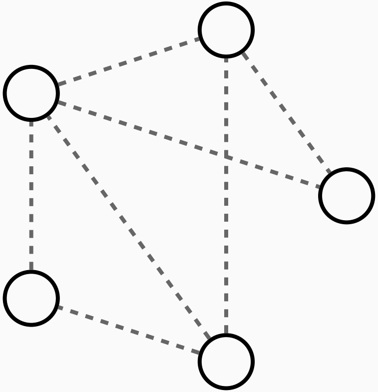
Learning Concepts Described by Weight Aggregation Logic

Steffen van Bergerem, Nicole Schweikardt

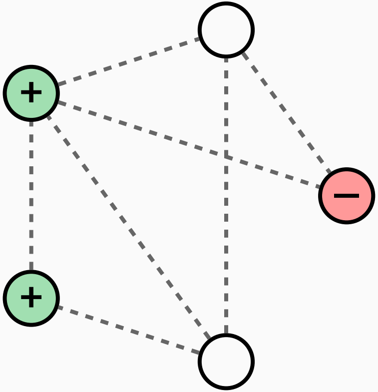
CSL 2021

Learning from Examples

Learning from Examples on Relational Structures

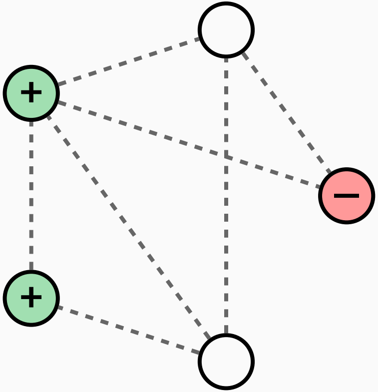


Learning from Examples on Relational Structures



Input: Labelled Examples
labelled tuples $(v_1, \ldots, v_k)$

Learning from Examples on Relational Structures



Input: Labelled Examples

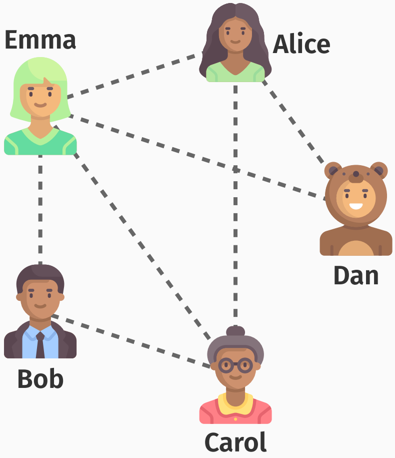
labelled tuples $(v_1, \ldots, v_k)$

Output: Concept

formula $\varphi(x_1, \ldots, x_k; y_1, \ldots, y_\ell)$,

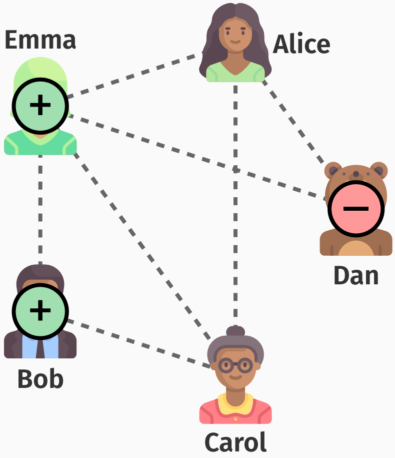
parameters $w_1, \ldots, w_\ell$

Learning from Examples on Relational Structures



Icons made by Freepik from Flaticon

Learning from Examples on Relational Structures



1

Positive Examples

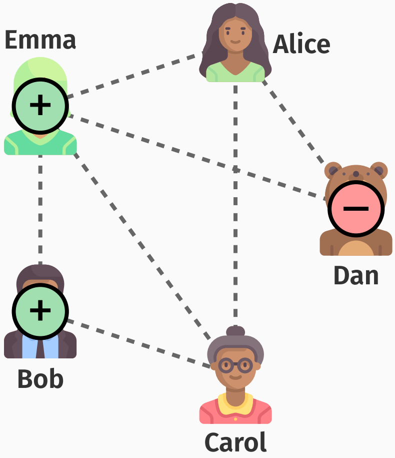
Bob

Emma

Negative Examples

Dan

Learning from Examples on Relational Structures



1

Positive Examples

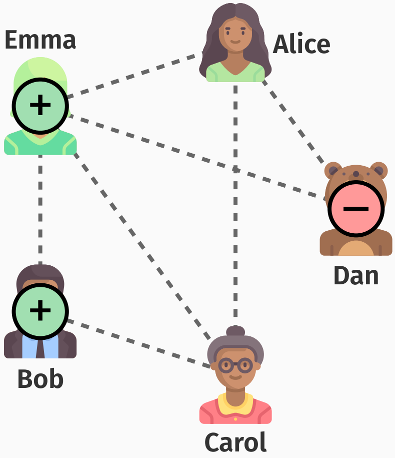
Bob

Emma

Negative Examples

Dan

Possible Concept
Carol’s friends



1

Positive Examples

Bob

Emma

Negative Examples

Dan

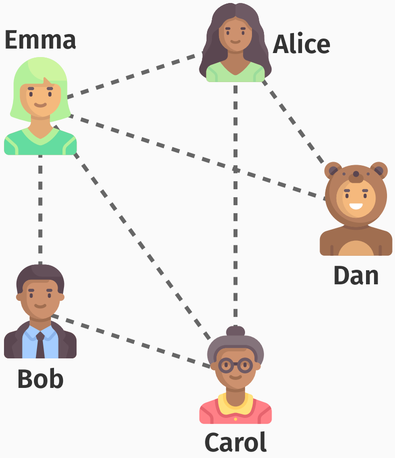
Possible Concept

Carol's friends

$$\varphi(x; y) = F(x, y)$$

parameter: Carol

Learning from Examples on Relational Structures

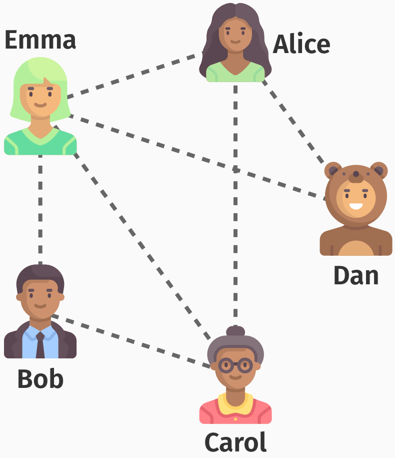


2

Positive Examples

(Alice, Bob)
(Dan, Emma)
(Carol, Dan)

Learning from Examples on Relational Structures



2

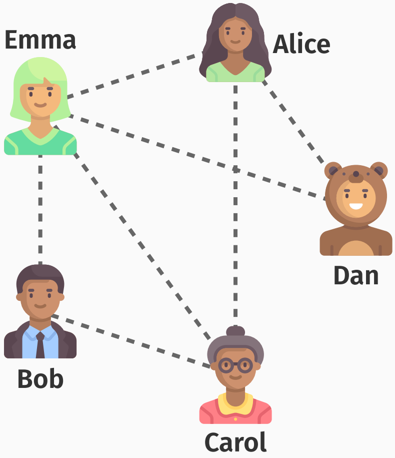
Positive Examples

(Alice, Bob)
(Dan, Emma)
(Carol, Dan)

Negative Examples

(Bob, Dan)
(Alice, Carol)

Learning from Examples on Relational Structures



2

Positive Examples

(Alice, Bob)
(Dan, Emma)
(Carol, Dan)

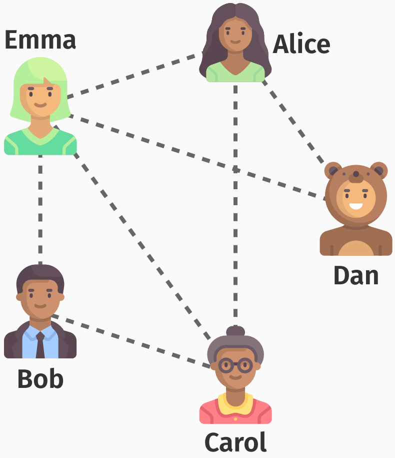
Negative Examples

(Bob, Dan)
(Alice, Carol)

Possible Concept
having a common friend who is not Emma

Icons made by Freepik from Flaticon

Learning from Examples on Relational Structures



2

Positive Examples

(Alice, Bob)
(Dan, Emma)
(Carol, Dan)

Negative Examples

(Bob, Dan)
(Alice, Carol)

Possible Concept

having a common friend who is not Emma

$$\varphi(x_1, x_2; y) = \exists z \left(F(x_1, z) \wedge F(x_2, z) \wedge z \neq y\right)$$

parameter: Emma

Icons made by Freepik from Flaticon

Learning Concepts Definable in Logics

Theorem (Grohe and Ritzert, 2017)

Concepts definable in FO on structures of small degree can be learned in sublinear time (in the size of the structure).

Learning Concepts Definable in Logics

Theorem (Grohe and Ritzert, 2017)

Concepts definable in FO on structures of small degree can be learned in sublinear time (in the size of the structure).

Theorem

Concepts definable in the weight aggregation logic FOWA_1 on weighted structures of small degree can be learned in sublinear time with quasilinear precomputation.

Learning Concepts in Sublinear Time

We only consider formulas of limited complexity
(limited nesting depth)

```
for all formulas do
    for all parameters do
        if concept is consistent then
            return concept
reject
```

Learning Concepts in Sublinear Time

We only consider formulas of limited complexity (limited nesting depth)

```
for all normal forms of formulas do
    for all parameters do
        if concept is consistent then
            return concept
reject
```

We only consider formulas of limited complexity (limited nesting depth)

constantly many

```
1: for all normal forms of formulas do
2:     for all parameters do
3:         if concept is consistent then
4:             return concept
5: reject
```

1) **Gaifman normal form**, uses enriched structure with quasilinear-time precomputation

Learning Concepts in Sublinear Time

We only consider formulas of limited complexity (limited nesting depth)

constantly many (→ normal forms)

1: **for all** normal forms of formulas **do**
2: **for all** parameters **in a certain neighbourhood** **do**
3: **if** concept is consistent **then**
4: **return** concept
5: **reject**

1) Gaifman normal form

Learning Concepts in Sublinear Time

We only consider formulas of limited complexity (limited nesting depth)

constantly many

```
for all normal forms of formulas do
    for all parameters in a certain neighbourhood do
        if concept is consistent then
            return concept
reject
```

1) Gaifman normal form
2) **local formulas and Feferman-Vaught decompositions**

Learning Concepts in Sublinear Time

We only consider formulas of limited complexity (limited nesting depth)

constantly many

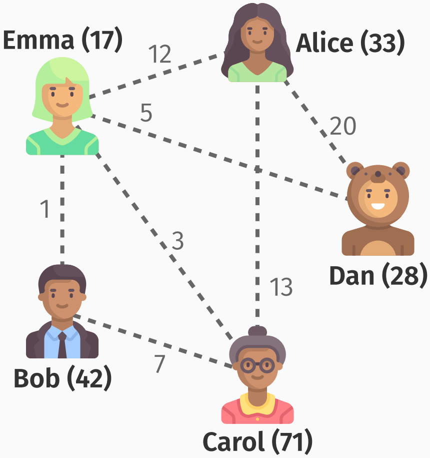
for all normal forms of formulas **do**
    for all parameters **in a certain neighbourhood do**
        if concept is consistent **then**
            return concept
reject

small degree, small neighbourhood (polylogarithmic)

1) Gaifman normal form
2) **local formulas and Feferman-Vaught decompositions**

Weight Aggregation Logic

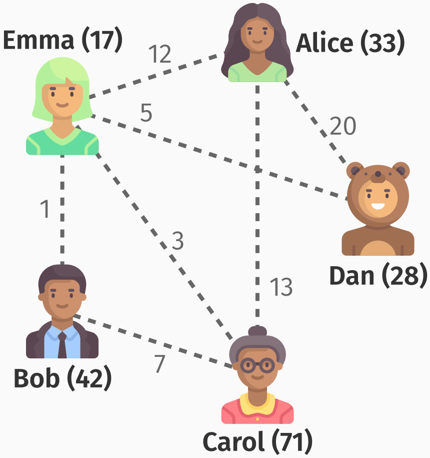
Weighted Structures



- relational structure

+ **weights** for weight symbols in $\mathbf{W}$

Weighted Structures

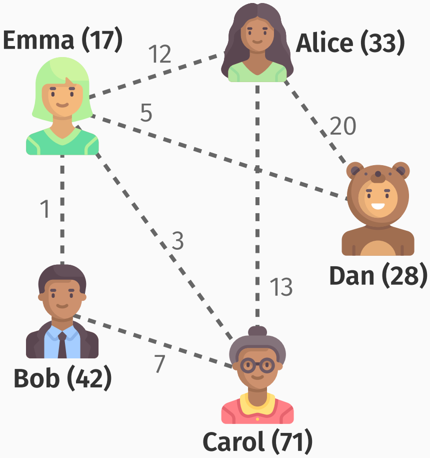


- relational structure
+ **weights** for weight symbols in $\mathbf{W}$

Example

- $\mathbf{W} = \{a, \ell\}$
- age $a(v) \in \mathbb{Z}$
- length of friendship $\ell(v, w) \in \mathbb{Z}$

First-Order Logic with Weight Aggregation (FOWA)

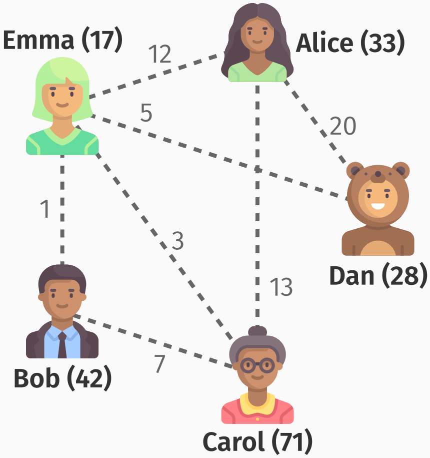


Terms

- $t_1(x, y) = 5 \cdot a(x) - a(y) \cdot \ell(x, y)$

Icons made by Freepik from Flaticon

First-Order Logic with Weight Aggregation (FOWA)

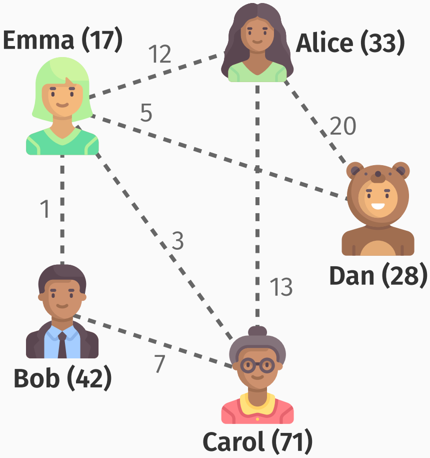


Terms

- $t_1(x, y) = 5 \cdot a(x) - a(y) \cdot \ell(x, y)$
- $t_2(x) = \sum_y a(y)\ell(x, y).F(x, y)$

Icons made by Freepik from Flaticon

First-Order Logic with Weight Aggregation (FOWA)

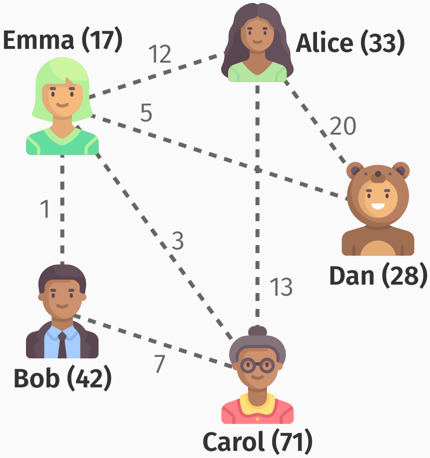


Terms

- $t_1(x, y) = 5 \cdot a(x) - a(y) \cdot \ell(x, y)$
- $t_2(x) = \sum_y a(y)\ell(x, y).F(x, y)$

Formulas

First-Order Logic with Weight Aggregation (FOWA)



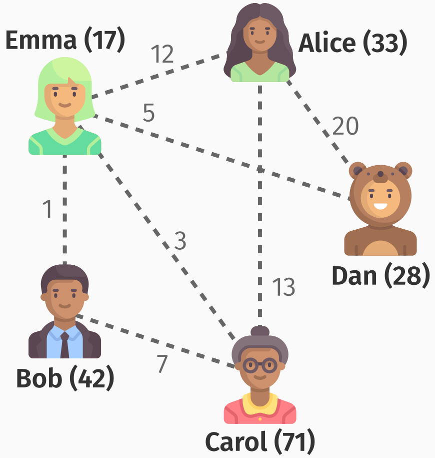
Terms

- $t_1(x, y) = 5 \cdot a(x) - a(y) \cdot \ell(x, y)$
- $t_2(x) = \sum_y a(y)\ell(x, y).F(x, y)$

Formulas

- $\varphi_1(x) = \big(8 = \sum_y \ell(x, y).F(x, y)\big)$

First-Order Logic with Weight Aggregation (FOWA)



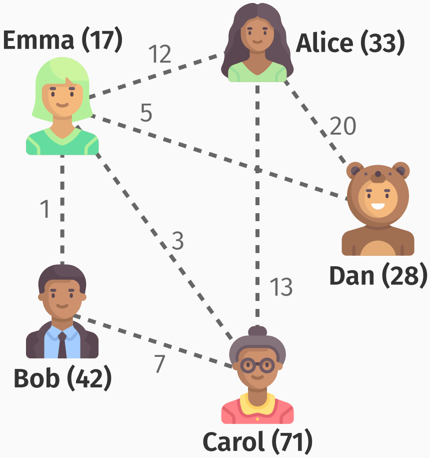
Terms

- $t_1(x, y) = 5 \cdot a(x) - a(y) \cdot \ell(x, y)$
- $t_2(x) = \sum_y a(y)\ell(x, y).F(x, y)$

Formulas

- $\varphi_1(x) = \big(8 = \sum_y \ell(x, y).F(x, y)\big)$
- $\varphi_2(x) = \big(\sum_y a(y).F(x, y) \leqslant \sum_y a(y).\neg F(x, y)\big)$

First-Order Logic with Weight Aggregation (FOWA)



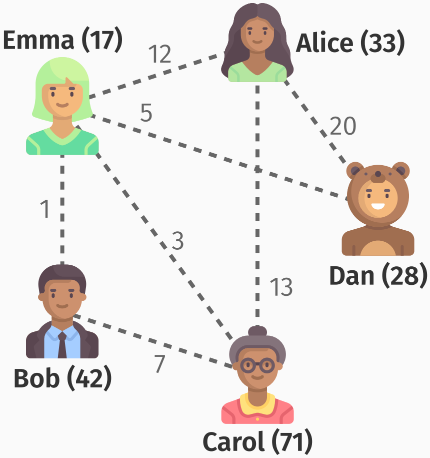
Terms

- $t_1(x, y) = 5 \cdot a(x) - a(y) \cdot \ell(x, y)$
- $t_2(x) = \sum_y a(y)\ell(x, y).F(x, y)$

Formulas

- $\varphi_1(x) = \left(8 = \sum_y \ell(x, y).F(x, y)\right)$
- $\varphi_2(x) = \left(\sum_y a(y).F(x, y) \leqslant \sum_y a(y).\neg F(x, y)\right)$
- $\varphi_3(x, y) = \left(\#(z).(F(x, z) \wedge F(y, z)) \geqslant 2 \right)$

First-Order Logic with Weight Aggregation (FOWA)



Terms

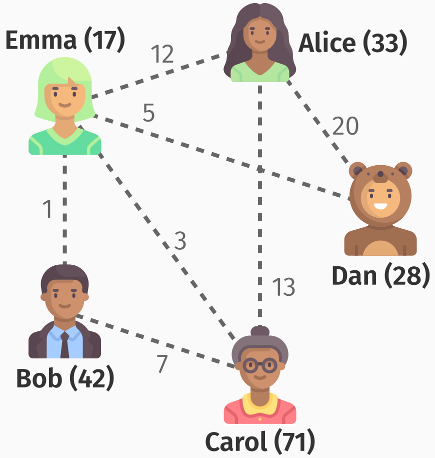
- $t_1(x, y) = 5 \cdot a(x) - a(y) \cdot \ell(x, y)$
- $t_2(x) = \sum_y a(y)\ell(x, y).F(x, y)$

Formulas

- $\varphi_1(x) = \left(8 = \sum_y \ell(x, y).F(x, y)\right)$
- $\varphi_2(x) = \left(\sum_y a(y).F(x, y) \leqslant \sum_y a(y).\neg F(x, y)\right)$
- $\varphi_3(x, y) = \left(\#(z).(F(x, z) \wedge F(y, z)) \geqslant 2 \right)$

FOWA$_1$

First-Order Logic with Weight Aggregation (FOWA)



Terms

- $t_1(x, y) = 5 \cdot a(x) - a(y) \cdot \ell(x, y)$
- $t_2(x) = \sum_y a(y)\ell(x, y).F(x, y)$

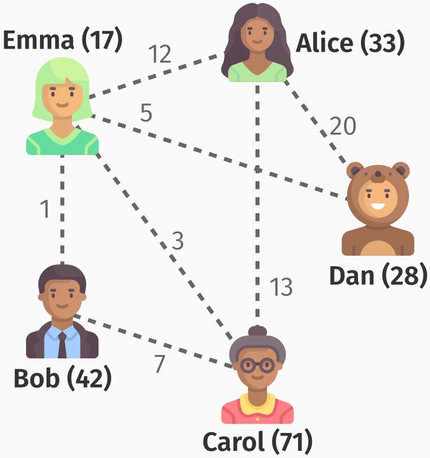
Formulas

- $\varphi_1(x) = (8 = \sum_y \ell(x, y).F(x, y))$
- $\varphi_2(x) = \left(\sum_y a(y).F(x, y) \leqslant \sum_y a(y).\neg F(x, y)\right)$
- $\varphi_3(x, y) = \left(\#(z).(F(x, z) \wedge F(y, z)) \geqslant 2 \right)$

FOWA_1

- only finite ring in $s = \sum w(y_1, \ldots, y_k).\varphi$

First-Order Logic with Weight Aggregation (FOWA)



Terms

- $t_1(x, y) = 5 \cdot a(x) - a(y) \cdot \ell(x, y)$
- $t_2(x) = \sum_y a(y)\ell(x, y).F(x, y)$

Formulas

- $\varphi_1(x) = \big(8 = \sum_y \ell(x, y).F(x, y)\big)$
- $\varphi_2(x) = \big(\sum_y a(y).F(x, y) \leqslant \sum_y a(y).\neg F(x, y)\big)$
- $\varphi_3(x, y) = \Big(\ \#(z).(F(x, z) \land F(y, z)) \ \geqslant \ 2\ \Big)$

FOWA$_1$

- only finite ring in $s = \sum w(y_1, \ldots, y_k).\varphi$
- only one free variable in $P(t_1, \ldots, t_m)$

Learning Concepts Definable in Weight Aggregation Logic

Theorem

Concepts definable in the weight aggregation logic FOWA_1 *on weighted structures of small degree can be learned in sublinear time with quasilinear precomputation.*

Learning Concepts Definable in Weight Aggregation Logic

Theorem

Concepts definable in the weight aggregation logic $FOWA_1$ *on weighted structures of small degree can be learned in sublinear time with quasilinear precomputation.*

1. **Localisation Theorem for** $\mathbf{FOWA_1}$
   Use local formula from FOW_1 on enriched structure instead.

Learning Concepts Definable in Weight Aggregation Logic

Theorem

Concepts definable in the weight aggregation logic FOWA_1 *on weighted structures of small degree can be learned in sublinear time with quasilinear precomputation.*

1. **Localisation Theorem for FOWA_1**
   Use local formula from FOW_1 on enriched structure instead.
2. **Gaifman normal form for FOW_1**
3. **Feferman-Vaught decompositions for FOW_1**

Learning Concepts Definable in Weight Aggregation Logic

Theorem

Concepts definable in the weight aggregation logic FOWA_1 on weighted structures of small degree can be learned in sublinear time with quasilinear precomputation.

+ PAC learning

1. **Localisation Theorem for FOWA_1**
   Use local formula from FOW_1 on enriched structure instead.
2. **Gaifman normal form for FOW_1**
3. **Feferman-Vaught decompositions for FOW_1**

Learning Concepts Definable in Weight Aggregation Logic

Theorem

Concepts definable in the weight aggregation logic FOWA_1 on weighted structures of small degree can be learned in sublinear time with quasilinear precomputation.

+ PAC learning

1. **Localisation Theorem for FOWA_1**
   Use local formula from FOW_1 on enriched structure instead.
2. **Gaifman normal form for FOW_1**
3. **Feferman-Vaught decompositions for FOW_1**

Future work

allow more free variables in $P(t_1, \dots, t_m)$ in a guarded setting