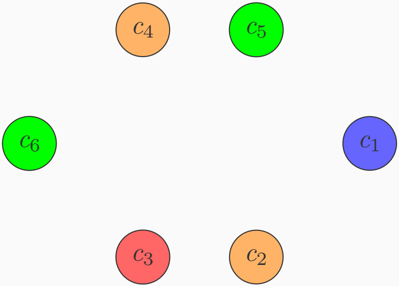
An arrangement of 1-Types



$k_1 = 1 \quad k_2 = 2 \quad k_3 = 1 \quad k_4 = 2$

2- Table Enumeration

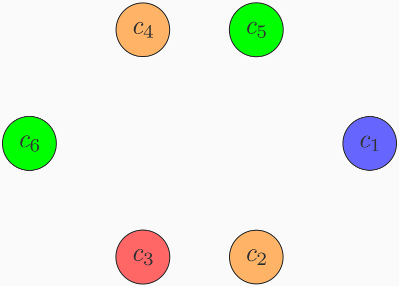
Another arrangement with the same 1-Type cardinalities



$k_1 = 1 \quad k_2 = 2 \quad k_3 = 1 \quad k_4 = 2$

2- Table Enumeration

Another arrangement with the same 1-Type cardinalities



$k_1 = 1 \quad k_2 = 2 \quad k_3 = 1 \quad k_4 = 2$

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

FO^2 Language: 2-tables

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

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

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

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

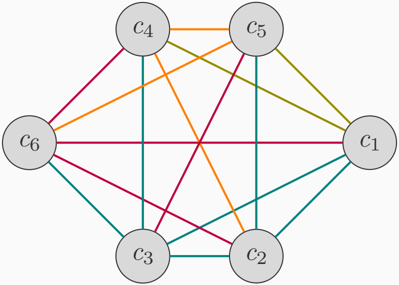
$$\neg\mathtt{Shaves}(\mathtt{c}, \mathtt{d}) \land \mathtt{Shaves}(\mathtt{d}, \mathtt{c})$$

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

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

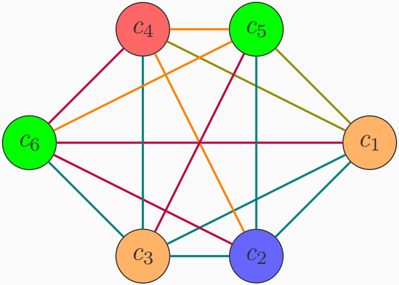
2- Table Enumeration

An arrangement for 2-tables



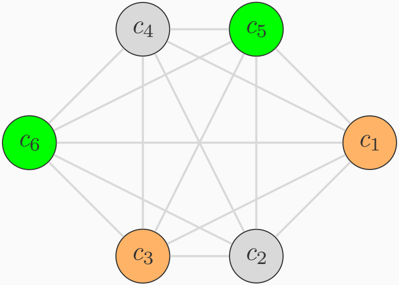
2- Table Enumeration given 1-Types

An arrangement for 2-tables given 1-types



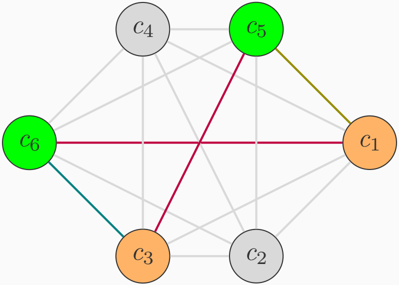
2- Table Enumeration given 1-Types

Picking a sub graph: Pick a pair of 1-Types



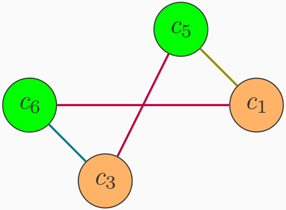
2- Table Enumeration given 1-Types

Picking a sub graph: Pick 2-Tables between them



2- Table Enumeration

Enumerating 2-tables given 1-types

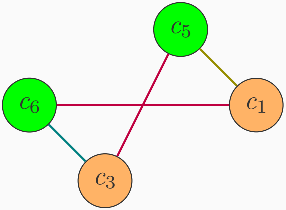


$$k_2 = 2 \quad k_4 = 2$$

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

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

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

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

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:

$$\neg\mathtt{Shaves}(\mathtt{x}, \mathtt{x})$$
$$\mathtt{Shaves}(\mathtt{x}, \mathtt{y}) \rightarrow \mathtt{Shaves}(\mathtt{y}, \mathtt{x})$$
$$\mathtt{FBK}(\mathtt{x}) \wedge \mathtt{Shaves}(\mathtt{x}, \mathtt{y}) \rightarrow \mathtt{FBK}(\mathtt{y})$$

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

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

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

$$\mathtt{FBK}(\mathtt{x}) \land \mathtt{Shaves}(\mathtt{x},\mathtt{y}) \rightarrow \mathtt{FBK}(\mathtt{y})$$

Not allowed:

$$\neg\mathtt{FBK}(\mathtt{c}) \land \neg\mathtt{Sh}(\mathtt{c},\mathtt{c}) \land \mathtt{FBK}(\mathtt{d}) \land \neg\mathtt{Sh}(\mathtt{d},\mathtt{d}) \land \mathtt{Sh}(\mathtt{c},\mathtt{d}) \land \mathtt{Sh}(\mathtt{d},\mathtt{c})$$

Allowed:

$$\mathtt{FBK}(\mathtt{c}) \land \neg\mathtt{Sh}(\mathtt{c},\mathtt{c}) \land \mathtt{FBK}(\mathtt{d}) \land \neg\mathtt{Sh}(\mathtt{d},\mathtt{d}) \land \mathtt{Sh}(\mathtt{c},\mathtt{d}) \land \mathtt{Sh}(\mathtt{d},\mathtt{c})$$

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

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

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

Unary Properties

Constraints: Φ

Binary Properties

Generalisations

1. **Existential Quantifiers**: $\forall \mathtt{x} \exists \mathtt{y}.\mathtt{Shaves}(\mathtt{x},\mathtt{y})$
    - Key Idea: Principle of Inclusion Exclusion

Generalisations

1. **Existential Quantifiers**: $\forall \mathtt{x} \exists \mathtt{y}.\mathtt{Shaves}(\mathtt{x}, \mathtt{y})$
   - Key Idea: Principle of Inclusion Exclusion
2. **Cardinality Constraints**: $|\mathtt{Shaves}| \geq 10$
   - Key Idea: Count only the $\boldsymbol{k}$ and $\boldsymbol{h}$ such that the cardinality is restricted

Generalisations

1. **Existential Quantifiers**: $\forall x \exists y.\mathtt{Shaves}(x, y)$
   - Key Idea: Principle of Inclusion Exclusion
2. **Cardinality Constraints**: $|\mathtt{Shaves}| \geq 10$
   - Key Idea: Count only the $\boldsymbol{k}$ and $\boldsymbol{h}$ such that the cardinality is restricted
3. **Counting Quantifiers**: $\forall x \exists^{\geq 5} y.\mathtt{Shaves}(x, y)$
   - Key Idea: Any such formula is reducible to a formula with cardinality constraints and existential quantifiers

Applications

- Learning and inference in relational models: Exponential Random Graphs, Probabilistic Logic, Probabilistic Databases etc.

Applications

- Learning and inference in relational models: Exponential Random Graphs, Probabilistic Logic, Probabilistic Databases etc.
- Network Reliability Testing: First Order Counters can serve as simulation set up for testing propositional model counters

Applications

- Learning and inference in relational models: Exponential Random Graphs, Probabilistic Logic, Probabilistic Databases etc.
- Network Reliability Testing: First Order Counters can serve as simulation set up for testing propositional model counters
- Software Verification: AlloyMC, already a higher order language.

Conclusion

Weighted Model Counting is an assembly language to a vast array of problems.

Conclusion

Weighted Model Counting is an assembly language to a vast array of problems.

Only first-order logic fragments have ever been shown to admit exact tractable counting.

Conclusion

Weighted Model Counting is an assembly language to a vast array of problems.

Only first-order logic fragments have ever been shown to admit exact tractable counting.

In this work, we expand the fragments admitting tractable counting and provide a combinatorial framework that admits closed form formulae.

Future Works

- Consistency of probabilistic inference [Preprint Online]
- Approximate counting with guarantees

Thank You !