UNIVERSITEIT GENT

FACULTY OF SCIENCES

Statistical Relational Learning with Soft Quantifiers

Golnoosh Farnadi,
S.H. Bach, M. Blondeel, M-F. Moens, L. Getoor, and M. De Cock

KU LEUVEN

UNIVERSITY OF MARYLAND 1856

IWT

CENTER FOR DATA SCIENCE
UNIVERSITY of WASHINGTON | TACOMA
Institute of Technology

Statistical Relational Learning

- Statistical relational learning (SRL)
  - knowledge representation
  - inference in application domains with uncertain data that is of a complex, relational nature.

- A variety of different SRL frameworks has been developed based on:
  - probabilistic graphical models
  - first-order logic
  - programming languages

MLNs

Probabilistic Soft Logic

Probabilistic Logic Programming

Quantifiers in SRLs

- First-order logic quantifiers:

- Universal quantifiers $\forall$
- Existential quantifiers $\exists$

Limitations of SRLs in addressing quantifiers also addressed in previous works but not as soft quantifiers, such as:

Kazemi et. al, 2014

Poole et. al, 2012

Beltagy et. al, 2015

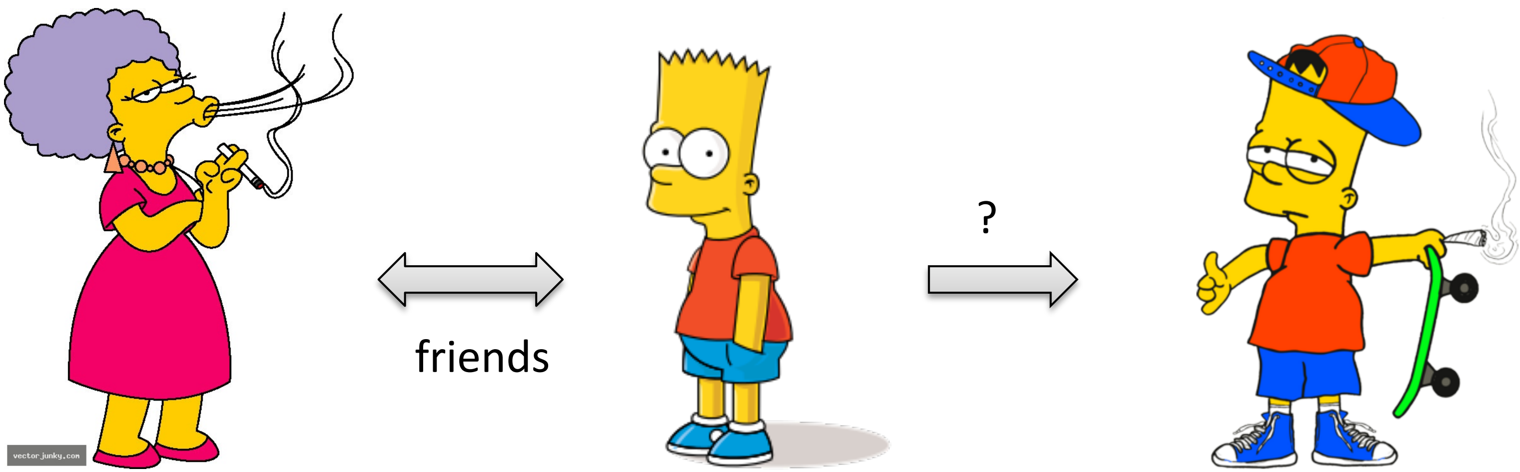
Example

- Smoking example in MLNs:

Richardson & Domingos, 2006

$$\forall X \forall Y \, Friends(X,Y) \rightarrow (Smokes(X) \leftrightarrow Smokes(Y))$$

This formula states that if two people are friends, then either both of them smoke or neither of them



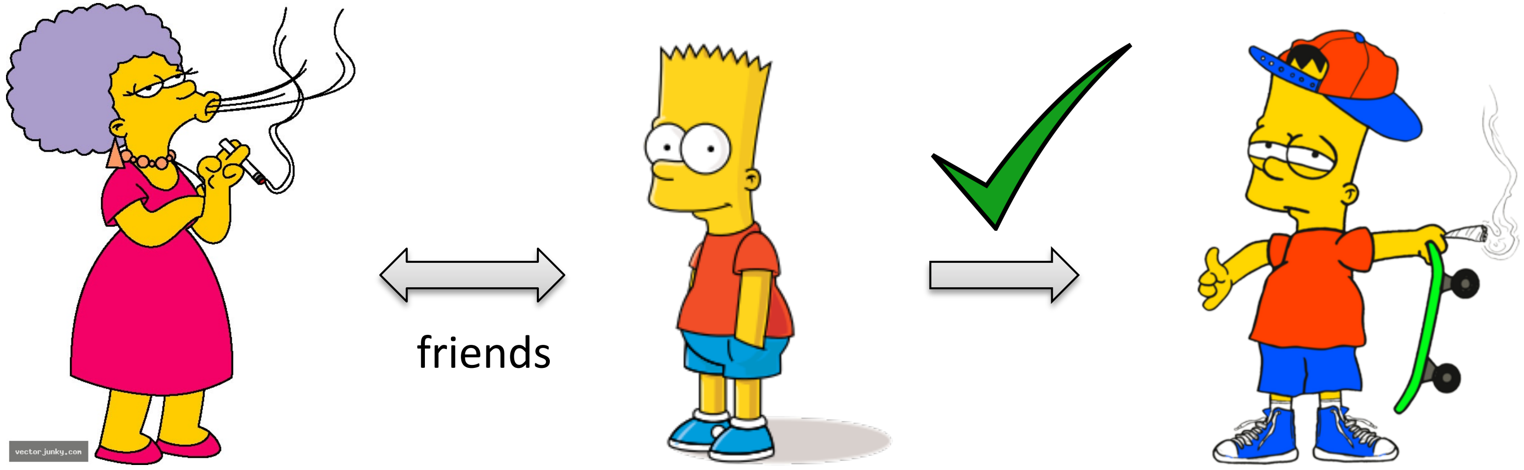
Example

- Smoking example in MLNs:

Richardson & Domingos, 2006

$$\forall X \forall Y\, Friends(X,Y) \rightarrow (Smokes(X) \leftrightarrow Smokes(Y))$$

This formula states that if two people are friends, then either both of them smoke or neither of them



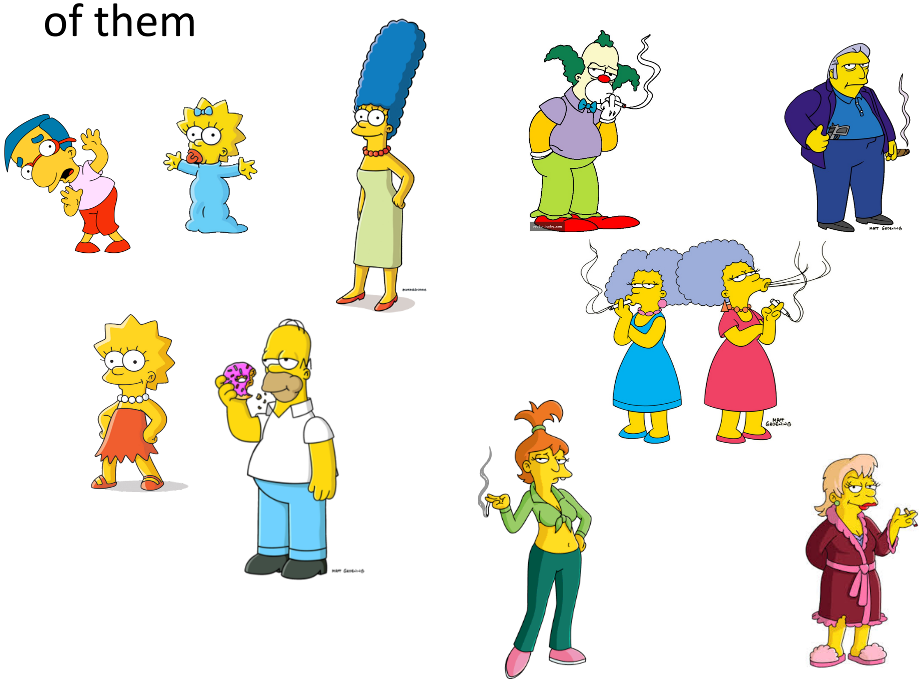
Example

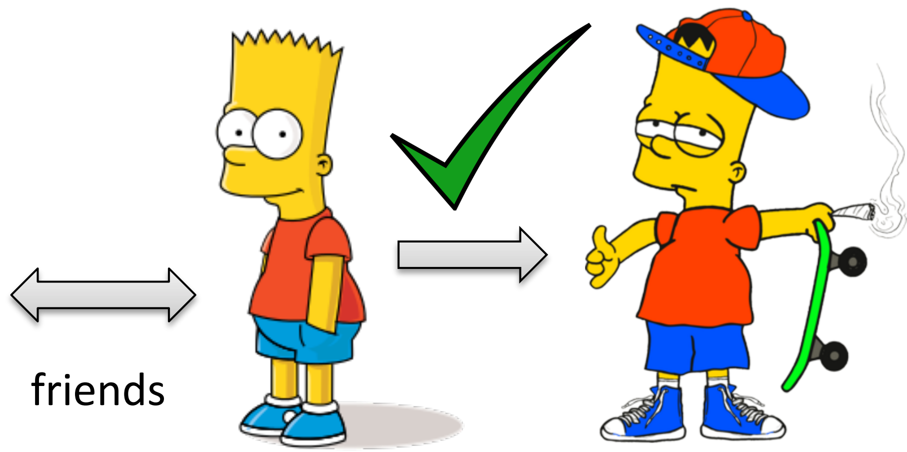
- Smoking example in MLNs:

Richardson & Domingos, 2006

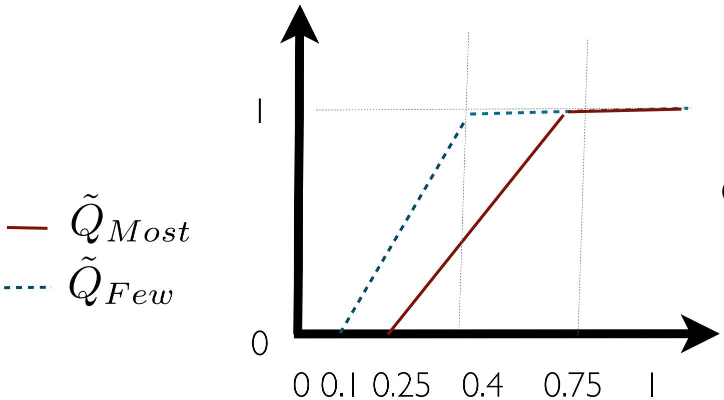
$$\forall X \forall Y\, Friends(X,Y) \rightarrow (Smokes(X) \leftrightarrow Smokes(Y))$$

This formula states that if two people are friends, then either both of them smoke or neither of them





Soft Quantifiers



$\tilde{Q} : [0,1] \rightarrow [0,1]$

- ***Most*** Japanese are …
- ***Some*** jetlagged researchers are …
- ***A few*** researchers at ILP2015 are …

UNIVERSITEIT GENT

FACULTY OF SCIENCES

Probabilistic Soft Logic (PSL) with Soft Quantifiers: **PSL^Q**

MAP Inference and weight learning in PSL^Q

Experimental Results

Probabilistic Soft Logic (PSL)

- We started with PSL framework in which atoms can take continuous values in [0,1]

Broecheler et. al, 2010

- PSL rule:

$$\lambda_r : \underbrace{T_1 \wedge T_2 \wedge \ldots \wedge T_k}_{r_{body}} \rightarrow \underbrace{H_1 \vee H_2 \vee \ldots \vee H_t}_{r_{head}}$$

- Distance to satisfaction of rule r under interpretation I :

$$d_r(I) = \max\{0, I(r_{body}) - I(r_{head})\}$$

How to deal with soft values?

- Lukasiewicz logic:

$$m \tilde{\wedge} n = \max(0, m + n - 1)$$

$$m \tilde{\vee} n = \min(m + n, 1)$$

$$\tilde{\neg} m = 1 - m$$

How to deal with soft values?

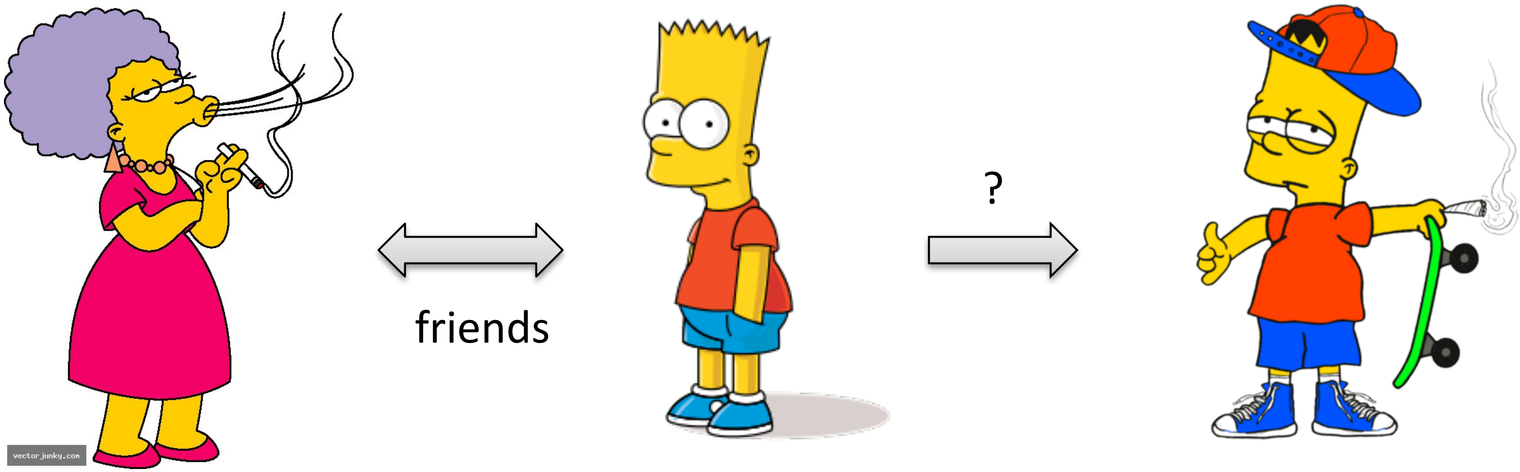
- Lukasiewicz logic:

$$m \tilde{\wedge} n = \max(0, m + n - 1)$$

$$m \tilde{\vee} n = \min(m + n, 1)$$

$$\tilde{\neg} m = 1 - m$$

$$\forall X, Y\, Friends(X, Y) \wedge Smokes(X) \rightarrow Smokes(Y)$$



$$I(Friends(Bart, Patty)) = 0.2$$

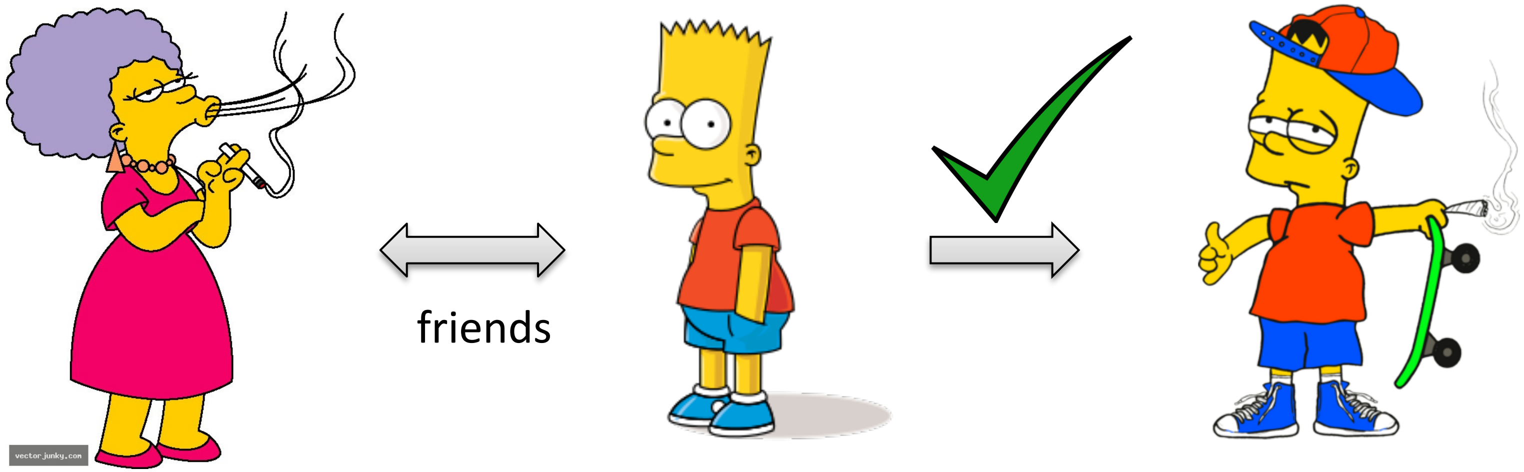
$$I(Smokes(Patty)) = 0.9$$

How to deal with soft values?

- Lukasiewicz logic:

$$m \tilde{\wedge} n = \max(0, m + n - 1)$$
$$m \tilde{\vee} n = \min(m + n, 1)$$
$$\tilde{\neg} m = 1 - m$$

$$\forall X, Y \, Friends(X, Y) \wedge Smokes(X) \rightarrow Smokes(Y)$$

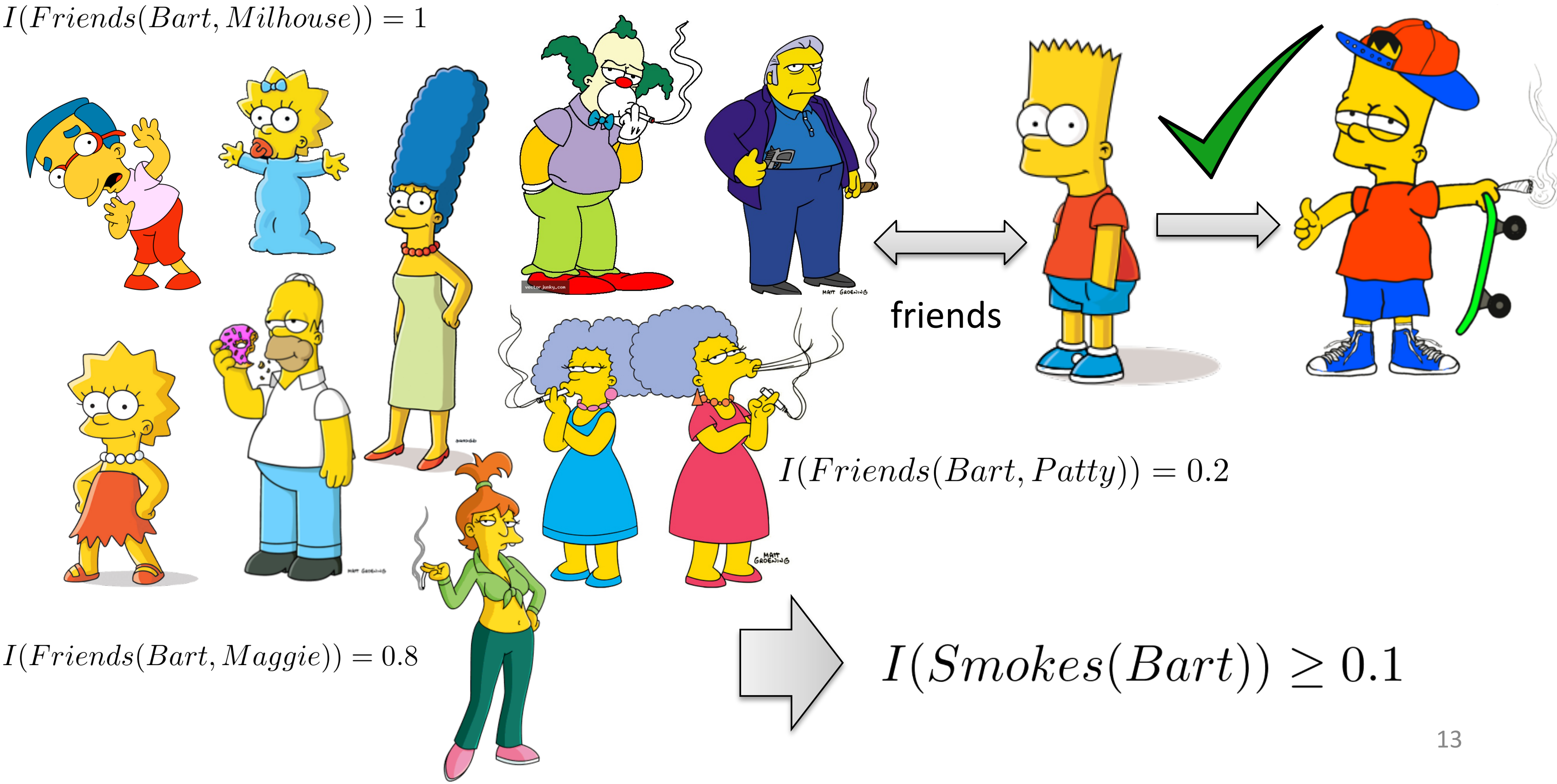


$$I(Friends(Bart, Patty)) = 0.2$$
$$I(Smokes(Patty)) = 0.9$$

$$\Rightarrow \quad I(Smokes(Bart)) \geq 0.1$$

How to use soft quantifiers?

$$\forall X, Y\, Friends(X, Y) \wedge Smokes(X) \rightarrow Smokes(Y)$$



Syntax of PSL^Q

- PSL rule:

$$\lambda_r : \underbrace{T_1 \wedge T_2 \wedge \ldots \wedge T_k}_{r_{body}} \rightarrow \underbrace{H_1 \vee H_2 \vee \ldots \vee H_t}_{r_{head}}$$

Soft **Q**uantifier

$$\mathrm{Q}(\mathrm{X}, F_1(X), F_2(X))$$

- **PSL^Q rule**

Semantics of PSL^Q

Semantic:

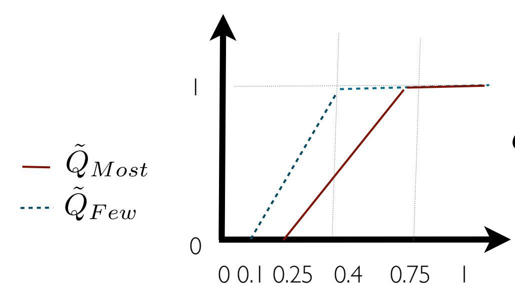
$$\mathrm{I}(\mathrm{Q}(\mathrm{X}, F_1(X), F_2(X))) = \tilde{Q}\left(\frac{\sum_{x \in D} I(F_1(x)) \tilde{\wedge} I(F_2(x))}{\sum_{x \in D} I(F_1(x))}\right)$$

where $\tilde{Q}$ is a soft quantifier mapping, F_1 and F_2 are formulas containing variable X ranging in the domain *D*.

Quantifier mapping $\tilde{Q}$:

$$\tilde{Q}_{[\alpha,\beta]}(x) = \begin{cases} 0 & \text{if } x < \alpha \\ \frac{x-\alpha}{\beta-\alpha} & \text{if } \alpha \leq x < \beta \\ 1 & \text{if } x \geq \beta \end{cases}$$

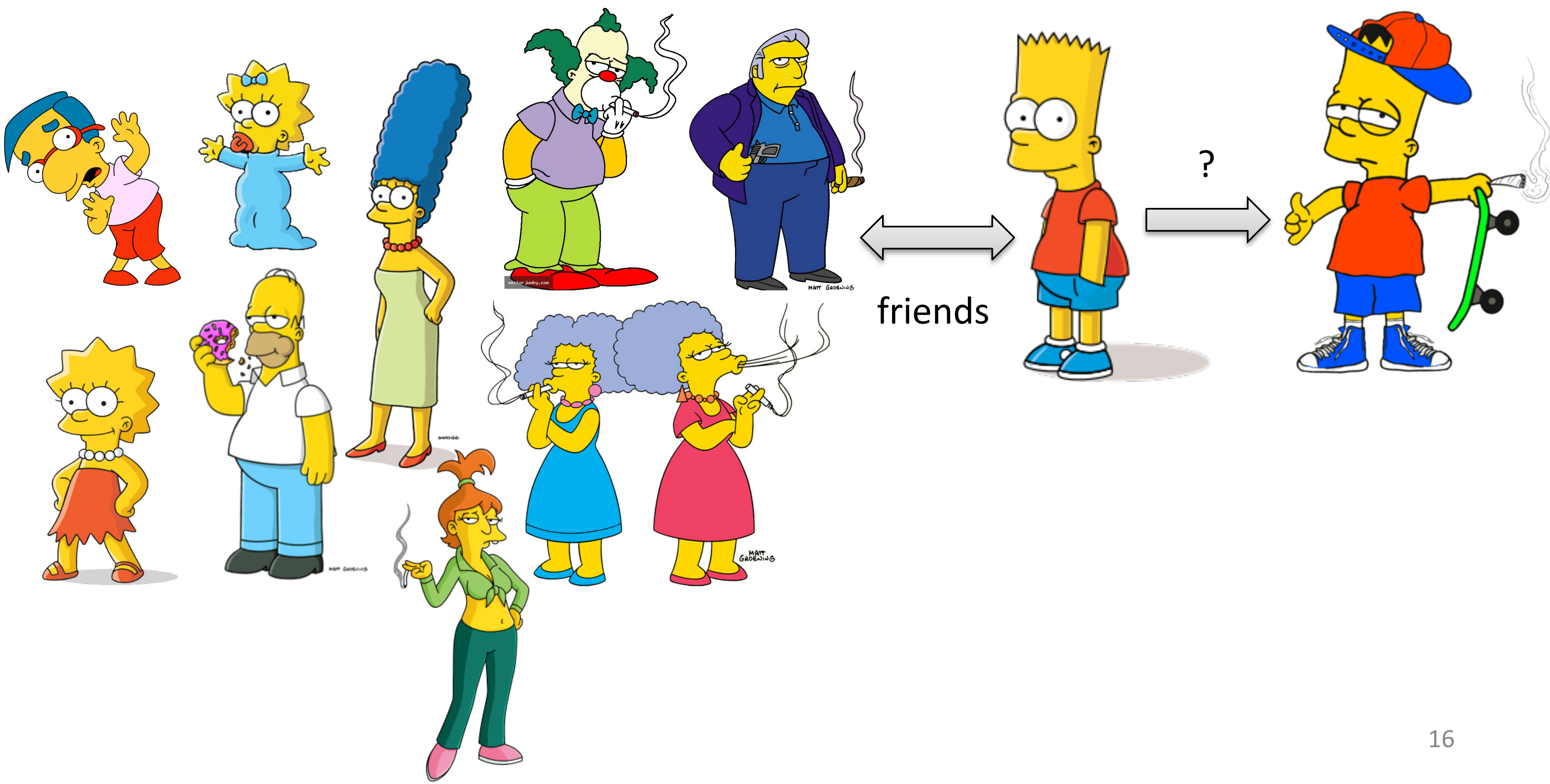
for example



$\tilde{Q} : [0,1] \to [0,1]$

How to use soft quantifiers?

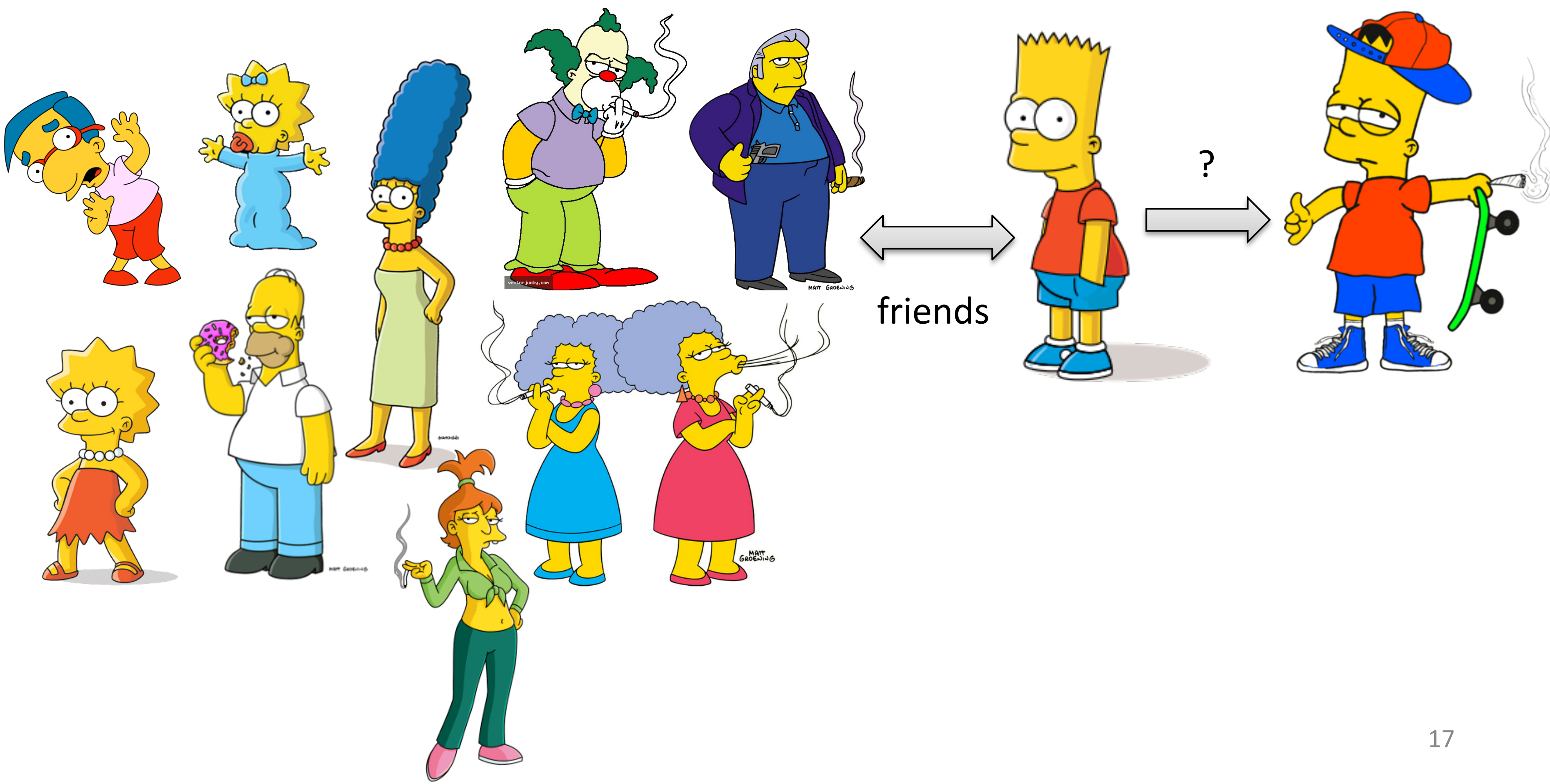
$$\forall X, Y\, Friends(X,Y) \wedge Smokes(X) \rightarrow Smokes(Y)$$



How to use soft quantifiers?

~~$$\forall X, Y\, Friends(X,Y) \wedge Smokes(X) \rightarrow Smokes(Y)$$~~

$$\forall X, Y\, \tilde{Q}_{Most}(X, Friends(X,Y) \wedge Smokes(X)) \rightarrow Smokes(Y)$$

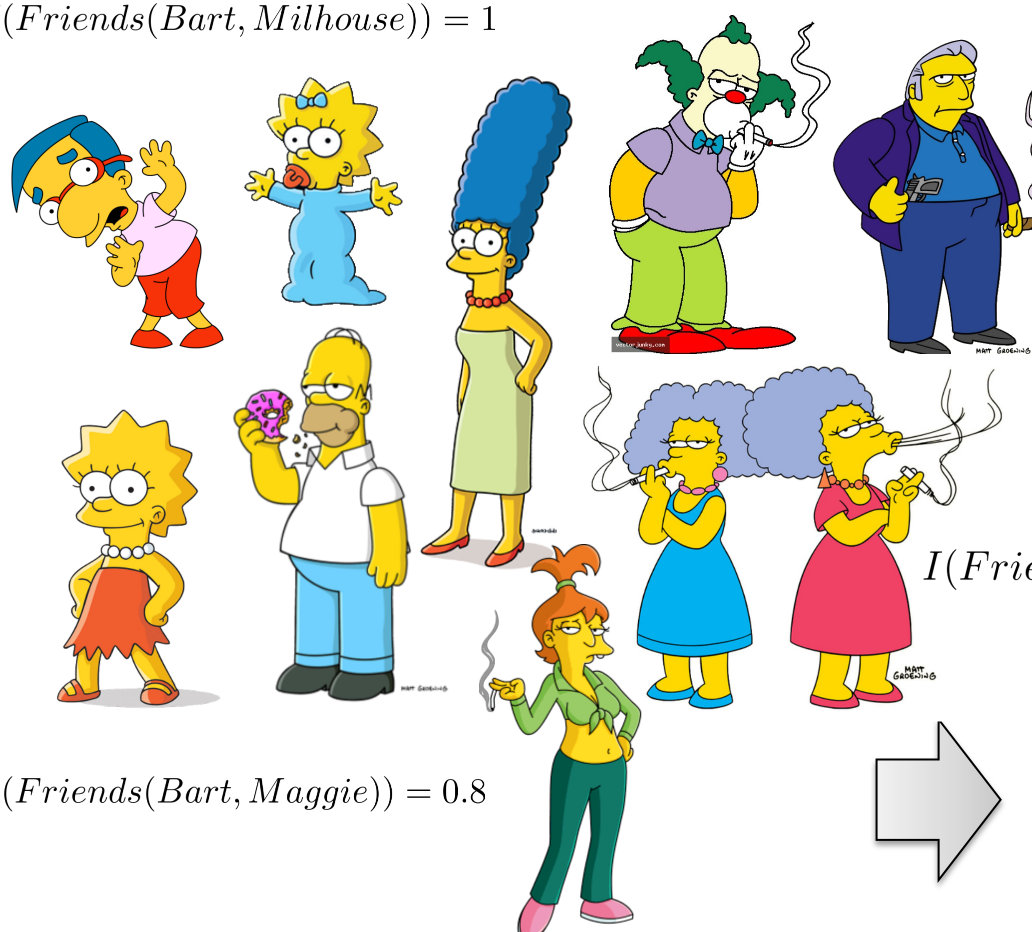


How to use soft quantifiers?

~~$\forall X, Y\, Friends(X,Y) \wedge Smokes(X) \rightarrow Smokes(Y)$~~

$$\forall X, Y\, \tilde{Q}_{Most}(X, Friends(X,Y) \wedge Smokes(X)) \rightarrow Smokes(Y)$$

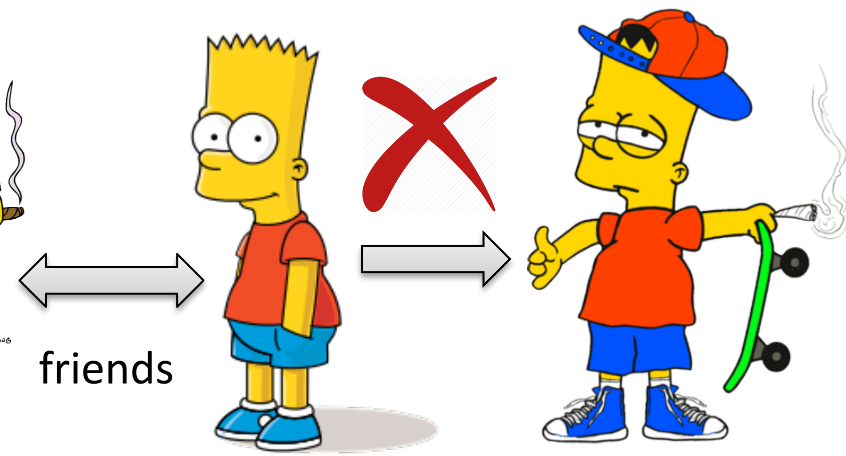
$I(Friends(Bart, Milhouse)) = 1$



$I(Friends(Bart, Patty)) = 0.2$

$I(Friends(Bart, Maggie)) = 0.8$

friends



$$I(Smokes(Bart)) \geq 0.0$$

Probabilistic Soft Logic (PSL) with Soft Quantifiers: **PSL^{Q}**

MAP Inference and weight learning in PSL^{Q}

Experimental Results

MAP inference

Hinge-loss potential function:

Bach. et al., 2013

$$f(I) = \frac{1}{Z} \exp[- \sum_{r \in R} \lambda_r (d_r(I))^p]$$

- $\frac{1}{Z}$: normalization constant
- $\sum_{r \in R}$: all rules in the model
- λ_r: rule's weight
- $d_r(I)$: rule's distance to satisfaction
- p: Choice of the distance metric, e.g., p=1 is linear

The goal of "maximum a posteriori inference" (MAP) is to find the most probable truth assignments of unknown propositions Y given the evidences X.

$$I_{MAP} = \arg\max f(I)$$

The goal of optimization is to minimize the weighted sum of the distances to satisfaction of all rules.

Soft Quantifier Expression is not linear!

- Soft quantifiers are not linear thus cannot be casted as linear constraints:

$$\mathrm{I}(\mathrm{Q}(\mathrm{X}, F_1(X), F_2(X))) = \tilde{Q}\left(\frac{\sum_{x \in D} I(F_1(x)) \tilde{\wedge} I(F_2(x))}{\sum_{x \in D} I(F_1(x))}\right)$$

Fraction of piecewise linear functions

$$\tilde{Q}_{[\alpha,\beta]}(x) = \begin{cases} 0 & \text{if } x < \alpha \\ \frac{x-\alpha}{\beta-\alpha} & \text{if } \alpha \leq x < \beta \\ 1 & \text{if } x \geq \beta \end{cases}$$

Piecewise linear function

Transformation 1: Quantifier mapping

- Quantifier mapping can be rewritten as:

$$\tilde{Q}_{[\alpha,\beta]}(x) = \begin{cases} 0 & \text{if } x < \alpha \\ \frac{x-\alpha}{\beta-\alpha} & \text{if } \alpha \leq x < \beta \\ 1 & \text{if } x \geq \beta \end{cases}$$

⇓

$$\tilde{Q}_{[\alpha,\beta]}(x) = \max(0, \frac{x-\alpha}{\beta-\alpha}) + \min(\frac{x-\alpha}{\beta-\alpha}, 1) - \frac{x-\alpha}{\beta-\alpha}$$

⇓ **min and max transformation**

A set of linear constraints

Transformation 2: FOQE

- Fully observed quantifier expression (**FOQE**):a ground quantifier expression that all ground atoms in F_1 and F_2 are in X .

$$\tilde{Q}\left(\frac{\sum_{x \in D_V} I(F_1(x)) \tilde{\wedge} I(F_2(x))}{\sum_{x \in D_V} I(F_1(x))}\right)$$

constant

constant

constant

Transformation 2: FOQE

- Fully observed quantifier expression (**FOQE**):a ground quantifier expression that all ground atoms in F_1 and F_2 are in X .

$$\tilde{Q}\left(\frac{\sum_{x\in D_V} I(F_1(x))\,\tilde{\wedge}\,I(F_2(x))}{\cdots}\right)$$

constant

Applications:

We aim to use the prior knowledge to infer a new relation or a label.

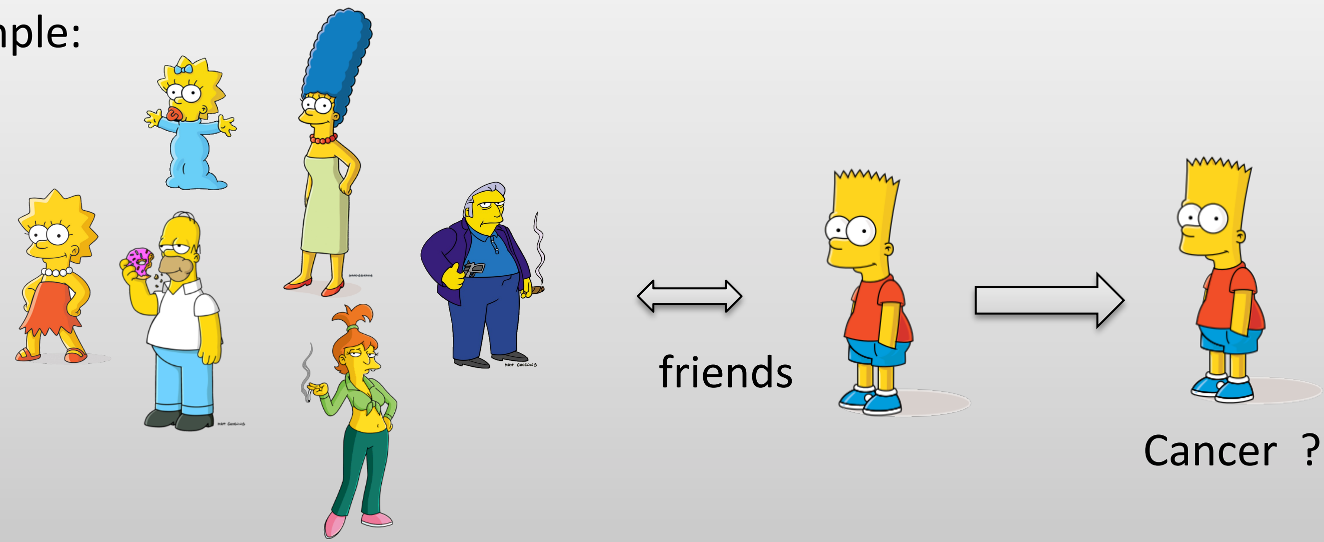
Transformation 2: FOQE

- Fully observed quantifier expression (**FOQE**):a ground quantifier expression that all ground atoms in F_1 and F_2 are in X .

$$\tilde{Q}\left(\sum_{x\in D_V} I(F_1(x))\,\tilde{\wedge}\, I(F_2(x))\right)$$

constant

For Example:



Transformation 3: $\text{POQE}^{(1)}$

- Partially observed quantifier expression of type 1 (**$\text{POQE}^{(1)}$**): a ground quantifier expression that all ground atoms in F_1 are in X.

$$\tilde{Q}\left(\frac{\sum_{x \in D_V} I(F_1(x)) \,\tilde{\wedge}\, I(F_2(x))}{\sum_{x \in D_V} I(F_1(x))}\right)$$

constant

Piecewise linear functions of min and max

Quantifier mapping transformation

min and max transformation

A set of linear constraints

Transformation 3: POQE$^{(1)}$

- Partially observed quantifier expression of type 1 (**POQE$^{(1)}$**): a ground quantifier expression that all ground atoms in F_1 are in X.

$$\tilde{Q}\left(\frac{\sum_{x\in D_V} I(F_1(x))\,\tilde{\wedge}\,I(F_2(x))}{\sum \cdots I(F_1(x))}\right)$$

constant

Applications: **Node labeling**

We have the network (relations) and we aim to infer the labels.

such as, inferring users' characteristics, behaviors, opinions, etc.

Quantifier mapping transformation

min and max transformation

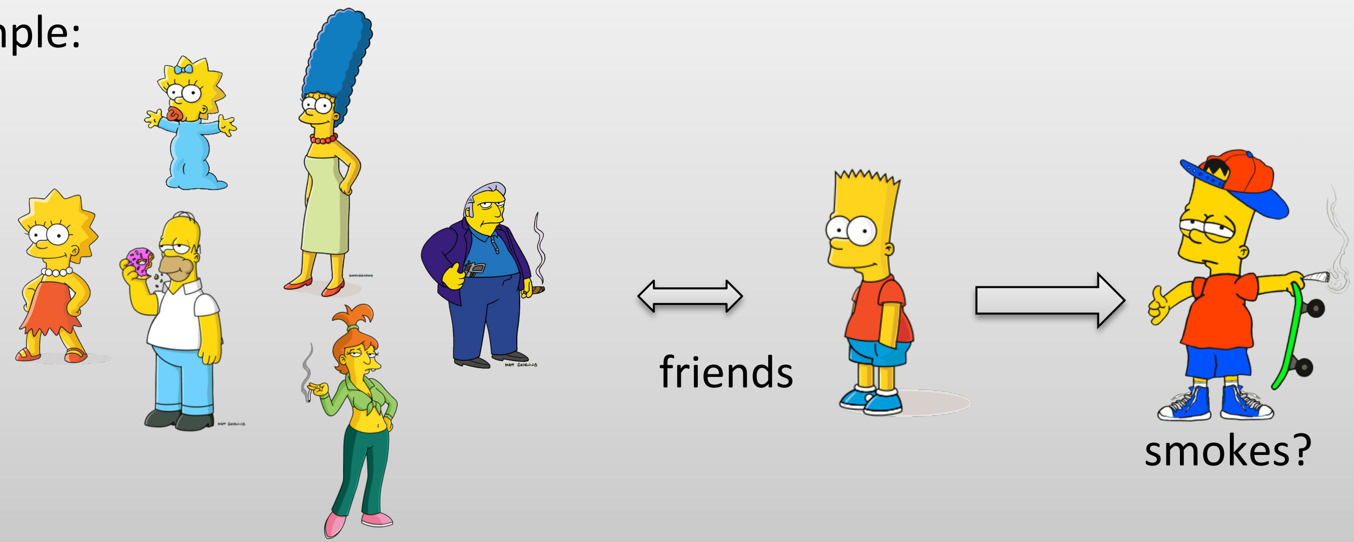
A set of linear constraints

Transformation 3: POQE$^{(1)}$

- Partially observed quantifier expression of type 1 (**POQE$^{(1)}$**): a ground quantifier expression that all ground atoms in F_1 are in X.

$$\tilde{Q}\left(\frac{\sum_{x \in D_V} I(F_1(x)) \,\tilde{\wedge}\, I(F_2(x))}{\cdots} \right)$$

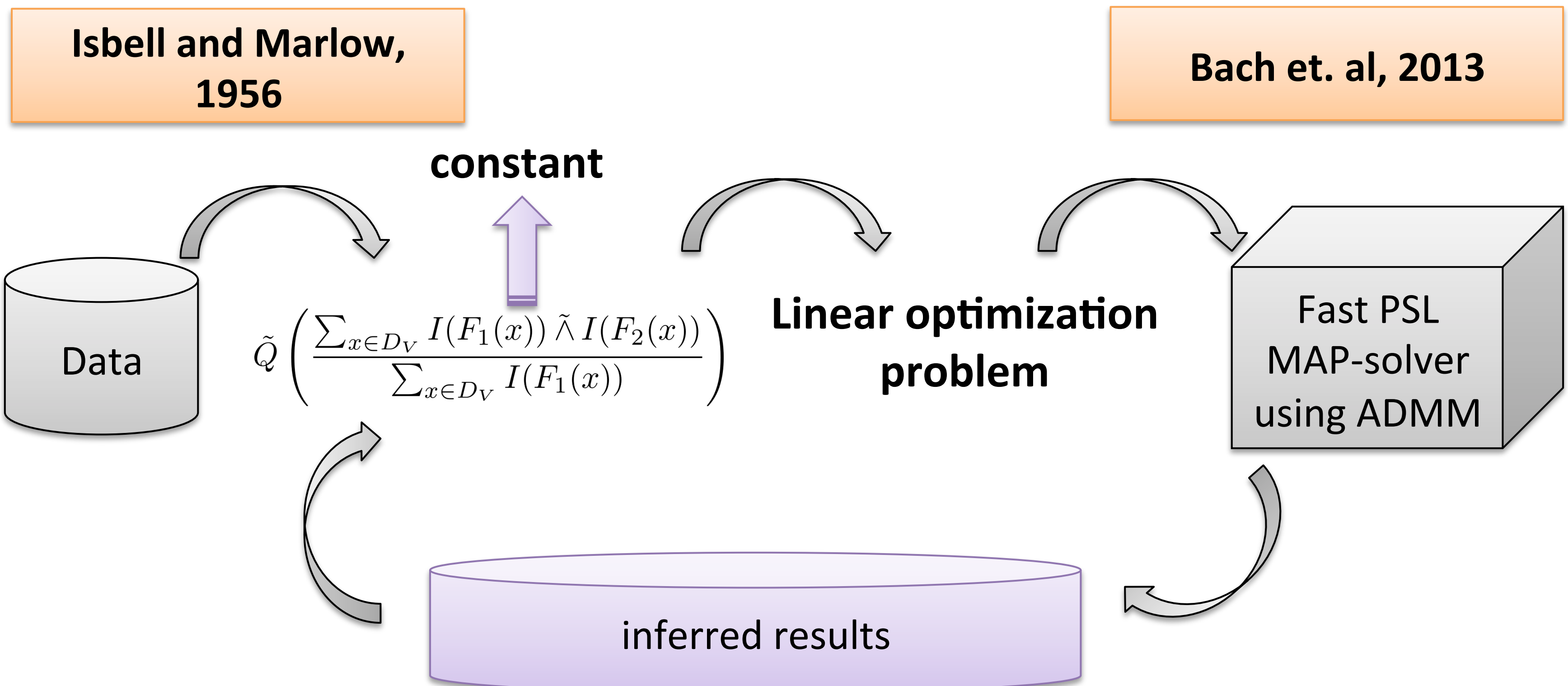
For Example:



A set of linear constraints

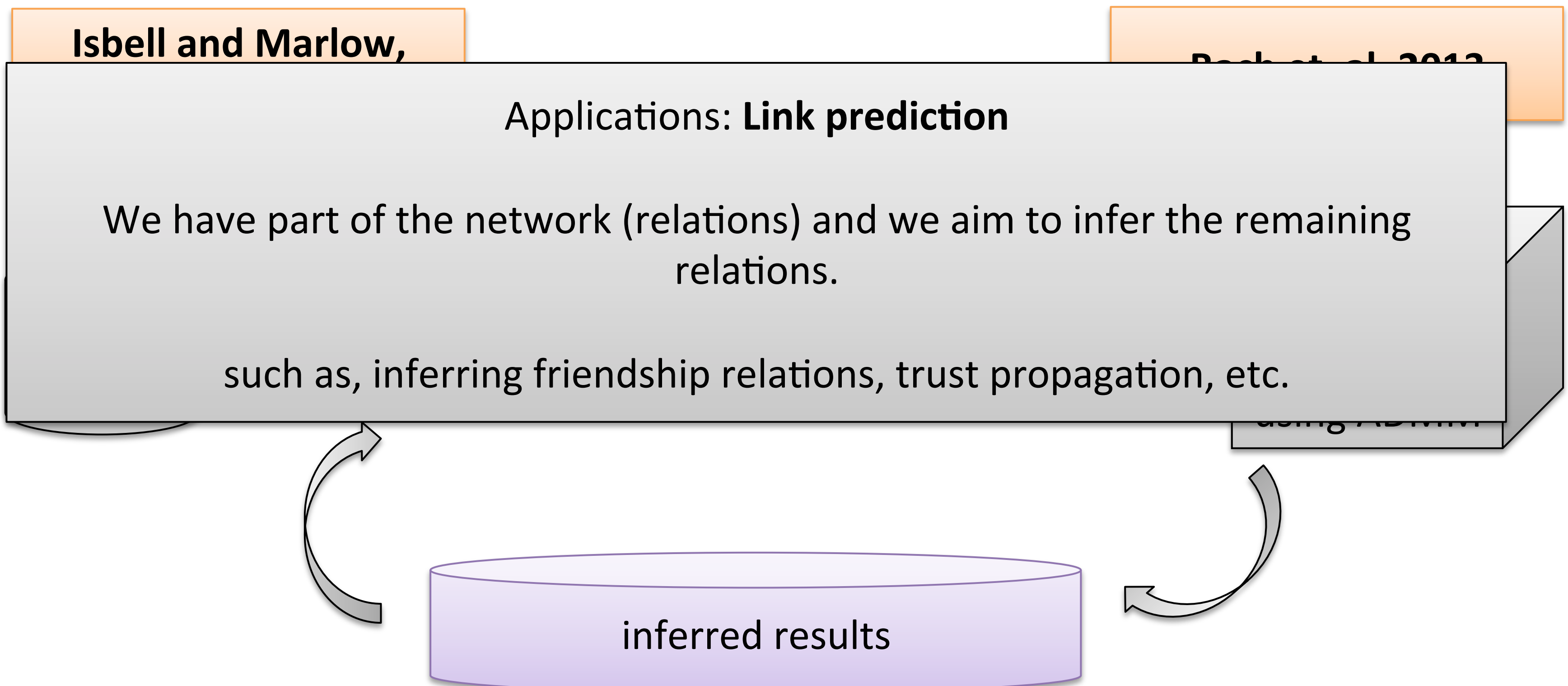
Transformation 4: POQE$^{(2)}$

- Partially observed quantifier expression of type 2 (**POQE$^{(2)}$**): a ground quantifier expression that all ground atoms in F_1 are not in X.



Transformation 4: POQE$^{(2)}$

- Partially observed quantifier expression of type 2 (**POQE$^{(2)}$**): a ground quantifier expression that all ground atoms in F_1 are not in X.



Applications: **Link prediction**

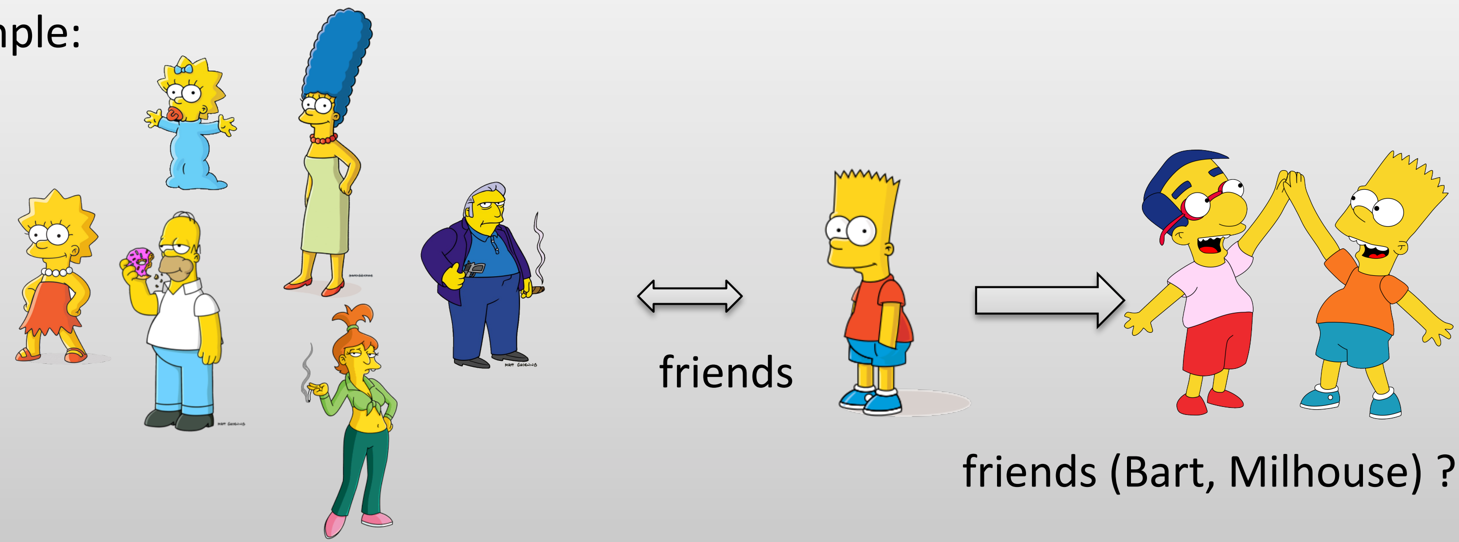
We have part of the network (relations) and we aim to infer the remaining relations.

such as, inferring friendship relations, trust propagation, etc.

Transformation 4: POQE$^{(2)}$

- Partially observed quantifier expression of type 2 (**POQE$^{(2)}$**): a ground quantifier expression that all ground atoms in F_1 are not in X.

For Example:



inferred results

Weight learning

The goal of weight learning based on maximum likelihood estimation (MLE) is to maximize the log likelihood of the rules' weight based on the training data:

$$-\frac{\delta log(f(I))}{\delta \lambda_i} = E_\lambda[\sum_{r \in R_g i} (d_r(I))^p] - \sum_{r \in R_g i} (d_r(I))^p$$

Collins, 2002

1. The optimization is based on the voted perception algorithm

2. To make the approximation tractable, a MPE approximation is used.

Probabilistic Soft Logic (PSL) with Soft Quantifiers: **PSL^Q**

MAP Inference and weight learning in PSL^Q

Experimental Results

Epinions.com

Social Trust

- Epinions dataset

Our sample dataset contains **2000 users** from Epinions.com. They are connected with **8,675** relations: **7,974 trust** relations and **701 distrust** relations.

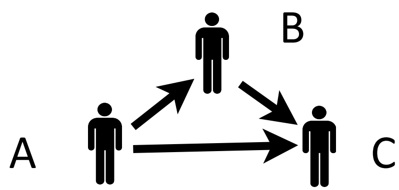
PSL rule vs. PSL^{Q} rule

Heider, 1958

Structural Balance theory implies the transitivity of a relation between users

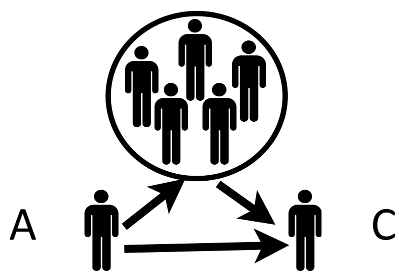
Huang et. al, 2012

- PSL rule



$$Knows(A,B) \wedge Trusts(A,B) \wedge Knows(B,C) \wedge Trusts(B,C) \wedge Knows(A,C) \rightarrow Trusts(A,C)$$

- PSL^{Q} rule

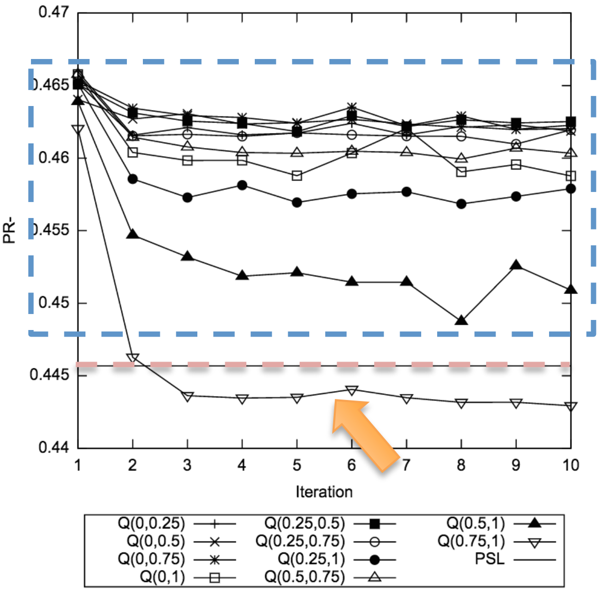


$$Q(X, Knows(A,X) \wedge Trusts(A,X), Knows(X,C) \wedge Trusts(X,C)) \wedge Knows(A,C) \rightarrow Trusts(A,C)$$

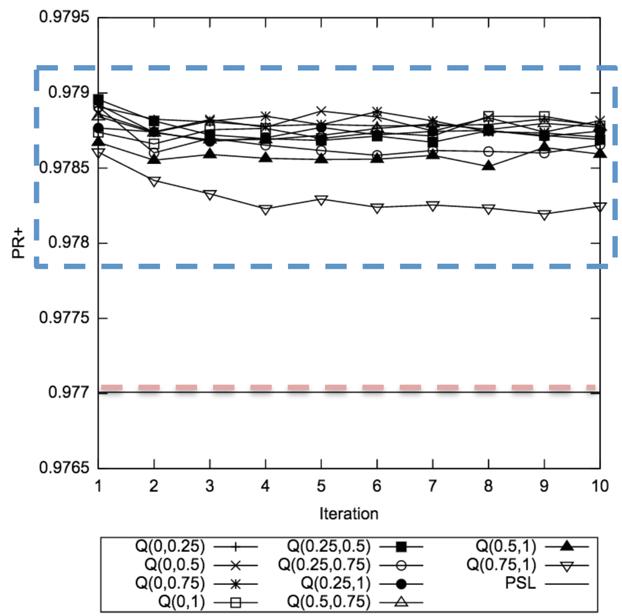
PSL^Q Model

	Transitive rules
(R#1)	$Knows(A,B) \wedge Trusts(A,B) \wedge Knows(B,C) \wedge Trusts(B,C) \wedge Knows(A,C) \rightarrow Trusts(A,C)$
(R#2)	$Knows(A,B) \wedge \neg Trusts(A,B) \wedge Knows(B,C) \wedge Trusts(B,C) \wedge Knows(A,C) \rightarrow \neg Trusts(A,C)$
(R#3)	$Knows(A,B) \wedge Trusts(A,B) \wedge Knows(B,C) \wedge \neg Trusts(B,C) \wedge Knows(A,C) \rightarrow \neg Trusts(A,C)$
(R#4)	$Knows(A,B) \wedge \neg Trusts(A,B) \wedge Knows(B,C) \wedge \neg Trusts(B,C) \wedge Knows(A,C) \rightarrow Trusts(A,C)$
	Cyclic rule
(R#5)	$Knows(A,B) \wedge Trusts(A,B) \wedge Knows(B,C) \wedge Trusts(B,C) \wedge Knows(C,A) \rightarrow Trusts(C,A)$
	Complementary rules
(R#6)	$Knows(A,B) \wedge Knows(B,A) \wedge Trusts(B,A) \rightarrow Trusts(A,B)$
(R#7)	$Knows(A,B) \wedge Knows(B,A) \wedge \neg Trusts(B,A) \rightarrow \neg Trusts(A,B)$
(R#8)	$Knows(A,B) \wedge Average(\{Trusts\}) \rightarrow Trusts(A,B)$
(R#9)	$Knows(A,B) \wedge Trusts(A,B) \rightarrow Average(\{Trusts\})$
	PSL^Q rules based on the transitive rules
(R#10)	$Q(X, Knows(A,X) \wedge Trusts(A,X), Knows(X,C) \wedge Trusts(X,C)) \wedge Knows(A,C) \rightarrow Trusts(A,C)$
(R#11)	$Q(X, Knows(A,X) \wedge \neg Trusts(A,X), Knows(X,C) \wedge Trusts(X,C)) \wedge Knows(A,C) \rightarrow \neg Trusts(A,C)$
(R#12)	$Q(X, Knows(A,X) \wedge Trusts(A,X), Knows(X,C) \wedge \neg Trusts(X,C)) \wedge Knows(A,C) \rightarrow \neg Trusts(A,C)$
(R#13)	$Q(X, Knows(A,X) \wedge \neg Trusts(A,X), Knows(X,C) \wedge \neg Trusts(X,C)) \wedge Knows(A,C) \rightarrow Trusts(A,C)$
	PSL^Q rule based on the cyclic rule
(R#14)	$Q(X, Knows(A,X) \wedge Trusts(A,X), Knows(X,C) \wedge Trusts(X,C)) \wedge Knows(C,A) \rightarrow Trusts(C,A)$

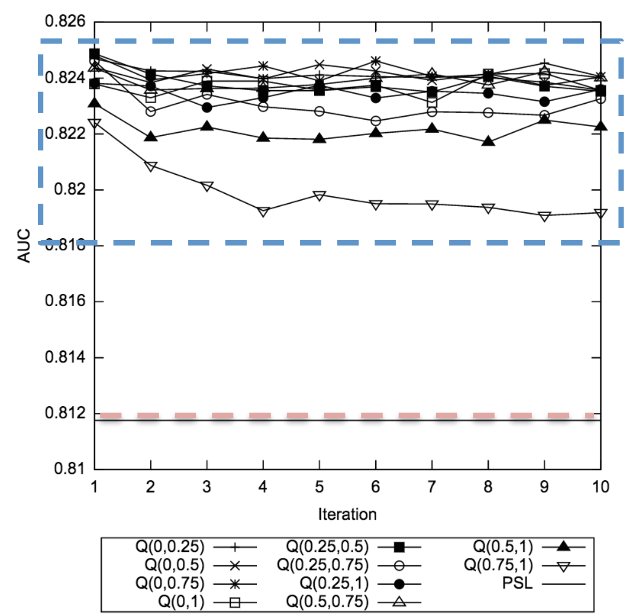
Effects of Quantifier Mapping



PR-



PR+

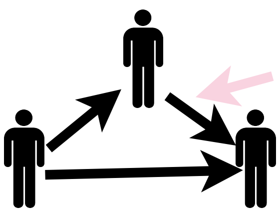


AUC

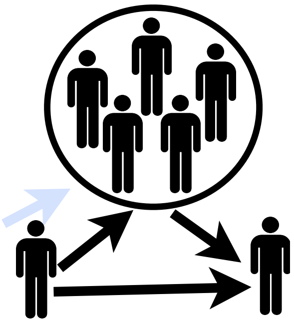
Experimental results

We systematically perform 8-fold cross-validation and to evaluate the results.

We first learn the weights of the rules based on 7/8 of the trust network and then apply the learned model on the remaining 1/8 to infer the trust/distrust relations.

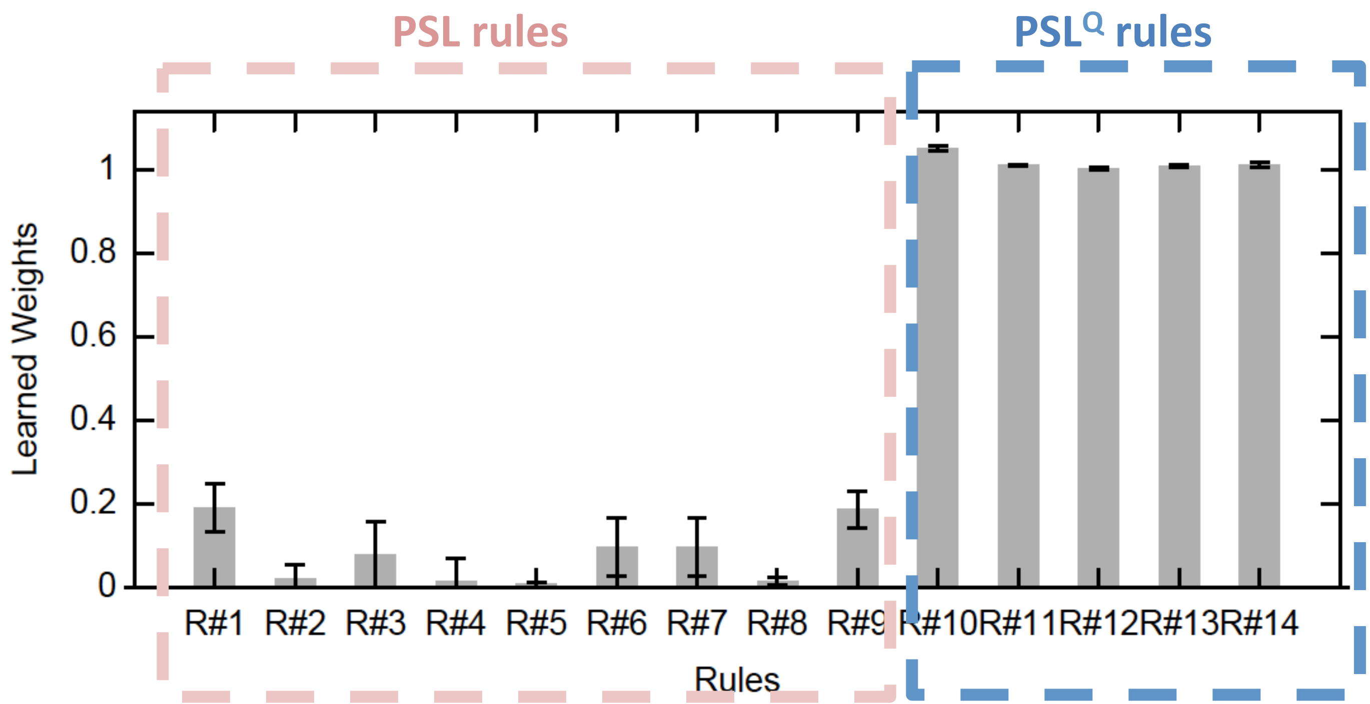


Model	PR+	PR-	AUC
MLN	0.942	0.270	0.655
PSL	0.977	0.446	0.812
PSLQ	**0.979**	**0.467**	**0.825**



statistically significant with a rejection threshold of 0.05

Rules' Weights



Using soft quantifiers not only improves the accuracy of trust and distrust predictions but also the rules containing soft quantifiers, i.e. rules 10-14, play a major part in this by dominating all other rules in terms of weight.

Future directions

- Besides social trust, many other AI applications could benefit from the use of soft quantifiers.
- We defined the semantics of a quantifier expression using the approach of Zadeh. Studying other approaches for quantifiers is a direction for our future work.
- Automatic way of interpreting the quantifier mapping
- New approaches of inference and weight learning for PSL^Q

Collaborators

Martine De Cock

Lise Getoor

Marie-Francine Moens

Stephen Bach

Marjon Blondeel

Code will be available soon!

ありがとう

Thank you.

See you at the poster session.

golnoosh.farnadi@ugent.be