Statistical Relational Learning with Soft Quantifiers

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

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

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

• Smoking example in MLNs:

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

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

• Smoking example in MLNs:

\[ \forall X \forall Y \text{Friends}(X, Y) \rightarrow (\text{Smokes}(X) \leftrightarrow \text{Smokes}(Y)) \]

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

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

\[ \tilde{Q} : [0, 1] \rightarrow [0, 1] \]
Probabilistic Soft Logic (PSL) with Soft Quantifiers: $\text{PSL}^Q$

MAP Inference and weight learning in $\text{PSL}^Q$

Experimental Results
Probabilistic Soft Logic (PSL)

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

- PSL rule:
  \[ \lambda_r : T_1 \land T_2 \land \ldots \land T_k \rightarrow H_1 \lor H_2 \lor \ldots \lor H_t \]

- Distance to satisfaction of rule \( \mathcal{r} \) under interpretation \( I \):
  \[ d_r(I) = \max\{0, I(r_{body}) - I(r_{head})\} \]

Broecheler et al., 2010
How to deal with soft values?

- Lukasiewicz logic:

\[
m\land n = \max(0, m + n - 1) \\
m\lor n = \min(m + n, 1) \\
\neg m = 1 - m
\]
How to deal with soft values?

• Lukasiewicz logic:

\[
\forall X, Y \text{Friends}(X, Y) \land \text{Smokes}(X) \rightarrow \text{Smokes}(Y)
\]

\[
m\land n = \max(0, m + n - 1)
\]

\[
m\lor n = \min(m + n, 1)
\]

\[
\neg m = 1 - m
\]

\[
I(\text{Friends(Bart, Patty)}) = 0.2
\]

\[
I(\text{Smokes(Patty)}) = 0.9
\]
How to deal with soft values?

- Lukasiewicz logic:
  \[ m \wedge n = \max(0, m + n - 1) \]
  \[ m \vee n = \min(m + n, 1) \]
  \[ \neg m = 1 - m \]

\[ \forall X, Y \text{Friends}(X, Y) \land \text{Smokes}(X) \rightarrow \text{Smokes}(Y) \]

\[ I(\text{Friends}(\text{Bart, Patty})) = 0.2 \]
\[ I(\text{Smokes}(\text{Patty})) = 0.9 \]

\[ I(\text{Smokes}(\text{Bart})) \geq 0.1 \]
How to use soft quantifiers?

\[ \forall X, Y \text{Friends}(X, Y) \land \text{Smokes}(X) \rightarrow \text{Smokes}(Y) \]

\[ I(\text{Friends}(\text{Bart, Milhouse})) = 1 \]

\[ I(\text{Friends}(\text{Bart, Patty})) = 0.2 \]

\[ I(\text{Friends}(\text{Bart, Maggie})) = 0.8 \]

\[ I(\text{Smokes}(\text{Bart})) \geq 0.1 \]
Syntax of PSL^{Q}

• PSL rule:

\[ \lambda_r : T_1 \land T_2 \land \ldots \land T_k \rightarrow H_1 \lor H_2 \lor \ldots \lor H_t \]

\[ r_{body} \quad r_{head} \]

• PSL^{Q} rule

Soft Quantifier

\[ Q(X, F_1(X), F_2(X)) \]
Semantics of PSL$^Q$

**Semantic:**

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

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

$\tilde{Q} : [0, 1] \rightarrow [0, 1]$
How to use soft quantifiers?

\[ \forall X, Y \text{Friends}(X, Y) \land \text{Smokes}(X) \rightarrow \text{Smokes}(Y) \]
How to use soft quantifiers?

$$\forall X, Y \text{Friends}(X, Y) \land \text{Smokes}(X) \rightarrow \text{Smokes}(Y)$$

$$\forall X, Y \tilde{Q}_{\text{Most}}(X, \text{Friends}(X, Y) \land \text{Smokes}(X)) \rightarrow \text{Smokes}(Y)$$
How to use soft quantifiers?

$$\forall X, Y \text{Friends}(X, Y) \land \text{Smokes}(X) \rightarrow \text{Smokes}(Y)$$

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

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

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

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

$I$(Smokes(Bart)) $\geq$ 0.0
Probabilistic Soft Logic (PSL) with Soft Quantifiers: $\text{PSL}^Q$

MAP Inference and weight learning in $\text{PSL}^Q$

Experimental Results
**MAP inference**

**Hinge-loss potential function:**

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

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.

Choice of the distance metric, e.g., \( p=1 \) is linear

Bach. et al., 2013
Soft Quantifier Expression is not linear!

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

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

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 $X$.

$$\tilde{Q} \left( \frac{\sum_{x \in D_V} I(F_1(x)) \wedge I(F_2(x))}{\sum_{x \in D_V} I(F_1(x))} \right)$$
Transformation 2: FOQE

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

\[
\tilde{Q} \left( \sum_{x \in D_Y} I(F_1(x)) \land I(F_2(x)) \right) \text{ constant}
\]

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 $X$.

\[
\tilde{\phi} \left( \sum_{x \in D_Y} I(F_1(x)) \land I(F_2(x)) \right) \text{ constant}
\]

For Example:

\[\text{friends} \quad \overset{\text{↔}}{\quad} \quad \text{Cancer ?}\]
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 $X$.

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

constant

Piecewise linear functions of min and max

Quantifier mapping transformation

A set of linear constraints

min and max transformation
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 \(X\).

\[
\tilde{Q} \left( \sum_{x \in D_V} I(F_1(x)) \, \tilde{\land} \, I(F_2(x)) \right)
\]

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 \(X\).

\[
\tilde{Q}\left(\sum_{x \in D_V} I(F_1(x)) \land I(F_2(x))\right)
\]

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 \(X\).

\[
\tilde{Q} \left( \frac{\sum_{x \in D^V} I(F_1(x)) \land I(F_2(x))}{\sum_{x \in D^V} I(F_1(x))} \right)
\]

Isbell and Marlow, 1956

Bach et. al, 2013

Data

Constant

Linear optimization problem

Fast PSL MAP-solver using ADMM

Inferred results
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:

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

.png
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:

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

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: \( \text{PSL}^Q \)

MAP Inference and weight learning in \( \text{PSL}^Q \)

Experimental Results
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\textsuperscript{Q} rule

**Structural Balance theory** implies the transitivity of a relation between users

- **PSL rule**

  \[\text{Knows}(A, B) \land \text{Trusts}(A, B) \land \text{Knows}(B, C) \land \text{Trusts}(B, C) \land \text{Knows}(A, C) \rightarrow \text{Trusts}(A, C)\]

- **PSL\textsuperscript{Q} rule**

  \[Q(X, \text{Knows}(A, X) \land \text{Trusts}(A, X), \text{Knows}(X, C) \land \text{Trusts}(X, C)) \land \text{Knows}(A, C) \rightarrow \text{Trusts}(A, C)\]

Heider, 1958

Huang et. al, 2012
# PSL\(^Q\) Model

<table>
<thead>
<tr>
<th>Rule</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>(R#1)</td>
<td>( \text{Knows}(A, B) \land \text{Trusts}(A, B) \land \text{Knows}(B, C) \land \text{Trusts}(B, C) \land \text{Knows}(A, C) \rightarrow \text{Trusts}(A, C) )</td>
</tr>
<tr>
<td>(R#2)</td>
<td>( \text{Knows}(A, B) \land \neg \text{Trusts}(A, B) \land \text{Knows}(B, C) \land \text{Trusts}(B, C) \land \text{Knows}(A, C) \rightarrow \neg \text{Trusts}(A, C) )</td>
</tr>
<tr>
<td>(R#3)</td>
<td>( \text{Knows}(A, B) \land \text{Trusts}(A, B) \land \text{Knows}(B, C) \land \neg \text{Trusts}(B, C) \land \text{Knows}(A, C) \rightarrow \neg \text{Trusts}(A, C) )</td>
</tr>
<tr>
<td>(R#4)</td>
<td>( \text{Knows}(A, B) \land \neg \text{Trusts}(A, B) \land \text{Knows}(B, C) \land \neg \text{Trusts}(B, C) \land \text{Knows}(A, C) \rightarrow \neg \text{Trusts}(A, C) )</td>
</tr>
<tr>
<td>(R#5)</td>
<td>( \text{Knows}(A, B) \land \text{Trusts}(A, B) \land \text{Knows}(B, C) \land \text{Trusts}(B, C) \land \text{Knows}(C, A) \rightarrow \text{Trusts}(C, A) )</td>
</tr>
<tr>
<td>Complementary rules</td>
<td></td>
</tr>
<tr>
<td>(R#6)</td>
<td>( \text{Knows}(A, B) \land \text{Knows}(B, A) \land \text{Trusts}(B, A) \rightarrow \text{Trusts}(A, B) )</td>
</tr>
<tr>
<td>(R#7)</td>
<td>( \text{Knows}(A, B) \land \text{Knows}(B, A) \land \neg \text{Trusts}(B, A) \rightarrow \neg \text{Trusts}(A, B) )</td>
</tr>
<tr>
<td>(R#8)</td>
<td>( \text{Knows}(A, B) \land \text{Average}({\text{Trusts}}) \rightarrow \text{Trusts}(A, B) )</td>
</tr>
<tr>
<td>(R#9)</td>
<td>( \text{Knows}(A, B) \land \text{Trusts}(A, B) \rightarrow \text{Average}({\text{Trusts}}) )</td>
</tr>
</tbody>
</table>

**PSL\(^Q\) rules based on the transitive rules**

<table>
<thead>
<tr>
<th>Rule</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>(R#10)</td>
<td>( Q(X, \text{Knows}(A, X) \land \text{Trusts}(A, X), \text{Knows}(X, C) \land \text{Trusts}(X, C)) \land \text{Knows}(A, C) \rightarrow \text{Trusts}(A, C) )</td>
</tr>
<tr>
<td>(R#11)</td>
<td>( Q(X, \text{Knows}(A, X) \land \neg \text{Trusts}(A, X), \text{Knows}(X, C) \land \text{Trusts}(X, C)) \land \text{Knows}(A, C) \rightarrow \neg \text{Trusts}(A, C) )</td>
</tr>
<tr>
<td>(R#12)</td>
<td>( Q(X, \text{Knows}(A, X) \land \text{Trusts}(A, X), \text{Knows}(X, C) \land \neg \text{Trusts}(X, C)) \land \text{Knows}(A, C) \rightarrow \neg \text{Trusts}(A, C) )</td>
</tr>
<tr>
<td>(R#13)</td>
<td>( Q(X, \text{Knows}(A, X) \land \neg \text{Trusts}(A, X), \text{Knows}(X, C) \land \neg \text{Trusts}(X, C)) \land \text{Knows}(A, C) \rightarrow \text{Trusts}(A, C) )</td>
</tr>
</tbody>
</table>

**PSL\(^Q\) rule based on the cyclic rule**

<table>
<thead>
<tr>
<th>Rule</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>(R#14)</td>
<td>( Q(X, \text{Knows}(A, X) \land \text{Trusts}(A, X), \text{Knows}(X, C) \land \text{Trusts}(X, C)) \land \text{Knows}(C, A) \rightarrow \text{Trusts}(C, A) )</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Model</th>
<th>PR+</th>
<th>PR-</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLN</td>
<td>0.942</td>
<td>0.270</td>
<td>0.655</td>
</tr>
<tr>
<td>PSL</td>
<td>0.977</td>
<td>0.446</td>
<td>0.812</td>
</tr>
<tr>
<td>PSLQ</td>
<td>0.979</td>
<td>0.467</td>
<td>0.825</td>
</tr>
</tbody>
</table>

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 $\text{PSL}^Q$
Collaborators

Martine De Cock
Lise Getoor
Marie-Francine Moens
Stephen Bach
Marjon Blondeel
Thank you.

See you at the poster session.

golnoosh.farnadi@ugent.be