Learning and Reasoning in Logic Tensor Networks

Luciano Serafini\(^1\), Ivan Donadello\(^1;2\), Artur d’Avila Garces\(^3\)

\(^1\)Fondazione Bruno Kessler, Italy
\(^2\)University of Trento, Italy
\(^3\)City University London, UK

May 7, 2017
Statistical Relational Learning is a subdiscipline of artificial intelligence that is concerned with domain models that exhibit both uncertainty and complex relational structure.
Hybrid domains

We are interested in Statistical Relational Learning over **hybrid domains**, i.e., domains that are characterized by the presence of

- structured data (categorical/semantic);
- continuous data (continuous features);
Hybrid domains

Example (SRL domain)

- Kurt
  - years: 34
  - age: 2/2/95
  - date: 2/2/95
  - livesIn: Rome
  - town: Detroit
  - area: 53.72 km²

- Car2
  - dollar: 15342
  - income: 10000
  - price: 130.00
  - engine power: 10000
  - company: FCA
  - locatedIn: Detroit

- Location
  - Rome
  - Detroit

- Entities
  - person: Kurt
  - car: Car2

- Relations
  - livesIn: Kurt → Rome
  - owns: Kurt → Car2
  - town: Rome → Detroit
  - locatedIn: Detroit → Car2

- Attributes
  - dollar: 15342
  - income: 10000
  - price: 130.00
  - engine power: 10000
  - area: 53.72 km²
Tasks in Statistical Relational Learning

- **Object Classification:** Predicting the type of an object based on its relations and attributes;

- **Relation detection:** Predicting if two objects are connected by a relation, based on types and attributes of the participating objects;

- **Regression:** Predicting the (distribution of) values of the attributes of an object, (a pair of related objects) based on the types and relations of the object(s) involved.

Example (SRL domain)
Real-world uncertain, structured and hybrid domains

**Robotics:** a robot's location is a continuous values while the **types of the objects it encounters** can be described by discrete set of classes

**Semantic Image Interpretation:** The visual features of a bounding box of a picture are continuous values, while the **types of objects contained** in a bounding box and the **relations between them** are taken from a discrete set

**Natural Language Processing:** The **distributional semantics** provide a vectorial (numerical) representation of the meaning of words, while WordNet associates to each word a set of synsets and a set of relations with other words which are finite and discrete
Semantic Image interpretation

Semantic Image Interpretation (SII)

detect the main objects shown in the picture;
assign to each object an object type;
determine the relations between the objects as shown in the picture
represent the outcome of the detection in a semantic structure.
Semantic Image interpretation

**semantic Image Interpretation (SII)**

- detect the **main objects** shown in the picture;
Semantic Image interpretation

Semantic Image Interpretation (SII)

- detect the **main objects** shown in the picture;
- assign to each object an **object type**;

![Semantic Image Interpretation Example]

Luciano Serafini, Ivan Donadello, Artur d’Avil
Learning and Reasoning in Logic Tensor Network
Semantic Image interpretation

**Semantic Image Interpretation (SII)**

- detect the **main objects** shown in the picture;
- assign to each object an **object type**;
- determine the **relations** between the objects as shown in the picture;
- represent the outcome of the detection in a **semantic structure**.

---

Luciano Serafini, Ivan Donadello, Artur d’Avila Garces (Fondazione Bruno Kessler, Italy University of Trento, Italy City University London, UK)

Learning and Reasoning in Logic Tensor Networks

May 7, 2017
Language - to specify knowledge about models

Two sorted first order language: (abstract sort and numeric sort)

- Abstract constant symbols ($b_1, b_2, \ldots, b_8$);
- Abstract relation symbols ($player(x), ball(x)$, $partOf(x,y), hasNum(x,y)$);
- Numeric function symbols ($xBL(x), yBL(x), width(x), height(h)$, $area(x), color(x), contRatio(x,y)$);

COLOR CODE:

- denotes objects and relations of the domain structure;
- denotes attributes and relations between attributes of the numeric part of the domain.
Example (Domain description:)

knowledge about object detection:
\( xBL(b_1) = 23, yBL(b_1) = 73, \)
\( width(b_1) = 20, height(b_1) = 21 \)
\( xBL(b_2) = 45, yBL(b_1) = 70, \)
\( width(b_1) = 40, height(b_1) = 104 \ldots \)
\( contRatio(b_2, b_4) = 1.0, \)
\( contRatio(b_2, b_5) = 0.4, \ldots \)
Example (Domain description:

knowledge about object detection:
\( xBL(b_1) = 23, yBL(b_1) = 73, \)
\( width(b_1) = 20, height(b_1) = 21 \)
\( xBL(b_2) = 45, yBL(b_1) = 70, \)
\( width(b_1) = 40, height(b_1) = 104 \ldots \)
\( contRatio(b_2, b_4) = 1.0, \)
\( contRatio(b_2, b_5) = 0.4, \ldots \)

partial knowledge about object types and relations

\( ball(b_1), player(b_2), player(b_3), \)
\( leg(b_4), leg(b_5), partOf(b_3, b_2), \)
\( kicks(b_2, b_1), hasNum(b_3, b_7), \ldots \)
Domain description and queries

Example (Domain description:)

knowledge about object detection:

\[ xBL(b_1) = 23, \quad yBL(b_1) = 73, \]
\[ width(b_1) = 20, \quad height(b_1) = 21 \]
\[ xBL(b_2) = 45, \quad yBL(b_1) = 70, \]
\[ width(b_1) = 40, \quad height(b_1) = 104 \ldots \]
\[ contRatio(b_2, b_4) = 1.0, \]
\[ contRatio(b_2, b_5) = 0.4, \ldots \]

partial knowledge about object types and relations

\[ ball(b_1), \quad player(b_2), \quad player(b_3), \]
\[ leg(b_4), \quad leg(b_5), \quad partOf(b_3, b_2), \]
\[ kicks(b_2, b_1), \quad hasNum(b_3, b_7), \ldots \]

ontological axioms

\[ \forall xy. partOf(x, y) \land leg(x) \to player(y), \]
\[ \forall xy. kick(x, y) \to player(x) \land ball(y), \]
\[ \forall xy. partOf(x, y) \to contRatio(x, y) > 0.9 \]
\[ \forall x. player(x) \to \neg ball(x), \]
Domain description and queries

Example (Domain description:)
knowledge about object detection:
\[ xBL(b_1) = 23, \ yBL(b_1) = 73, \]
\[ width(b_1) = 20, \ height(b_1) = 21 \]
\[ xBL(b_2) = 45, \ yBL(b_1) = 70, \]
\[ width(b_1) = 40, \ height(b_1) = 104 \ldots \]
\[ contRatio(b_2, b_4) = 1.0, \]
\[ contRatio(b_2, b_5) = 0.4, \ldots \]
partial knowledge about object types and relations
\[ ball(b_1), \ player(b_2), \ player(b_3), \]
\[ leg(b_4), \ leg(b_5), \ partOf(b_3, b_2), \]
\[ kicks(b_2, b_1), \ hasNum(b_3, b_7), \ldots \]

Example (Queries)
Query about missing knowledge about object types and relations
\[ player(b_{10}) \]
\[ xBL(b_{10}) = 83, \]
\[ yBL(b_{10}) = 42, \]
\[ width(b_{10}) = 30 \ldots \]
\[ partOf(b_{10}, b_{11}) \]
\[ xBL(b_{10}) = 83, \]
\[ yBL(b_{11}) = 42, \]
\[ width(b_{11}) = 30 \ldots \]
\[ contRatio(b_{10}, b_{11}) = 0.6 \]
\[ contRatio(b_{11}, b_{10}) = 0.9 \ldots \]

ontological axioms
\[ \forall xy. \ partOf(x, y) \land leg(x) \rightarrow player(y), \]
\[ \forall xy. \ kick(x, y) \rightarrow player(x) \land ball(y), \]
\[ \forall xypartOf(x, y) \rightarrow contRatio(x, y) > .9 \]
\[ \forall xplayer(x) \rightarrow \neg ball(x), \]
Logic Tensor Network basic idea

Logic Tensor Network that computes the truth value of the formula $\phi(x, y)$ on the basis of the numeric features of $x, y$ and the pair $\langle x, y \rangle$.
Logic Tensor Network basic idea

\[ \phi(x, y) \]

Network for fuzzy logic

Deep Neural networks that compute the values of all the atomic formulas composing \( \phi(x, y) \) starting from the numeric features

\[ f(x) \quad f(y) \quad g(x) \quad g(y) \quad h(x, y) \quad h(y, x) \]
**LTN for predicates**

\( n \) unary numeric function \( f_1(x), \ldots, f_n(x) \) and \( m \) binary numeric function \( g_1(x, y), \ldots, g_m(x, y) \)

**LTN for unary predicate/type \( P(x) \)**

\[
\text{LTN}_P(v) = \sigma(u_P^\top \tanh(v^\top W_P^{[1:k]} v + V_P v + b_P))
\]

\( w_P \in \mathbb{R}^{k \times n \times n}, V_P \in \mathbb{R}^{k \times n}, b_P \in \mathbb{R}^k, \text{ and } u_P \in \mathbb{R}^k \) are parameters.

**LTN for binary relation \( R(x, y) \)**

\[
\text{LTN}_P(v) = \sigma(u_P^\top \tanh(v^\top W_P^{[1:k]} v + V_P v + b_P))
\]

\( w_P \in \mathbb{R}^{k \times h \times h}, V_P \in \mathbb{R}^{k \times h}, b_P \in \mathbb{R}^k, \text{ and } u_P \in \mathbb{R}^k \) are parameters, and \( h = 2(n + m) = \) the total number of numeric features that can be obtained applying \( f_i \) and \( g_i \) to \( x \) and \( y \).
Fuzzy semantics for propositional connectives

- In fuzzy semantics, atoms are assigned with some **truth value** in real interval $[0,1]$
- Connectives have functional semantics. E.g., a binary connective $\circ$ must be interpreted in a function $f_\circ : [0, 1]^2 \rightarrow [0, 1]$.
- Truth values are ordeblue, i.e., if $x > y$, then $x$ is a stronger truth than $y$
- Generalization of classical propositional logic:
  
  0 corresponds to FALSE and
  1 corresponds to TRUE
### Fuzzy semantics for connectives and quantifiers

<table>
<thead>
<tr>
<th>T-norm</th>
<th>$a \land b$</th>
<th>$\max(0, a + b - 1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T-conorm</td>
<td>$a \lor b$</td>
<td>$\min(1, a + b)$</td>
</tr>
<tr>
<td>residual</td>
<td>$a \rightarrow$</td>
<td>$\begin{cases} 1 - a + b &amp; \text{if } a &gt; b \ 1 &amp; \text{if } a \leq b \end{cases}$</td>
</tr>
<tr>
<td>precomplement</td>
<td>$\neg a$</td>
<td>$1 - a$</td>
</tr>
<tr>
<td>aggregation</td>
<td>$\forall x. a(x)$</td>
<td>$\lim_{n \to \infty} \left( \frac{1}{n} \sum_{i=1}^{n} (a(i) - 1)^{-1}\right)$</td>
</tr>
</tbody>
</table>

Alternatively, use Gödel or Product T-norm, and geometric or arithmetic mean as aggregator.
Constructive semantics for Existential quantifier

- LTN interprets existential quantifiers constructively via Skolemization.
- Every formula $\forall x_1, \ldots, x_n \exists y \phi(x_1, \ldots, x_n, y)$ is rewritten as $\forall x_1, \ldots, x_{m} \phi(x_1, \ldots, x_n, f(x_1, \ldots, x_{m}))$,
- by introducing a new $m$-ary function symbol $f$,

**Example**

$$\forall x. (\text{cat}(x) \rightarrow \exists y. \text{partof}(y, x) \land \text{tail}(y))$$

is transformed in

$$\forall x (\text{cat}(x) \rightarrow \text{partOf}(\text{tailOf}(x), x) \land \text{tail}(\text{tailOf}(x)))$$
Grounding = relation between logical symbols and data

\[ G(P(v, u) \rightarrow A(u)) \]

\[ v = \langle v_1, \ldots, v_n \rangle \]

\[ u = \langle u_1, \ldots, u_n \rangle \]
Grounding = relation between logical symbols and data

\[ G(P(v, u) \rightarrow A(u)) \]

\[ G(\neg P(v, u)) \]

\[ G(A(u)) \]

\[ v = \langle v_1, \ldots, v_n \rangle \]

\[ u = \langle u_1, \ldots, u_n \rangle \]
Parameter learning = best satisfiability

Given a FOL theory $K$ the **best satisfiability problem** as the problem of finding the set of parameters $\Theta$ of the LTN, then the problems become

$$g^* = LTN(K, \Theta^*)$$

$$\Theta^* = \arg\max_{\Theta} \left( \min_{K \models \phi} LTN(K, \Theta)(\phi) \right)$$

Luciano Serafini, Ivan Donadello, Artur d’Avila Garces (Fondazione Bruno Kessler, Italy University of Trento, Italy City University London, UK)
Learning from model description and answering queries

\[ \Theta^* = \arg\max_{\Theta} \left( \min_K | \phi \right) \]

\[ x_{BL}(b_1) = 23, \quad y_{BL}(b_1) = 73, \]
\[ width(b_1) = 20, \quad height(b_1) = 21 \]
\[ x_{BL}(b_2) = 45, \quad y_{BL}(b_1) = 70, \]
\[ width(b_1) = 40, \quad height(b_1) = 104 \ldots \]
\[ \text{contRatio}(b_2, b_4) = 1.0, \quad \text{contRatio}(b_2, b_5) = 0.4, \ldots \]
\[ \text{ball}(b_1), \quad \text{player}(b_2), \quad \text{player}(b_3), \]
\[ \text{leg}(b_4), \quad \text{leg}(b_5), \quad \text{partOf}(b_3, b_2), \]
\[ \text{kicks}(b_2, b_1), \quad \text{hasNum}(b_3, b_7), \ldots \]
\[ \forall xy. \text{partOf}(x, y) \land \text{leg}(x) \rightarrow \text{player}(y), \]
\[ \forall xy. \text{kick}(x, y) \rightarrow \text{player}(x) \land \text{ball}(y), \]
\[ \forall xy. \text{partOf}(x, y) \rightarrow \text{contRatio}(x, y) > .9 \]
\[ \forall x. \text{player}(x) \rightarrow \neg \text{ball}(x), \]
Learning from model description and answering queries

\[ \Theta^* = \arg\max_\Theta \left( \min_{K \models \phi} LTN(K, \Theta)(\phi) \right) \]

\[ \begin{align*}
 xBL(b_1) &= 23, \quad yBL(b_1) = 73, \\
 width(b_1) &= 20, \quad height(b_1) = 21 \\
 xBL(b_2) &= 45, \quad yBL(b_1) = 70, \\
 width(b_1) &= 40, \quad height(b_1) = 104 \ldots \\
 \text{contRatio}(b_2, b_4) &= 1.0, \quad \text{contRatio}(b_2, b_5) = 0.4, \ldots \\
 \text{ball}(b_1), \quad \text{player}(b_2), \quad \text{player}(b_3), \\
 \text{leg}(b_4), \quad \text{leg}(b_5), \quad \text{partOf}(b_3, b_2), \\
 \text{kicks}(b_2, b_1), \quad \text{hasNum}(b_3, b_7), \ldots \\
 \forall xy. \text{partOf}(x, y) \land \text{leg}(x) \rightarrow \text{player}(y), \\
 \forall xy. \text{kick}(x, y) \rightarrow \text{player}(x) \land \text{ball}(y), \\
 \forall xy. \text{partOf}(x, y) \rightarrow \text{contRatio}(x, y) > .9 \\
 \forall x. \text{player}(x) \rightarrow \neg \text{ball}(x),
\end{align*} \]
Learning from model description and answering queries

\[
\Theta^* = \arg\max_{\Theta} \left( \min_{K \models \phi} LTN(K, \Theta)(\phi) \right)
\]

Given a set of objects and their properties, we can use Logic Tensor Networks (LTN) to learn the model description and answer queries. The diagram illustrates the process:

- \( K \): Knowledge base
- \( Q \): Query

The table shows examples of properties and objects:

<table>
<thead>
<tr>
<th>Property</th>
<th>Object 1</th>
<th>Object 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>( xBL(b_1) = 23 ), ( yBL(b_1) = 73 ), width(b_1) = 20, height(b_1) = 21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( xBL(b_2) = 45 ), ( yBL(b_1) = 70 ), width(b_1) = 40, height(b_1) = 104 \ldots</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \text{contRatio}(b_2, b_4) = 1.0 ), ( \text{contRatio}(b_2, b_5) = 0.4 ), \ldots</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \text{ball}(b_1), \text{player}(b_2), \text{player}(b_3), \text{leg}(b_4), \text{leg}(b_5), \text{partOf}(b_3, b_2), \text{kicks}(b_2, b_1), \text{hasNum}(b_3, b_7), \ldots )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \forall xy. \text{partOf}(x, y) \wedge \text{leg}(x) \rightarrow \text{player}(y), \forall xy. \text{kick}(x, y) \rightarrow \text{player}(x) \wedge \text{ball}(y), \forall xy. \text{partOf}(x, y) \rightarrow \text{contRatio}(x, y) &gt; .9 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \forall x. \text{player}(x) \rightarrow \neg \text{ball}(x), )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Luciano Serafini, Ivan Donadello, Artur d’Avila Garces (Fondazione Bruno Kessler, Italy University of Trento, Italy City University London, UK)
Semantic Image interpretation

Semantic Image Interpretation (SII)

object detection: Fast RCNN (state of the art object detector)

For each pair of bounding boxes we compute additional binary feature that measure the mutual overlap between the two bounding boxes.

\[ x_{BL}(b_1) = 14 \]
\[ y_{BL}(b_1) = 17 \]
\[ \text{width}(b_1) = 40 \]
\[ \text{height}(b_1) = 100 \]
\[ \text{rcnn ball}(b_1) = 0.1 \]
\[ \text{rcnn player}(b_1) = 0.7 \]
\[ \text{rcnn logo}(b_1) = 0.02 \]

...
Semantic Image interpretation

**Semantic Image Interpretation (SII)**

- **Object detection**: Fast RCNN (state of the art object detector)
Semantic Image interpretation

Semantic Image Interpretation (SII)

- object detection: Fast RCNN (state of the art object detector)
- Fast-RCNN returns candidate bounding boxes, associated with weights for each object class;

\[ x_{BL}(b_1) = 12 \]
\[ y_{BL}(b_1) = 27 \]
\[ width(b_1) = 30 \]
\[ height(b_1) = 30 \]
\[ rcnn_{ball}(b_1) = 0.8 \]
\[ rcnn_{player}(b_1) = 0.3 \]
\[ rcnn_{logo}(b_1) = 0.02 \]

...
Semantic Image interpretation

Semantic Image Interpretation (SII)

- object detection: Fast RCNN (state of the art object detector)
- Fast-RCNN returns candidate bounding boxes, associated with weights for each object class;

\[
x_{BL}(b_1) = 12
y_{BL}(b_1) = 27
width(b_1) = 30
height(b_1) = 30
rcnn_{ball}(b_1) = .8
rcnn_{player}(b_1) = .3
rcnn_{logo}(b_1) = .02
\]

\[
x_{BL}(b_1) = 14
y_{BL}(b_1) = 17
width(b_1) = 40
height(b_1) = 100
rcnn_{ball}(b_1) = .1
rcnn_{player}(b_1) = .7
rcnn_{logo}(b_1) = .02
\]

Luciano Serafini, Ivan Donadello, Artur d’Avila Garces (Fondazione Bruno Kessler, Italy University of Trento, Italy City University London, UK)

Learning and Reasoning in Logic Tensor Networks

May 7, 2017 18 / 23
Semantic Image interpretation

Semantic Image Interpretation (SII)

- object detection: Fast RCNN (state of the art object detector)
- Fast-RCNN returns candidate bounding boxes, associated with weights for each object class;

```
xBL(b1) = 12
yBL(b1) = 27
width(b1) = 30
height(b1) = 30
rcnn_ball(b1) = .8
rcnn_player(b1) = .7
rcnn_logo(b1) = .02
```

```
xBL(b1) = 14
yBL(b1) = 17
width(b1) = 40
height(b1) = 100
rcnn_ball(b1) = .1
rcnn_player(b1) = .7
rcnn_logo(b1) = .02
```
Semantic Image interpretation

**Semantic Image Interpretation (SII)**

- **Object detection:** Fast RCNN (state of the art object detector)
- Fast-RCNN returns candidate bounding boxes, associated with weights for each object class;
- For each pair of bounding boxe we compute additional binary feature that measure the mutual overlap between the two bounding boxes.

Example:

- $x_{BL}(b_1) = 12$
- $y_{BL}(b_1) = 27$
- $\text{width}(b_1) = 30$
- $\text{height}(b_1) = 30$
- $\text{rcnnball}(b_1) = .8$
- $\text{rcnnpplayer}(b_1) = .3$
- $\text{rcnnlogo}(b_1) = .02$

- $x_{BL}(b_1) = 34$
- $y_{BL}(b_1) = 17$
- $\text{width}(b_1) = 40$
- $\text{height}(b_1) = 100$
- $\text{rcnnball}(b_1) = .1$
- $\text{rcnnpplayer}(b_1) = .74$
- $\text{rcnnlogo}(b_1) = .02$

- $\text{contRatio}(b_2, b_3) = 0.3$
- $\text{contRatio}(b_3, b_2) = 0.2$
PascalPart contains **10103 pictures** annotated with a set of bounding boxes labelled with object types (60 classes among animals, vehicles, and indoor objects).

We train an LTN with the approx 2/3 pictures and test on 1/3. by including the following **background knowledge**

- positive/negative examples for object classes (from training set)
  - \(\text{wheel}(\text{bb1}), \ 	ext{car}(\text{bb2}), \ 
eg\text{horse}(\text{bb2}), \neg\text{person}(\text{bb4})\)
- positive/negative examples for relations (we focus on parthood relation).
  - \(\text{partOf}(\text{bb1}, \text{bb2}), \ 
eg\text{partOf}(\text{bb2}, \text{bb3}), \ldots\)
- general axioms about parthood relation:
  - \(\forall x. \text{car}(x) \land \text{partof}(y, y) \rightarrow \text{wheel}(y) \lor \text{mirror}(y) \lor \text{door}(y) \lor \ldots\)
**LTN for SII results**

- **$LTN_{\text{prior}}$** is an LTN trained with positive and negative examples + general axioms about partOf relation

- **$LTN_{\text{expl}}$** is an LTN trained only with positive and negative examples of types and partOf

- **FRCNN** is the baseline proposal classification for types given by Fast-RCNN

- **RBPOF** is the baseline for partOf based on the naive criteria

  \[ \text{area containment} \geq \text{threshold} \]
Robustness w.r.t. noisy data

- Logical axioms improve the robustness of the system in presence of noise in the labels of training data.
- We artificially add an increasing amount of noise to the PascalPart-dataset training data, and we measure the degradation of the performance.

**AUC evolution for part-of**

**AUC evolution for types**
Conclusions

- we introduce **Logic Tensor Networks**, a general framework for SRL that integrates fuzzy logical reasoning and machine learning based on neural networks;

- We apply LTN to the challenging problem of *semantic image interpretation*;

- We experimentally show that the usage of logic based background knowledge improves the performance of automatic classification based only on numeric features.
Thanks for your attention