

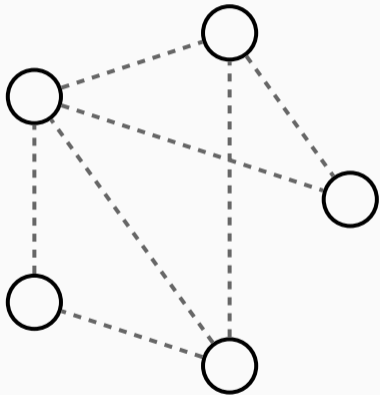
Learning Concepts Described by Weight Aggregation Logic

Steffen van Bergerem, Nicole Schweikardt

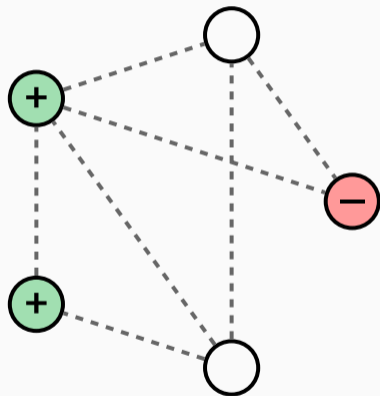
CSL 2021

Learning from Examples

Learning from Examples on Relational Structures



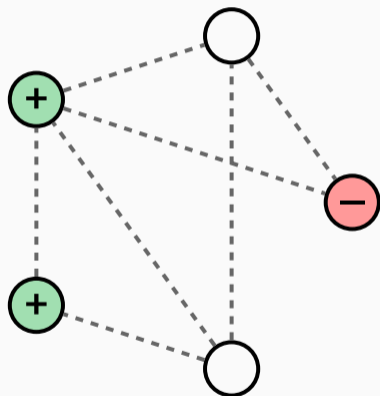
Learning from Examples on Relational Structures



Input: Labelled Examples

labelled tuples (v_1, \dots, v_k)

Learning from Examples on Relational Structures



Input: Labelled Examples

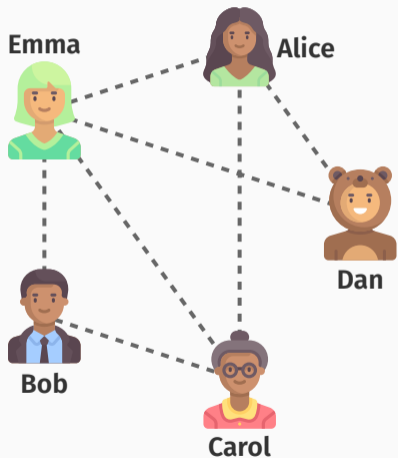
labelled tuples (v_1, \dots, v_k)

Output: Concept

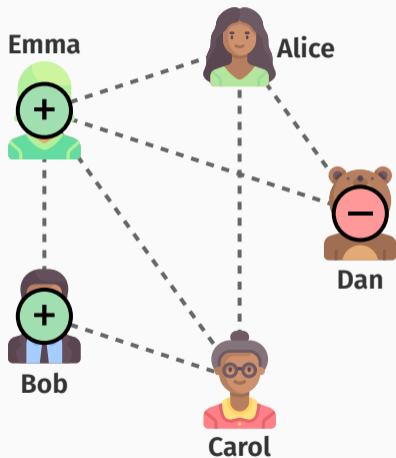
formula $\varphi(x_1, \dots, x_k; y_1, \dots, y_\ell)$,

parameters w_1, \dots, w_ℓ

Learning from Examples on Relational Structures



Learning from Examples on Relational Structures



1

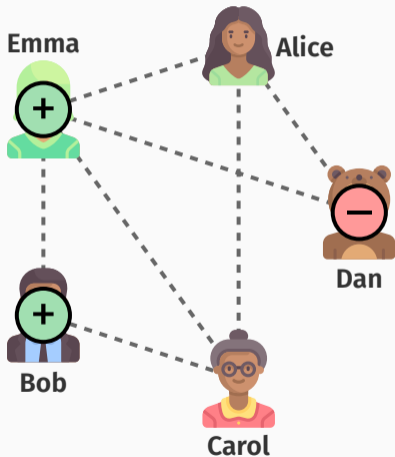
Positive Examples

Bob
Emma

Negative Examples

Dan

Learning from Examples on Relational Structures



1

Positive Examples

Bob
Emma

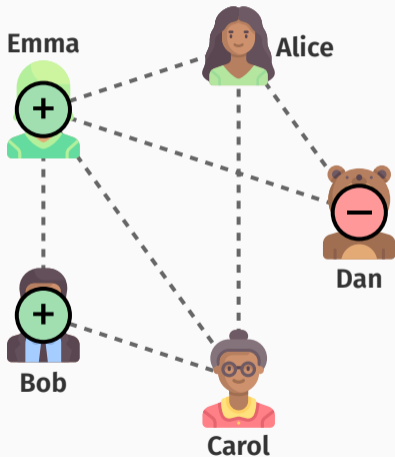
Negative Examples

Dan

Possible Concept

Carol's friends

Learning from Examples on Relational Structures



1

Positive Examples

Bob
Emma

Negative Examples

Dan

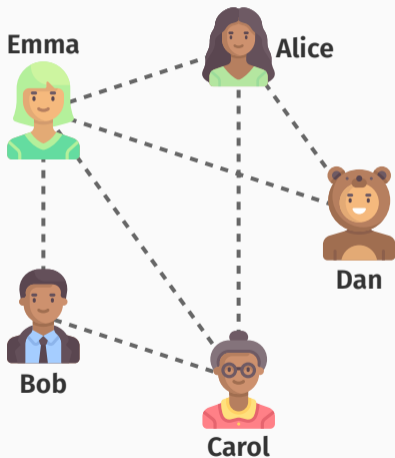
Possible Concept

Carol's friends

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

parameter: Carol

Learning from Examples on Relational Structures

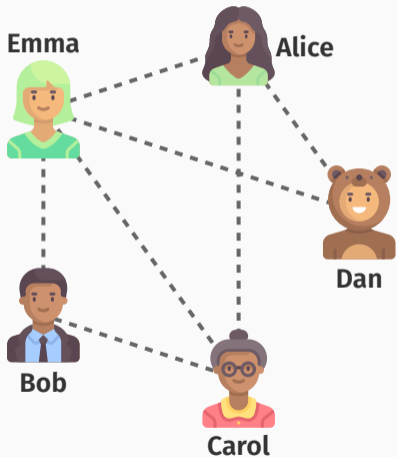


2

Positive Examples

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

Learning from Examples on Relational Structures



2

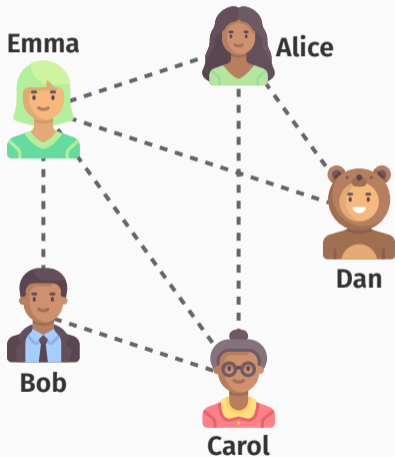
Positive Examples

(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)

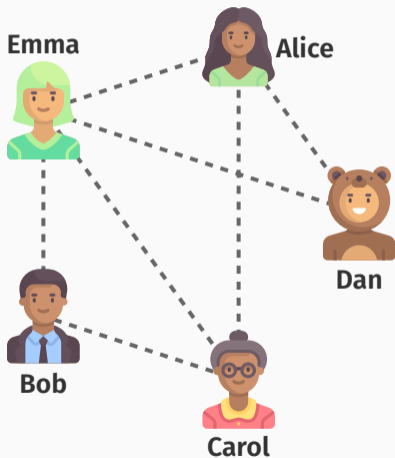
Negative Examples

(Bob, Dan)
(Alice, Carol)

Possible Concept

having a common friend who is not Emma

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

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

parameter: Emma

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 (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)

- 1: **for all** formulas **do**
- 2: **for all** parameters **do**
- 3: **if** concept is consistent **then**
- 4: **return** concept
- 5: **reject**

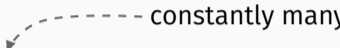
Learning Concepts in Sublinear Time

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

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

Learning Concepts in Sublinear Time

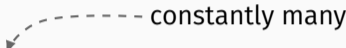
We only consider formulas of limited complexity
(limited nesting depth)

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

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

Learning Concepts in Sublinear Time

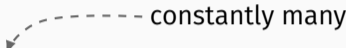
We only consider formulas of limited complexity
(limited nesting depth)

- 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)

- 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
- 2) **local formulas and Feferman-Vaught decompositions**

Learning Concepts in Sublinear Time

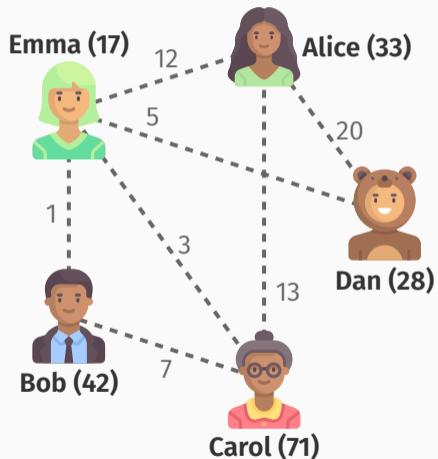
We only consider formulas of limited complexity
(limited nesting depth)

- 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**
- constantly many
- small degree, small neighbourhood
(polylogarithmic)

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

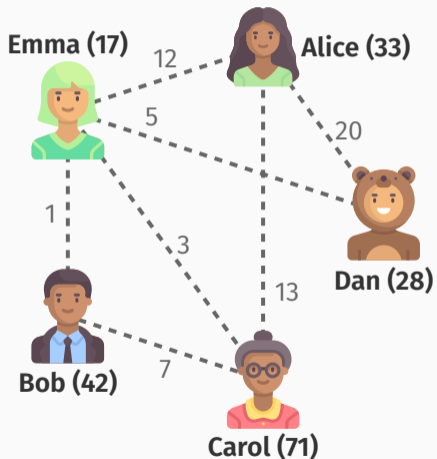
Weight Aggregation Logic

Weighted Structures



- relational structure
- + **weights** for weight symbols in W

Weighted Structures

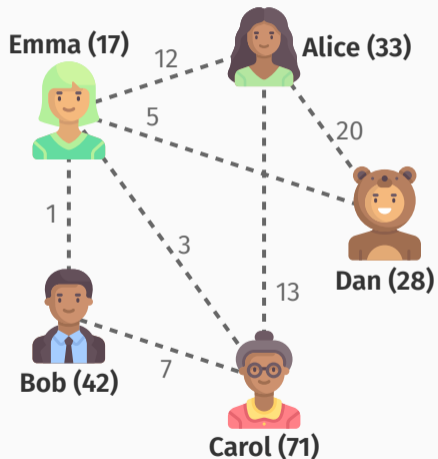


- relational structure
- + **weights** for weight symbols in \mathbf{W}

Example

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

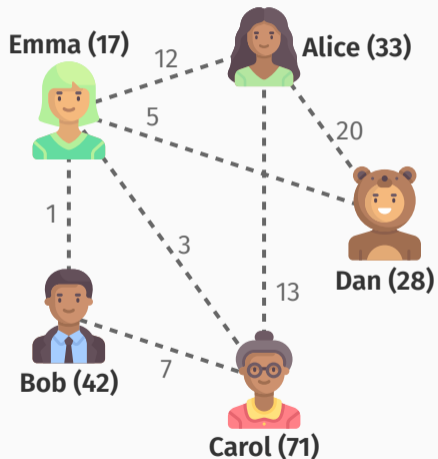
First-Order Logic with Weight Aggregation (FOWA)



Terms

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

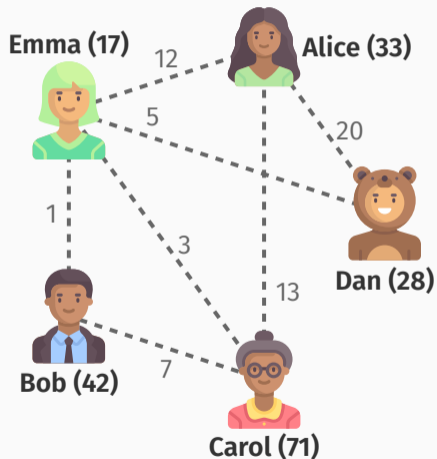
First-Order Logic with Weight Aggregation (FOWA)



Terms

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

First-Order Logic with Weight Aggregation (FOWA)

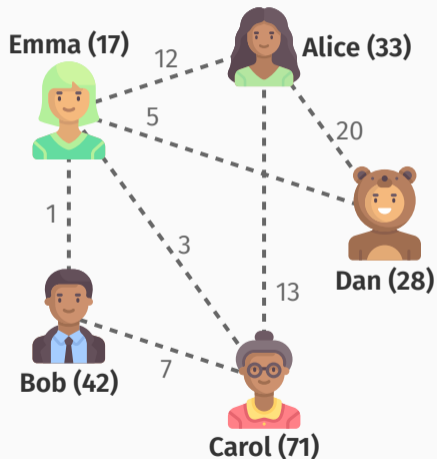


Terms

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

Formulas

First-Order Logic with Weight Aggregation (FOWA)



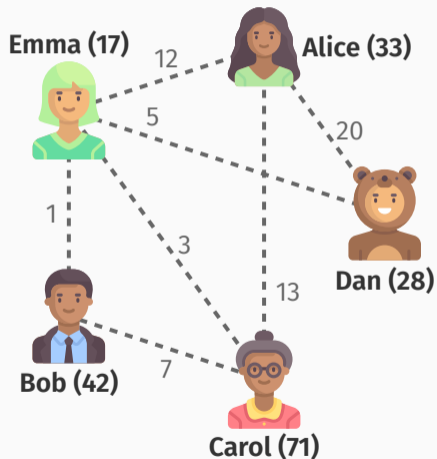
Terms

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

Formulas

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

First-Order Logic with Weight Aggregation (FOWA)



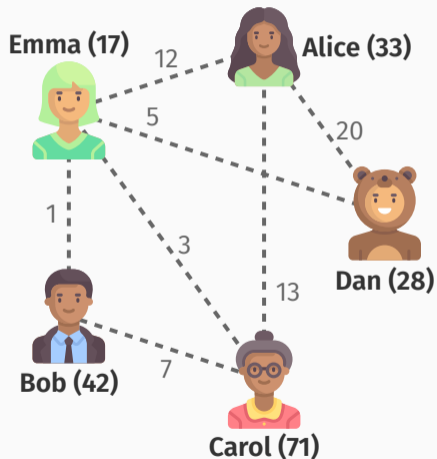
Terms

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

Formulas

- $\varphi_1(x) = (8 = \sum_y \ell(x, y) \cdot F(x, y))$
- $\varphi_2(x) = (\sum_y a(y) \cdot F(x, y) \leq \sum_y a(y) \cdot \neg F(x, y))$

First-Order Logic with Weight Aggregation (FOWA)



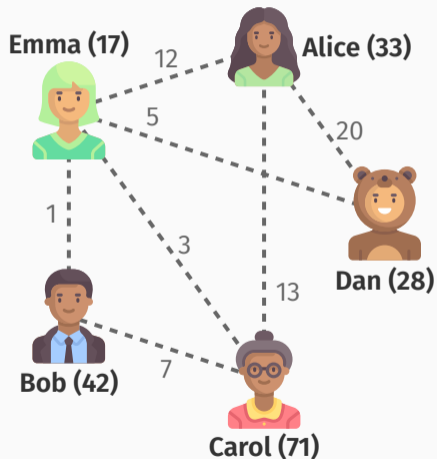
Terms

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

Formulas

- $\varphi_1(x) = (8 = \sum_y \ell(x, y) \cdot F(x, y))$
- $\varphi_2(x) = (\sum_y a(y) \cdot F(x, y) \leq \sum_y a(y) \cdot \neg F(x, y))$
- $\varphi_3(x, y) = (\#(z) \cdot (F(x, z) \wedge F(y, z)) \geq 2)$

First-Order Logic with Weight Aggregation (FOWA)



Terms

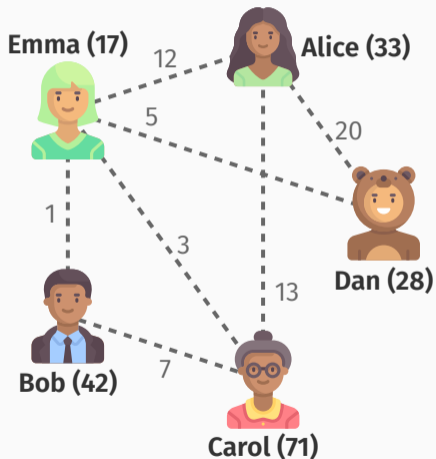
- $t_1(x, y) = 5 \cdot a(x) - a(y) \cdot \ell(x, y)$
- $t_2(x) = \sum_y a(y) \ell(x, y) \cdot F(x, y)$

Formulas

- $\varphi_1(x) = (8 = \sum_y \ell(x, y) \cdot F(x, y))$
- $\varphi_2(x) = (\sum_y a(y) \cdot F(x, y) \leq \sum_y a(y) \cdot \neg F(x, y))$
- $\varphi_3(x, y) = (\#(z) \cdot (F(x, z) \wedge F(y, z)) \geq 2)$

FOWA₁

First-Order Logic with Weight Aggregation (FOWA)



Terms

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

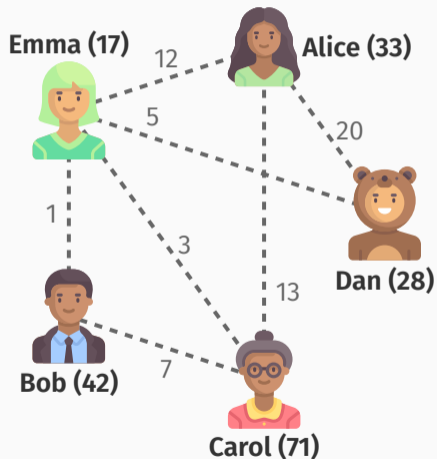
Formulas

- $\varphi_1(x) = (8 = \sum_y \ell(x, y) \cdot F(x, y))$
- $\varphi_2(x) = (\sum_y a(y) \cdot F(x, y) \leq \sum_y a(y) \cdot \neg F(x, y))$
- $\varphi_3(x, y) = (\#(z) \cdot (F(x, z) \wedge F(y, z)) \geq 2)$

FOWA₁

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

First-Order Logic with Weight Aggregation (FOWA)



Terms

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

Formulas

- $\varphi_1(x) = (8 = \sum_y \ell(x, y) \cdot F(x, y))$
- $\varphi_2(x) = (\sum_y a(y) \cdot F(x, y) \leq \sum_y a(y) \cdot \neg F(x, y))$
- $\varphi_3(x, y) = (\#(z) \cdot (F(x, z) \wedge F(y, z)) \geq 2)$

FOWA₁

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

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.

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*.



Localisation Theorem for $FOWA_1$

Use local formula from FOW_1 on enriched structure instead.

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