On the Design of PSyKI: a Platform for Symbolic Knowledge Injection into Sub-Symbolic Predictors

Magnini Matteo[*] Ciatto Giovanni[*]
Omicini Andrea[*]

[*]Dipartimento di Informatica – Scienza e Ingegneria (DISI)
Alma Mater Studiorum – Università di Bologna
{*matteo.magnini* , giovanni.ciatto, andrea.omicini}@unibo.it

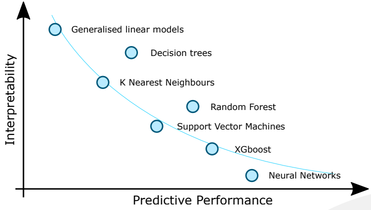
International Workshop on
EXplainable and TRasparent AI and Multi-Agent Systems
(EXTRAAMAS)
20th May, 2022, Auckland, NZ (fully online)

Next in Line...

1. Motivation & Context
2. Design overview
3. Use case and example
4. Conclusions & future works

Context

- Most of modern ML predictors are black-boxes [Lipton, 2018]
  - interpretability vs performance trade-off



⇒ Dual (not exclusive) approaches to tackle the problem:

- ! symbolic knowledge extraction (SKE)
- ! symbolic knowledge injection (SKI)

- Binding of inductive and deductive reasoning

Motivation

- More and more ML applications in critical domains
  - e.g. medicine, finance, law
  - ⇒ Prevent predictors to become black-boxes
    - symbolic knowledge is used to guide predictors during learning
      - ⇒ predictors do not violate or violate less the prior knowledge
- Overcome the performances of ML state-of-the-art solutions
  - e.g. accuracy, learning time, need for less training data
- Fill the lack of public and usable implementation of SKI algorithms

Contribution

Contributions of the work

- Elicit common steps of SKI workflow algorithms from literature
- Implementation of a Python library for usage and development of SKI algorithms
  - ⇒ that supports the key phases of the SKI workflow
  - ⇒ with interoperability for PSyKE [Sabbatini et al., 2021] and 2ppy

Next in Line. . .

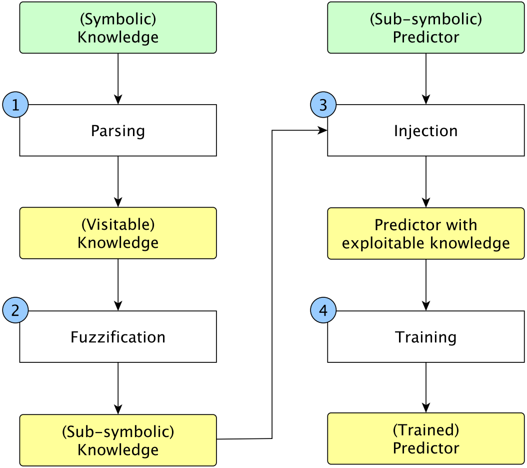
1. Motivation & Context
2. Design overview
3. Use case and example
4. Conclusions & future works

Modelling

Common SKI workflow steps from literature
[Besold et al., 2017, Xie et al., 2019, Calegari et al., 2020]

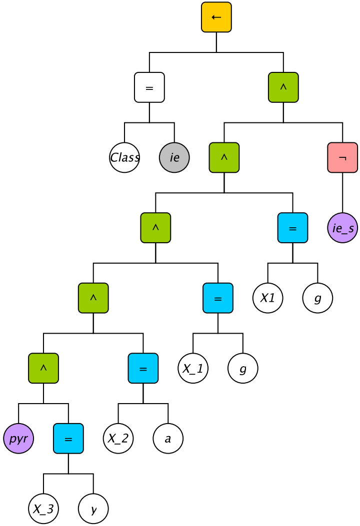
1. Knowledge is commonly provided using a logic formalism
   e.g. first order logic, knowledge graph, propositional logic, etc.
2. Symbolic knowledge is somehow encoded into a sub-symbolic form
3. Ad-hoc methods to inject the sub-symbolic knowledge into a predictor
   e.g. modifying the loss function, structuring the architecture, etc.
4. Training (virtually always required)

General SKI workflow



1 – Parsing

$$
\begin{aligned}
&\mathit{class}(X_{-30}, \ldots, X_{30}, \mathit{ie}) \leftarrow \\
&\qquad \mathit{pyramidine\text{-}rich}(\ldots) \land \\
&\qquad X_{-3} = y \land \\
&\qquad X_{-2} = a \land \\
&\qquad X_{-1} = g \land \\
&\qquad X_{1} = g \land \\
&\qquad \neg(\mathit{ie\text{-}stop}(\ldots))
\end{aligned}
$$



2 – Fuzzification

Formula	Continuous interpretation
$[\![\neg\phi]\!]$	$1 - [\![\phi]\!]$
$[\![\phi \wedge \psi]\!]$	$min\{[\![\phi]\!], [\![\psi]\!]\}$
$[\![\phi \vee \psi]\!]$	$max\{[\![\phi]\!], [\![\psi]\!]\}$
$[\![\phi = \psi]\!]$	$[\![\neg(\phi \neq \psi)]\!]$
$[\![\phi \neq \psi]\!]$	$\lvert[\![\phi]\!] - [\![\psi]\!]\rvert$
$[\![\phi > \psi]\!]$	$max\{0, [\![\phi]\!] - [\![\psi]\!]\}$
$[\![\phi \geq \psi]\!]$	$[\![(\phi > \psi) \vee (\phi = \psi)]\!]$
$[\![\phi < \psi]\!]$	$max\{0, [\![\psi]\!] - [\![\phi]\!]\}$
$[\![\phi \leq \psi]\!]$	$[\![(\phi < \psi) \vee (\phi = \psi)]\!]$
$[\![\phi \Rightarrow \psi]\!]$	$min\{1, 1 - [\![\psi]\!] + [\![\phi]\!]\}$
$[\![\phi \Leftarrow \psi]\!]$	$min\{1, 1 - [\![\phi]\!] + [\![\psi]\!]\}$
$[\![\phi \Leftrightarrow \psi]\!]$	$min\{1, 1 - \lvert[\![\phi]\!] - [\![\psi]\!]\rvert\}$
$[\![\mathrm{expr}(\bar{X})]\!]$	$\mathrm{expr}([\![\bar{X}]\!])$
$[\![\mathtt{true}]\!]$	1
$[\![\mathtt{false}]\!]$	0
$[\![X]\!]$	x
$[\![\mathrm{k}]\!]$	k
$[\![p(\bar{X})]\!]^{**}$	$[\![\psi_1 \vee \ldots \vee \psi_k]\!]$
$[\![class(\bar{X}, \mathrm{y}_i) \leftarrow \psi]\!]$	$[\![\psi]\!]^*$

* encodes the value for the i^{th} output

** assuming p is defined by k clauses of the form:
$p(\bar{X}) \leftarrow \psi_1, \ \ldots, \ p(\bar{X}) \leftarrow \psi_k$

$$class(X_{-30}, \ldots, X_{30}, ie) \leftarrow$$
$$X_{-3} = y \ \wedge$$
$$X_{-2} = a \ \wedge$$
$$X_{-1} = g \ \wedge$$
$$X_1 = g$$

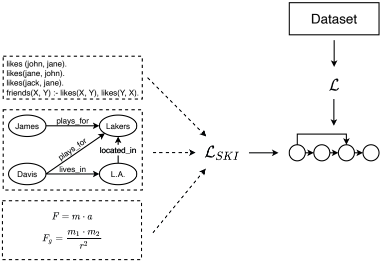
$$\downarrow$$

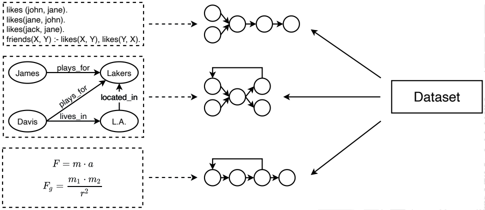
$$min\{min\{min\{1 - |X_{-3} - y|, 1 - |X_{-2} - a|\}, 1 - |X_{-1} - g|\}, 1 - |X_1 - g|\}$$

3 – Injection

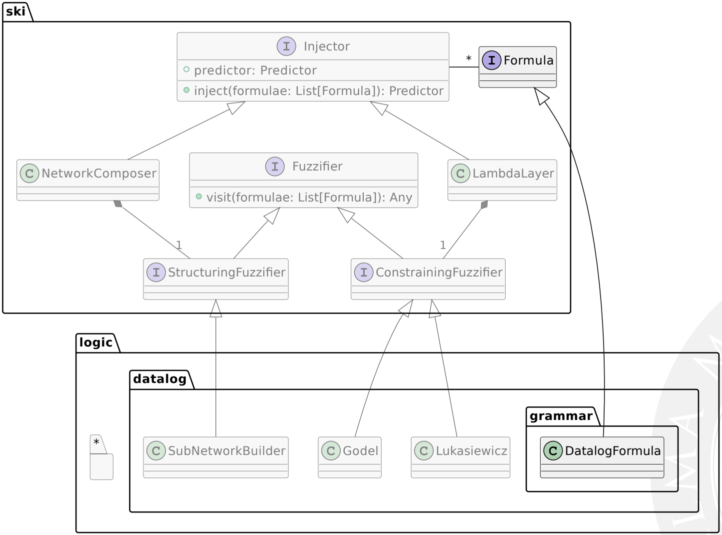
Usually injection is performed

- by adding a cost factor to the loss function [Tresp et al., 1992]
- by structuring the predictor [Ballard, 1986, Towell et al., 1990]

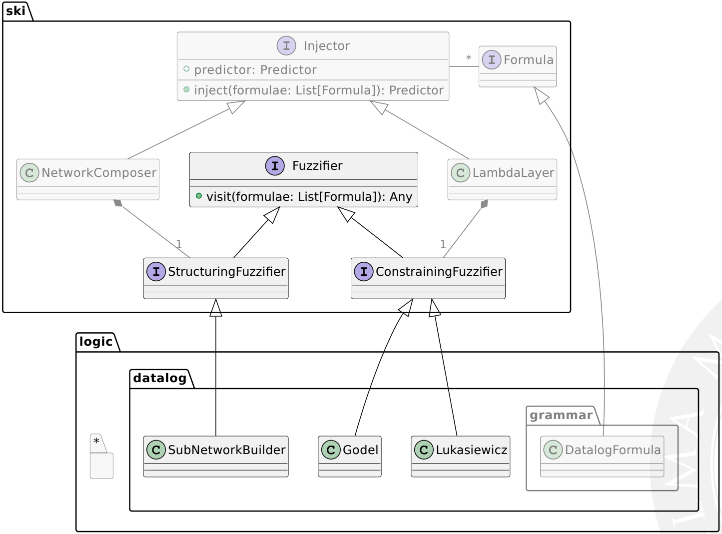




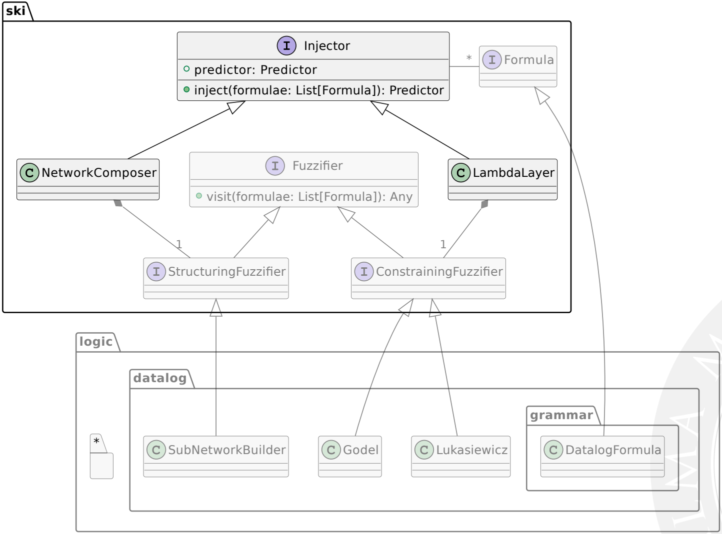
PSyKI high level class diagram



PSyKI high level class diagram



PSyKI high level class diagram



Next in Line...

1. Motivation & Context
2. Design overview
3. Use case and example
4. Conclusions & future works

Use case and example

Splice junction dataset and knowledge from UCI [Dua and Graff, 2017]

```
EI-stop ::- @-3 'TAA'.
EI-stop ::- @-3 'TAG'.
EI-stop ::- @-3 'TGA'.
EI-stop ::- @-4 'TAA'.
EI-stop ::- @-4 'TAG'.
EI-stop ::- @-4 'TGA'.
EI-stop ::- @-5 'TAA'.
EI-stop ::- @-5 'TAG'.
EI-stop ::- @-5 'TGA'.

IE-stop ::- @1 'TAA'.
IE-stop ::- @1 'TAG'.
IE-stop ::- @1 'TGA'.
IE-stop ::- @2 'TAA'.
IE-stop ::- @2 'TAG'.
IE-stop ::- @2 'TGA'.
IE-stop ::- @3 'TAA'.
IE-stop ::- @3 'TAG'.
IE-stop ::- @3 'TGA'.

pyramidine-rich :- 6 of (@-15 'YYYYYYYYYY').

EI :- @-3 'MAGGTRAGT', not(EI-stop).

IE :- pyramidine-rich, @-3 'YAGG', not(IE-stop).
```

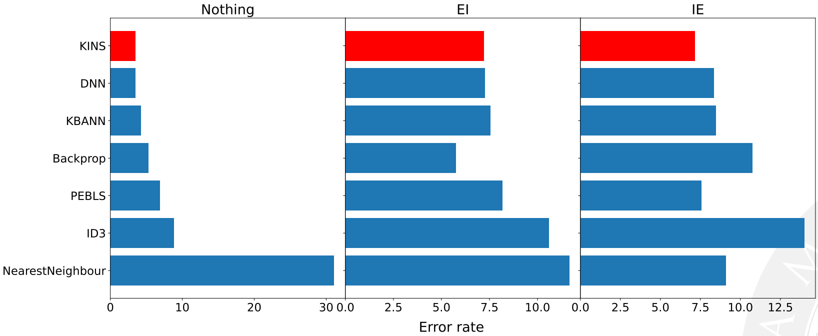
```
Class, Id, DNA-sequence

EI,ATRINS-DONOR-521,CCAGCTGCAT...AGCCAGTCTG
EI,ATRINS-DONOR-905,AGACCCGCCG...GTGCCCCCGC
EI,BABAPOE-DONOR-30,GAGGTGAAGG...CACGGGGATG
...
IE,ATRINS-ACCEPTOR-701,TTCAGCGGCC...GCCCTGTGGA
IE,ATRINS-ACCEPTOR-1678,GGACCTGCTC...GGGGGCTCTA
IE,BABAPOE-ACCEPTOR-801,GCGGTTGATT...AAGATGAAGG
...
N,AGMKPNRSB-NEG-1,CAAAAGAACA...CAAGGCTACA
N,AGMORS12A-NEG-181,AGGGAGGTGT...GGGCATGGGG
N,AGMORS9A-NEG-481,TGGTCAATTC...TCTTGCTCTG
...
```

Code snippet

General script to perform injection

```python
from psyki.logic.datalog.grammar.adapters.antlr4 import get_formula_from_string
from psyki.ski.injectors import NetworkComposer

# ...

# Symbolic knowledge
with open(filename) as file:
    rows = file.readlines()
# 1 - Parsing
knowledge = [get_formula_from_string(row) for row in rows]

predictor = build_NN()
# 2 and 3 - Fuzzification and injection
injector = InjectorX(predictor)
predictor_with_knowledge = injector.inject(knowledge)

# 4 - Training
predictor_with_knowledge.fit(train_x, train_y)
```

Results


Nothing
EI
IE
KINS
DNN
KBANN
Backprop
PEBLS
ID3
NearestNeighbour
0
10
20
30
0.0
2.5
5.0
7.5
10.0
0.0
2.5
5.0
7.5
10.0
12.5
Error rate

Next in Line...

1 Motivation & Context

2 Design overview

3 Use case and example

4 Conclusions & future works

Conclusions & future works

Summing up

Relevant contributions of this work:

- detecting main steps of SKI workflow
- implementation of a Python library that
  - provide already existing SKI algorithms
  - support the development of new algorithms

Future works

Some future research directions

- training predictors combining SKE and SKI
- implementation of the most successful SKI algorithms

On the Design of PSyKI: a Platform for Symbolic Knowledge Injection into Sub-Symbolic Predictors

Magnini Matteo[*] Ciatto Giovanni[*]
Omicini Andrea[*]

[*]Dipartimento di Informatica – Scienza e Ingegneria (DISI)
Alma Mater Studiorum – Università di Bologna
{*matteo.magnini* , giovanni.ciatto, andrea.omicini}@unibo.it

International Workshop on
EXplainable and TRasparent AI and Multi-Agent Systems
(EXTRAAMAS)
20^{th} May, 2022, Auckland, NZ (fully online)

References I

[Ballard, 1986] Ballard, D. H. (1986).
Parallel logical inference and energy minimization.
In Kehler, T., editor, *Proceedings of the 5th National Conference on Artificial Intelligence. Philadelphia, PA, USA, August 11-15, 1986. Volume 1: Science*, pages 203–209. Morgan Kaufmann
http://www.aaai.org/Library/AAAI/1986/aaai86-033.php.

[Besold et al., 2017] Besold, T. R., d'Avila Garcez, A. S., Bader, S., Bowman, H., Domingos, P. M., Hitzler, P., Kühnberger, K., Lamb, L. C., Lowd, D., Lima, P. M. V., de Penning, L., Pinkas, G., Poon, H., and Zaverucha, G. (2017).
Neural-symbolic learning and reasoning: A survey and interpretation.
CoRR, abs/1711.03902
http://arxiv.org/abs/1711.03902.

References II

[Calegari et al., 2020] Calegari, R., Ciatto, G., and Omicini, A. (2020).
On the integration of symbolic and sub-symbolic techniques for XAI: A survey.
Intelligenza Artificiale, 14(1):7–32
DOI:10.3233/IA-190036.

[Dua and Graff, 2017] Dua, D. and Graff, C. (2017).
UCI machine learning repository
http://archive.ics.uci.edu/ml.

[Lipton, 2018] Lipton, Z. C. (2018).
The mythos of model interpretability.
Commun. ACM, 61(10):36–43
DOI:10.1145/3233231.

References III

[Sabbatini et al., 2021] Sabbatini, F., Ciatto, G., Calegari, R., and Omicini, A. (2021).
On the design of PSyKE: A platform for symbolic knowledge extraction.
In Calegari, R., Ciatto, G., Denti, E., Omicini, A., and Sartor, G., editors, *WOA 2021 – 22nd Workshop "From Objects to Agents"*, volume 2963 of *CEUR Workshop Proceedings*, pages 29–48. Sun SITE Central Europe, RWTH Aachen University.
22nd Workshop "From Objects to Agents" (WOA 2021), Bologna, Italy, 1–3 September 2021. Proceedings

http://ceur-ws.org/Vol-2963/paper14.pdf.

[Towell et al., 1990] Towell, G. G., Shavlik, J. W., and Noordewier, M. O. (1990).
Refinement of approximate domain theories by knowledge-based neural networks.
In Shrobe, H. E., Dietterich, T. G., and Swartout, W. R., editors, *Proceedings of the 8th National Conference on Artificial Intelligence. Boston, Massachusetts, USA, July 29 - August 3, 1990, 2 Volumes*, pages 861–866. AAAI Press / The MIT Press

http://www.aaai.org/Library/AAAI/1990/aaai90-129.php.

References IV

[Tresp et al., 1992] Tresp, V., Hollatz, J., and Ahmad, S. (1992).
Network structuring and training using rule-based knowledge.
In Hanson, S. J., Cowan, J. D., and Giles, C. L., editors, *Advances in Neural Information Processing Systems 5, [NIPS Conference, Denver, Colorado, USA, November 30 - December 3, 1992]*, pages 871–878. Morgan Kaufmann

http://papers.nips.cc/paper/638-network-structuring-and-training-using-rule-based-knowledge.

[Xie et al., 2019] Xie, Y., Xu, Z., Meel, K. S., Kankanhalli, M. S., and Soh, H. (2019).
Embedding symbolic knowledge into deep networks.
In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4235–4245

https://proceedings.neurips.cc/paper/2019/hash/7b66b4fd401a271a1c7224027ce111bc-Abstract.html.