Can you trust your Artificial Intelligence?

Daniele Zonca
Principal Software Engineer
Daniele Zonca

Principal Software Engineer @ Red Hat - Business Automation (2018-now)
- Drools: rule engine
- jBPM: process engine
- OptaPlanner: constraint solver
- Kogito: cloud native business automation

Team Leader @ Unicredit - Big Data Dept (2015-2018)
- Spark analytical engines
- Corporate CRM
Agenda
AGENDA

- Introduction to AI
- Symbolic
  - Tracing
  - Embed knowledge
- Sub-symbolic
  - Manage noisy data
  - Data Driven
- Right to explanation
- TrustyAI
  - Interpretability
  - Compliance
Introduction to AI
In computer science, artificial intelligence (AI) is intelligence demonstrated by machines, in contrast to the natural intelligence displayed by humans (Wikipedia).

Two main approaches:
- Symbolic: logic/rule based
- Sub-symbolic: statistical learning
Symbolic AI
- Tracing
- Embed knowledge
Prolog (1972)

Predicates/Rules:

\[
\begin{align*}
\text{siblings} & : = \text{parent-child}(Z, X), \text{parent-child}(Z, Y) \\
\text{parent-child}(X, Y) & : = \text{father-child}(X, Y) \\
\text{parent-child}(X, Y) & : = \text{mother-child}(X, Y)
\end{align*}
\]

Facts:

\[
\begin{align*}
\text{mother-child}(\text{trude}, \text{sally}) \\
\text{father-child}(\text{tom}, \text{sally}) \\
\text{father-child}(\text{tom}, \text{erica}) \\
\text{father-child}(\text{mike}, \text{tom})
\end{align*}
\]

Query

?- \text{siblings}(\text{sally}, \text{erica}).

Yes
Drools

Rules:

```plaintext
rule "validate holiday"
when
    $h1 : Month( name == "july" )
then
    drools.insert(new HolidayNotification($h1));
end
```

Facts:

```plaintext
drools.insert(new Month("july"))
drools.insert(new Month("may")
```

Query

```plaintext
query "checkHolidayNotification" (String monthName)
    holiday := HolidayNotification(month.name == monthName )
end
```
Symbolic AI
- Tracing
- Embed knowledge
Predicates/Rules:

\[
\text{sibling}(X, Y) \ := \ \text{parent\_child}(Z, X), \ \text{parent\_child}(Z, Y).
\]

\[
\text{parent\_child}(X, Y) \ := \ \text{father\_child}(X, Y).
\]

\[
\text{parent\_child}(X, Y) \ := \ \text{mother\_child}(X, Y).
\]

Query

?\text{- sibling}(sally, erica).

Yes

Facts:

\[
\text{mother\_child}(trude, sally).
\]

\[
\text{father\_child}(tom, sally).
\]

\[
\text{father\_child}(tom, erica).
\]

\[
\text{father\_child}(mike, tom).
\]
Predicates/Rules:
sibling(X, Y) :- parent_child(Z, X), parent_child(Z, Y).
parent_child(X, Y) :- father_child(X, Y).
parent_child(X, Y) :- mother_child(X, Y).

Facts:
mother_child(trude, sally).
father_child(tom, sally).
father_child(tom, erica).
father_child(mike, tom).

Query
?- trace, sibling(sally, erica).
Yes
Predicates/Rules:

\[
\text{siblings}(X, Y) \quad \text{:-} \quad \text{parent} \_ \text{child}(Z, X), \text{parent} \_ \text{child}(Z, Y).
\]

\[
\text{parent} \_ \text{child}(X, Y) \quad \text{:-} \quad \text{father} \_ \text{child}(X, Y).
\]

\[
\text{parent} \_ \text{child}(X, Y) \quad \text{:-} \quad \text{mother} \_ \text{child}(X, Y).
\]

Facts:

\[
\text{mother} \_ \text{child}(\text{trude}, \text{sally}).
\]

\[
\text{father} \_ \text{child}(\text{tom}, \text{sally}).
\]

\[
\text{father} \_ \text{child}(\text{tom}, \text{eric}a).
\]

\[
\text{father} \_ \text{child}(\text{mike}, \text{tom}).
\]

Query

\[
?\text{- trace}, \text{siblings(sally, erica)}.
\]

Yes

Call: sibling(sally, erica)

Call: parent_child(_4150, sally)

Call: father_child(_4150, sally)

Exit: father_child(tom, sally)

Exit: parent_child(tom, sally)

Call: parent_child(tom, erica)

Call: father_child(tom, erica)

Exit: father_child(tom, erica)

Exit: parent_child(tom, erica)

Exit: sibling(sally, erica)
Symbolic AI
- Tracing
- Embed knowledge
Prolog (1972)

Predicates/Rules:
\[
\text{siblings}(X, Y) \iff \text{parent\_child}(Z, X), \text{parent\_child}(Z, Y).
\]
\[
\text{parent\_child}(X, Y) \iff \text{father\_child}(X, Y).
\]
\[
\text{parent\_child}(X, Y) \iff \text{mother\_child}(X, Y).
\]

Facts:
\[
\text{mother\_child}(\text{trude}, \text{sally}).
\]
\[
\text{father\_child}(\text{tom}, \text{sally}).
\]
\[
\text{father\_child}(\text{tom}, \text{erica}).
\]
\[
\text{father\_child}(\text{mike}, \text{tom}).
\]
Prolog (1972)

Predicates/Rules:

\[
\text{sibling}(X, Y) \quad :- \quad \text{parent\_child}(Z, X), \text{parent\_child}(Z, Y).
\]

\[
\text{parent\_child}(X, Y) \quad :- \quad \text{father\_child}(X, Y).
\]

\[
\text{parent\_child}(X, Y) \quad :- \quad \text{mother\_child}(X, Y).
\]

\[
\text{grandfather}(X, Y) \quad :- \quad \text{parent\_child}(Z, Y), \text{father\_child}(X, Z).
\]

Facts:

\[
\text{mother\_child}(\text{trude}, \text{sally}).
\]

\[
\text{father\_child}(\text{tom}, \text{sally}).
\]

\[
\text{father\_child}(\text{tom}, \text{erica}).
\]

\[
\text{father\_child}(\text{mike}, \text{tom}).
\]

\[
\text{mother\_child}(\text{erica}, \text{max}).
\]
Is this enough to cover all use cases?
- Image recognition
- Speech recognition
- Anomaly detection
Sub-symbolic AI
- Data driven
- Manage noisy data
Sub-symbolic AI
- Data driven
- Manage noisy data
SUB-SYMBOLIC AI - DATA DRIVEN
Sub-symbolic AI
- Data driven
- Manage noisy data
Random Forest Simplified

- Random Forest

- Instance

- Tree-1
  - Class-A

- Tree-2
  - Class-B

- Tree-n
  - Class-B

- Majority-Voting

- Final-Class
Google Photos Tags Two African-Americans As Gorillas Through Facial Recognition Software

Maggie Zhang  Forbes Staff
Tech
I write about technology, innovation, and startups.

This article is more than 2 years old.

When It Comes to Gorillas, Google Photos Remains Blind

Google promised a fix after its photo-categorization software labeled black people as gorillas in 2015. More than two years later, it hasn’t found one.

Google ‘fixed’ its racist algorithm by removing gorillas from its image-labeling tech

Nearly three years after the company was called out, it hasn’t gone beyond a quick workaround

By James Vincent  Jan 12, 2018, 10:35am EST
Amazon scraps secret AI showed bias against won

Jeffrey Dustin

SAN FRANCISCO (Reuters) - Amazon.com Inc’s specialists uncovered a big problem: their

The group created 500 computer models focused on specific job functions and locations. They taught each to recognize some 50,000 terms that showed up on past candidates’ resumes. The algorithms learned to assign little significance to skills that were common across IT applicants, such as the ability to write various computer codes, the people said.

Instead, the technology favored candidates who described themselves using verbs more commonly found on male engineers’ resumes, such as “executed” and “captured,” one person said.

Amazon trained a sexism-fighting, resume-screening AI with sexist hiring data, so the bot became sexist

Amazon reportedly scraps internal AI recruiting tool that was biased against women

The secret program penalized applications that contained the word “women’s”
Right to explanation
Articles 13-15 of the regulation
“meaningful information about the logic involved”
“the significance and the envisaged consequences”

Article 22 of the regulation
that data subjects have the right not to be subject to such decisions when they’d have the type of impact described above

Recital 71 (part of a non-binding commentary included in the regulation)
States that data subjects are entitled to an explanation of automated decisions after they are made, in addition to being able to challenge those decisions.
Can you trust your artificial intelligence?

TrustyAI
- Interpretability
- Compliance
The Personas
- Case Worker (End User)
- Software Engineer (Developer)
- Data Scientist (Applied Theorist / Developer)
- Compliance Worker (Ethicists / End User)
What Are We Trying To Do?

Today

- Training Data → Learning Process → Learned Function → This is a cat (p = .93) → Output → User with a Task

- Why did you do that?
- Why not something else?
- When do you succeed?
- When do you fail?
- When can I trust you?
- How do I correct an error?

Tomorrow

- Training Data → New Learning Process → Explainable Model → Explanation Interface → This is a cat: • It has fur, whiskers, and claws. • It has this feature: → User with a Task

- I understand why
- I understand why not
- I know when you’ll succeed
- I know when you’ll fail
- I know when to trust you
- I know why you erred
TrustyAI
- Interpretability
- Compliance
Input:
4 years old passenger from 1st class. Paid 72 for the ticket
Input:
4 years old passenger from 1st class. Paid 72 for the ticket

Random Forest prediction: 0.422
Input:
4 years old passenger from 1st class. Paid 72 for the ticket

What is the contribution of each variable to the final odds?
(model: Random Forest)

Random Forest prediction: 0.422
Input:
4 years old passenger from 1st class. Paid 72 for the ticket

What is the contribution of each variable to the final odds?
(model: Random Forest)

Random Forest prediction: 0.422

---

iBreakDown: Uncertainty of Model Explanations for Non-additive Predictive Models
286 Egyptian cat 0.000892741
242 EntleBucher 0.0163564
239 Greater Swiss Mountain dog 0.0171362
241 Appenzeller 0.0393639

240 Bernese mountain dog 0.829222

LIME

SHAP
CAN YOU TRUST YOUR ARTIFICIAL INTELLIGENCE?

TrustyAI
- Interpretability
- Compliance
**Dataflow tracing**: reporting that spans and links across design, accept, build, deploy and runtime with shallow provenance.
Questions?
Thank you

Red Hat is the world’s leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat  facebook.com/redhatinc
youtube.com/user/RedHatVideos  twitter.com/RedHat