eXplainable Artificial Intelligence (XAI)
A Gentle Introduction

Matteo Magnini   Giovanni Ciatto   Andrea Omicini

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

Advanced School in Artificial Intelligence – 17-28 July 2023

Next in Line...

Drivers & Limitations I

Socio-political requirements

- both individuals and human organisations rely more and more upon artificial systems
  - which are delegated *increasingly-complex* functions, tasks, and goals that human processes depend upon
- artificial systems are nowadays required to
  - understand the *context*, the *users*, and the *goals* of the system itself, and behave accordingly
  - operate autonomously in *dynamic environments*
  - work with physically-sparse components, each one *placed* in its own physical location

Drivers & Limitations II

Drivers

- drawing from the aforementioned requirements, we can see that the main drivers for the engineering of artificial systems nowadays are
  - intelligence
  - *autonomy*
  - *physical distribution*
- today we obviously focus on intelligence as our main line
  - possibile keeping in mind the other two for any future reference

Drivers & Limitations III

Limitations

- Dually, artificial systems are also *ideally* required to
  - be trustable by humans—so, transparent, understandable, accountable, . . . for human users
  - respect human autonomy at their core, possibly mitigating their own autonomous behaviour, and supporting human users in their choices and deliberations
  - be non-intrusive, both physically and cognitively, while respecting and protecting privacy and safety of human users
- Yet, we are far far away from there

Where is AI from? I

- understanding *how intelligence works* is a persistent issue for humans
  - Aristotle's *logics* is the most outstanding example of that [De Rijk, 2002]
- "understanding", for humans, typically means to be able to *model* and *reproduce*
- building machines that can *reproduce intelligence*
  - either as by reproducing some known intelligent process
  - or by reproducing some observed intelligent behaviour

  is a way to measure how much we understand the way in which intelligence works

Where is AI from? II

The birth of AI

- the dualism between AI as *intelligent behaviour* and AI as *intelligent process* was already there in AI since the very beginning
- Dartmouth College, New Hampshire, USA – Summer School, 1956
  - John McCarthy invites all scholars interested in *computing towards intelligence*
- among those
  - Marvin Minsky, co-founder of AI Lab at MIT
  - Alan Newell, Herb Simon, authors of Logic Theorist (an automatic theorem prover)—likely the *first AI program* [Newell and Simon, 1956]
  - John McCarthy, inventor of LISP, the *first programming language* for AI [McCarthy, 1981]
- the term "Artificial Intelligence" was actually coined there, to describe the overall new field of research

General AI I

General purpose AI

- building general-purpose intelligence machines is the goal of General AI
- we do have a *poor understanding* of human intelligence, and of intelligence in general
- early AI focussed then on *intelligent components*

Components of intelligence

- perception
- problem solving & planning
- reasoning
- machine learning
- natural language understanding

General AI II

Perception

- understanding the environment
- through sensors of any sort
- *interpreting* the overall situation

! one of the most difficult task of AI

Problem solving & planning

- devising a course of actions towards a *goal*
- based on a *repertoire of actions*
- e.g., playing games

General AI III

Machine learning

- learning from data
- building models (e.g., classification)
- making *predictions*
- e.g., face recognition through training

Reasoning

- representing knowledge
- inferring new knowledge from available one
- in a *consistent* and *robust* way

General AI IV

Natural language understanding

- ability to understand human languages
- either spoken or written
- possibly engage in conversations with humans

! currently the main focus of the *natural language processing* (NLP) field

AI: The Contemporary Era I

1 – Grand DARPA Challenges

- where AI and autonomous systems shared their first success
- race for autonomous vehicles in the desert of Nevada (2005)
  - won by STANLEY,[Thrun et al., 2006] a converted Volkswagen Touareg, equipped with seven onboard computers, interpreting sensor data from GPS, laser rangefinders, radar, and video feed
- the sudden global attention towards *autonomous cars* came from this very stream

AI: The Contemporary Era II

2 – Alpha Go: Triumph of ML [Silver et al., 2016]

- in 2014 DeepMind demonstrated a system learning how to play arcade games just looking at the video and accessing the scores, using the same controls as humans
  - acquired by Google, they built Alpha Go, which beat Go champion Lee Sedol 4 to 1 in 2016
- exploiting *deep neural networks* along with *self-training*
- Go search space is so huge that brute force just does not work: so, it was considered impossible for a machine to beat a human at Go
  - so, this also made everybody aware that there were no known limits to the ability that machine intelligence could reach

AI: The Contemporary Era III

ML: Three factor for success

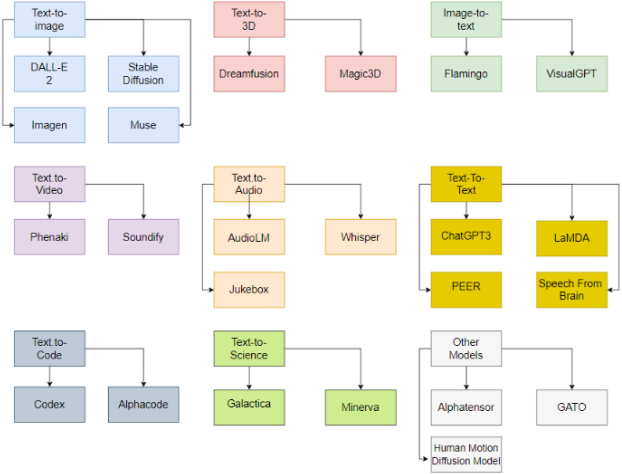
- scientific breakthroughs—deep learning dealing with complex problems
- training requires lots of data—nowadays data are hugely available
- training requires computational power—nowadays computational power is more and more available

AI: The Contemporary Era IV

3 – ChatGPT and Beyond: Generative AI

- "classic" AI techniques mostly deal with analysing or acting on existing data
  - e.g., expert systems, built upon *knowledge bases* and an *inference engine* generating content via an *if-else rule database*
- generative AI[Gozalo-Brizuela and Garrido-Merchan, 2023] includes instead techniques that can *generate novel content*, using mechanisms like *probabilistic machine learning*[Murphy, 2022]

AI: The Contemporary Era V



A taxonomy of current Generative AI available technologies [Gozalo-Brizuela and Garrido-Merchan, 2023]

Intelligent Socio-Technical Systems

- in the realm of intelligent systems, nowadays, humans are legitimate components in the same way as software and physical agents
- where both *human* and *software agents* accounts for activity, knowledge, intelligence, goals, learning, . . .
- as legitimate components of intelligent socio-technical systems
- so that now the fundamental question becomes
  - ? how are we going to shape the interaction between heterogeneous intelligent components within *intelligent socio-technical systems*?

?? e.g., is NLP the answer?

People Need to Understand Systems

- human users rely more and more on intelligent systems for their everyday activities, as well as for critical aspects such as health and money
- humans and intelligent agents work together in intelligent socio-technical systems to produce overall intelligent behaviour
  - e.g. *decision support systems* exploit intelligent systems in order to promote rational human decisions

→ information and actions by intelligent agents need to be *understandable* by humans to be accepted and *trusted*

→ humans need *explanations*

- which is where explainable artificial intelligence (XAI) comes from[Gunning, 2016b]

Why Don't Humans Understand Intelligent Systems?

- the technical XAI problem in short
  - *symbolic* approaches are *transparent* yet slow—e.g., computational logic
  - *sub-symbolic* approaches are *fast* yet opaque—e.g., deep learning
- so, symbolic / sub-symbolic *integration* is the most promising way out
  - and, everyone is already doing that[Calegari et al., 2020]
- yet: integration how?
  - based on what integration model?
  - which *conceptual foundation* for integrating symbolic / sub-symbolic techniques within a coherent intelligent system model / architecture?
  - and mostly, how do we keep the benefits of both without the drawbacks?

Explanation Everywhere

- the notion of *explanation* is the core of many research efforts
  - along with accessory notions such as *interpretation* and *understandability*
- and undergone a constant flow of diverse and (sometimes) even extravagant definitions
  - e.g., even GDPR[Voigt and von dem Bussche, 2017] recognises "the citizens' right to explanation"[Goodman and Flaxman, 2017]
- most encompassing in the same acceptation of the term 'explanation' both the explanator and the explainee acts
  - ! the dialectical notion of explanation
- whereas a notion of *explanation as an explanator's act* is where we mostly insist today
  - so that we can focus on the cognitive process of the explainee
  - and on the technical side of our intelligent systems, as well

Explanation as Representation & Transformation

- contribution from *math teaching* [D'Amore, 2005]
    - being math the most difficult subject to explain & teach
- a semiotic representation is required whenever the object of an explanation is inaccessible to perception

    noetics — *conceptual acquisition* of an object

    semiotics — acquisition of a *representation built out of signs*
- explaining a concept via different *semiotic representations*

    transformation of treatment — changing representation within the same register of semiotics

    transformation of conversion — changing register of semiotics for the representation
- *explanation* as
    - first, *generation of semiotic representation*
    - then, transformation of semiotic register
    - finally, sharing of the transformed representation

! explainers *share* their cognitive process with explainees as explanation

Humans Share Knowledge

- it is not brain size (or whatever like that) that separates humans from other intelligent animals like primates
  - instead, it is mostly our will to *share knowledge*[Dean et al., 2012]
- in general, knowledge sharing is a peculiar trait of humanity
  - it is how we do understand each other
  - it is how we learn
  - it is the foundation of human society
  - where human culture is a *cumulative* one

e.g. human science is a shared *social construct*
  - scientific artefacts are required to be *understandable* for the community
  - so as to enable *reproducibility* and *refutability* in the scientific process[Popper, 2002]

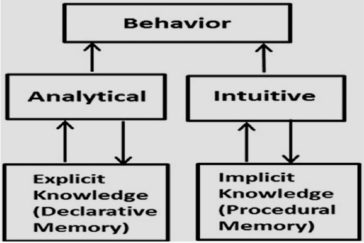
Sharing is Rational

- there is *intelligence without representation* [Brooks, 1991b] and *without reason* [Brooks, 1991a]
  - yet, human cumulative culture is based on *representation* tools—language, writing, books, the Web
- *repeatable*, *systematic* sharing requires rational representation
  - even when we are sharing *intuitive, implicit knowledge*
- and, sharing implicit knowledge typically calls for *rational* explanation

Cognition is (Not Just) Rational

Rationality vs. intuition

- two sorts of cognitive processes
  - esprit de finesse *vs.* esprit de géométrie—rationality has limits [Pascal, 1669]
  - *cognitivism* against *behaviourism* in psychology [Skinner, 1985]



- concepts and distinctions *not* born in the CS / AI fields
  - surely not in the ML community
- yet, they roughly match the two main families of AI techniques
  - symbolic vs. sub-/non-symbolic
  - informally, *classic AI* vs. *ML-based AI*
- and, the two sides of today intelligent systems

Focus on ML

- (Mostly) in ML, we let machines learn specific tasks from data
  - through the production of numeric predictors, a.k.a. black-boxes
  - instead of programming those tasks ourselves
- Unfortunately, black boxes are inherently
  - opaque w.r.t. the knowledge they acquire from data[Lipton, 2018]
  - sub-optimal in performance, as they are trained to minimise errors

Opaqueness

Opaqueness of ML-based predictors brings several *drawbacks*:[Guidotti et al., 2018, Lipton, 2018]

- difficulty in understanding what a black box has learned from data
  e.g. "snowy background" problem[Ribeiro et al., 2016]
- difficulty in spotting "bugs" in what a numeric predictor has learned
  - because that knowledge is not explicitly represented
- several blatant failures of ML-based systems reported so far
  e.g. black people classified as gorillas[Crawford, 2016]
  e.g. wolves classified because of snowy background[Ribeiro et al., 2016]
  e.g. unfair decisions in automated legal systems[Wexler, 2017]
- lawmakers recognised citizens' right to meaningful explanations[Selbst and Powles, 2017]
  - about the logic behind automated decision making
  e.g. in General Data Protection Regulation (GDPR) [Parliament and Council, 2016]

The Problem with ML-based AI

Trustworthiness

How can we trust machines we do not fully control?

↓

Controllability

How can we control machines we do not fully understand?

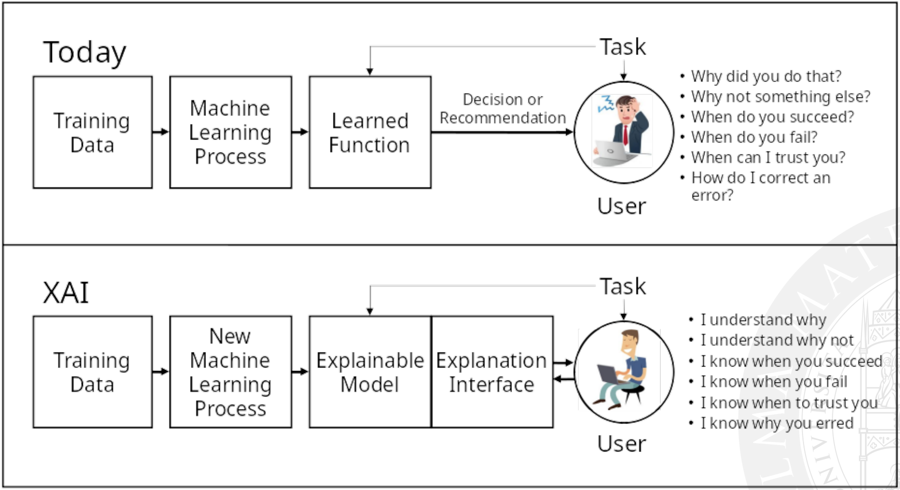
↓

Understandability

How can we understand distributed, numeric representations of knowledge?

The eXplanable AI (XAI) Approach [Gunning, 2016a]

The XAI community is nowadays facing those understandability issues



Next in Line...

Focus on. . .

Relevant Questions for XAI

1. What are we trying to explain?
   - in general, AI-based systems
2. Who is in charge of producing explanations?
   - the AI system itself? human experts? ordinary users?
3. To whom are explanations addressed?
   - humans (developers, end users)? other AI systems?
4. How are we going to create explanations?
   - this is the actual core of XAI research
5. Which are the most adequate sorts of explanation?
   - this depends on the answers to the questions above
6. When should explanations be presented to the user?
   - this, too, depends on the answers to the questions above

Current Practice of XAI

1. What are we trying to explain?
   - mostly data-driven, ML-powered systems
2. Who is in charge of producing explanations?
   - AI experts, data scientists, ML engineers
3. To whom are explanations addressed?
   - people having a certain degree of expertise in AI/ML
4. How are we going to create explanations?
   - via task-, model-, and data-specific algorithms
5. Which are the most adequate sorts of explanation?
   - depends on task, model, data, and consumer at hand
   - other than on the available XAI algorithms
6. When should explanations be presented to the user?
   - mostly in the training phase; possibly in inference phase

The Future of XAI

1. What are we trying to explain?
   - any system including computational agents with some degree of autonomy
2. Who is in charge of producing explanations?
   - the system, i.e., the agents themselves
3. To whom are explanations addressed?
   - people with diverse levels of expertise
   - other computational agents
4. How are we going to create explanations?
   - via task-, model-, and data-specific algorithms
   - plus consumer-specific presentation strategies
5. Which are the most adequate sorts of explanation?
   - the ones which better adapt to the needs of the user
6. When should explanations be presented to the user?
   - upon request—i.e., as part of a dialogue

Focus on. . .

Explain What? I

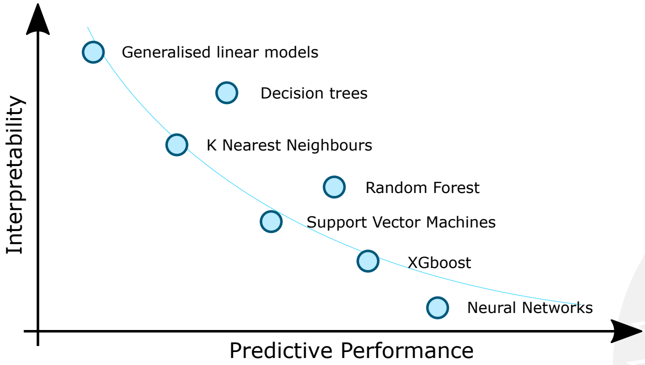
Most efforts are devoted to *supervised* ML, and in particular:

- specific sorts of tasks, e.g. classification and regression
- specific sorts of data, e.g. images, text, or tables
- specific sorts of predictors, e.g. neural networks, SVM

  i.e. essentially, functions of the form $f : \mathcal{X} \subseteq \mathbb{R}^n \rightarrow \mathcal{Y} \subseteq \mathbb{R}^m$

Explain What? II

Interpretability–Predictivity trade-off:



Explain What? III

Conventionally...

- ...linear models, or decision trees/rules are considered interpretable
- ...other kinds of predictors are considered poorly interpretable
  - hence in need of explanation

Explain What? IV

Our focus is on *supervised ML*, but XAI is wider than that

- explainable *unsupervised* learning—e.g., clustering [Sabbatini and Calegari, 2022]
- explainable reinforcement learning (XRL)[Milani et al., 2022]
- explainable planning (XAIP)[Hoffmann and Magazzeni, 2019]
- explainable agents and robots (XMAS)[Ciatto et al., 2019, Anjomshoae et al., 2019]
- ...

Focus on. . .

Global vs. Local Explanations I

Global explanation

- How does a predictor produce its outcomes in general?
  e.g. how does a neural network classify images of animals?

Local explanation

- How did a predictor produce a particular outcome?
  e.g. why did the neural network classify that image as a cat?

Global vs. Local Explanations II

About the global/local dichotomy

- firstly introduced in [Ribeiro et al., 2016]
- along with LIME, i.e. one of the earliest and most successful XAI techniques

Global vs. Local Explanations III

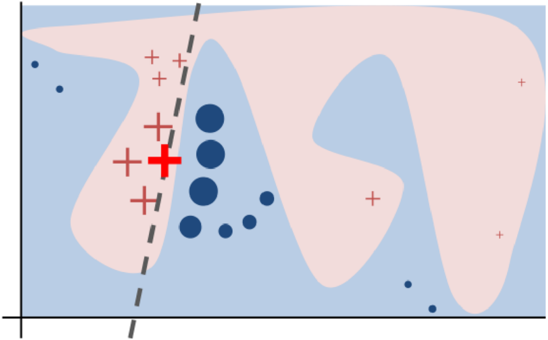
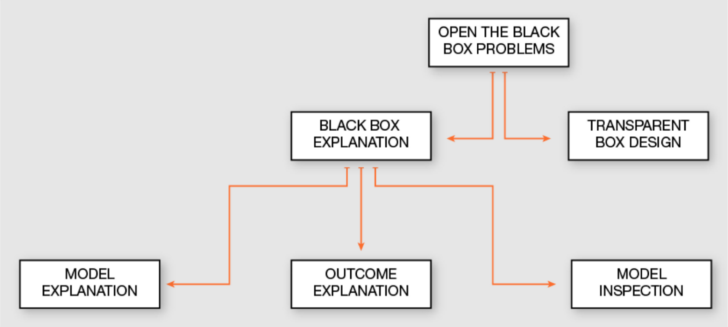


Figure: [Ribeiro et al., 2016] Toy example to present intuition for LIME. The black-box model's complex decision function f (unknown to LIME) is represented by the blue/pink background, which cannot be approximated well by a linear model. The bold red cross is the instance being explained. LIME samples instances, gets predictions using f, and weighs them by the proximity to the instance being explained (represented here by size). The dashed line is the learned explanation that is locally (but not globally) faithful.

Overview on XAI approaches I

Four major approaches [Guidotti et al., 2018]



About notation

- “model” ≈ “predictor”

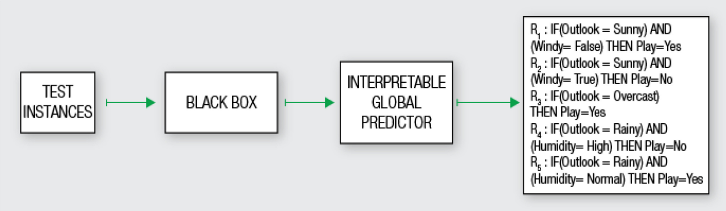
Overview on XAI approaches II

Model explanation ($\approx$ global explanation)

explanation $\approx$ interpretable predictor trained to mimic the one to be explained

- w.r.t. the entire input space

  e.g. surrogate models (e.g. decision trees)



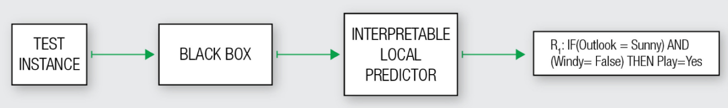
Overview on XAI approaches III

Outcome explanation (≈ local explanation)

explanation ≈ interpretable predictor trained to mimic the one to be explained

- w.r.t. a small portion of the input space

e.g. saliency maps—e.g. LIME[Ribeiro et al., 2016], SHAP[Lundberg and Lee, 2017]



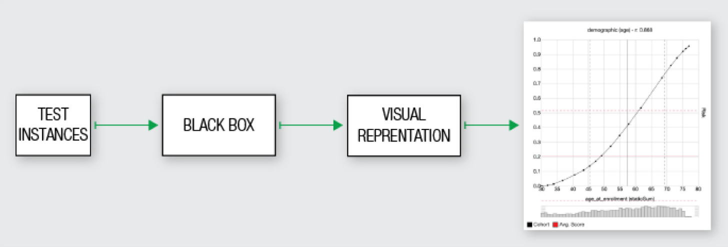
Overview on XAI approaches IV

Model inspection

explanation ≈ representation summarising the behaviour of the predictor to be explained

- w.r.t. a given portion of the input space (or, possibly, all of it)

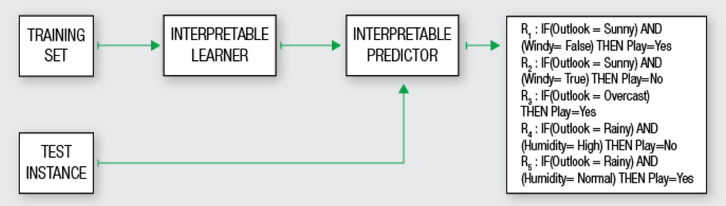
  e.g. feature importance, sensitivity analysis



Overview on XAI approaches V

Transparent box design

- just train an interpretable predictor and look at it



Focus on. . .

Interpretation or Explanation?

The two terms are not synonyms

- in spite of the fact that they are often used interchangeably

Insights

interpretation $\approx$ binding objects with meaning

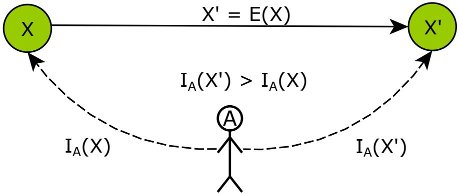
- that is what the human mind does

explanation $\approx$ eliciting relevant aspects of objects—to ease their interpretation

The Role of Representations

! this is just a representation of a pipe

An Abstract Framework for XAI [Ciatto et al., 2020] I



- X object to be explained
- A observer agent
- $I_A(\cdot)$ a function "measuring" the "degree of interpretability" of X, w.r.t. A
- $E(\cdot)$ an explanation function, mapping objects into (different) objects
- X' the result of the explanation, i.e. a more-interpretable object

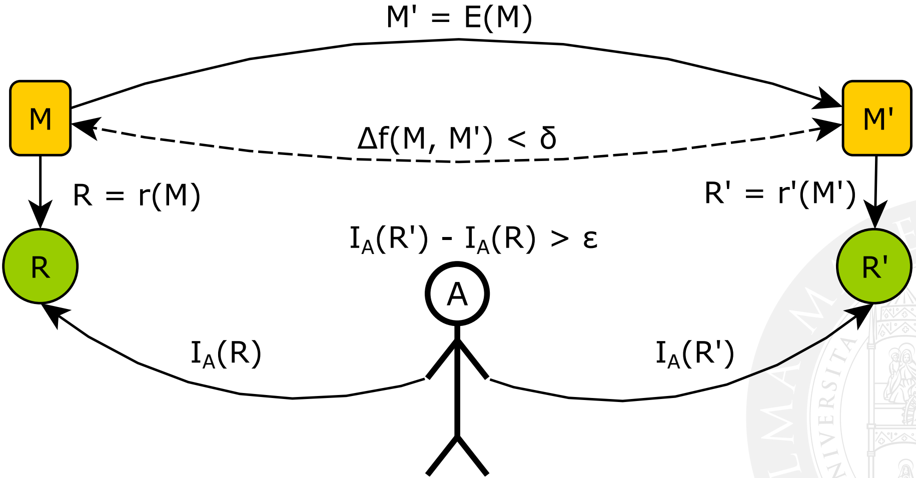
An Abstract Framework for XAI[Ciatto et al., 2020] II

Key points

- interpretation is subjective
- explanation is an operation transforming poorly interpretable objects into more-interpretable ones
- ‘interpretability’ does not need to be measurable (only comparisons matter)

An Abstract Framework for XAI [Ciatto et al., 2020] III

In the particular case of ML-based AI:



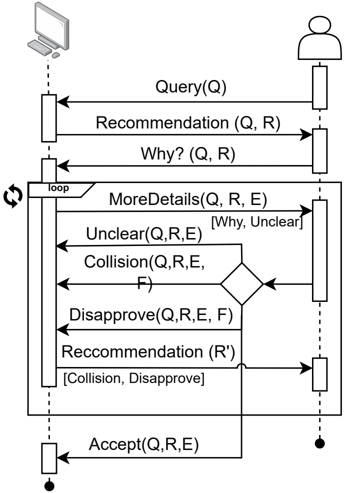
An Abstract Framework for XAI[Ciatto et al., 2020] IV

- we need to explain a model M
  - having a poorly interpretable representation R (w.r.t. A)
- explanation produces another model M'
  - having an interpretable representation R' (w.r.t. A)
- performance difference among M and M' (i.e. $\Delta f(M, M')$) must be small ($< \delta$)
  - or, dually, M' must have an high fidelity w.r.t. M

Key points

- explanation $\approx$ search of a surrogate interpretable model
- representation is important as much as explanation
- explanation must maximise fidelity

The Role of Interaction



- explanation as an interaction protocol
  - among an explainer/recommender
  - and explainee
- possibly repeating the protocol several times . . .
- . . . until selecting the explanation/representation which better suits the explainee

Next in Line...

1. AI, ML & XAI
2. XAI Background
3. Explanations via Feature Importance
4. Explanations via Symbolic Knowledge Extraction
5. Transparent Box Design via Symbolic Knowledge Injection
6. XAI in Practice

Overview I

Insight

- quantify each *input feature*'s contribution to

  a single prediction (*local* explanation)
  the predictor's behavior in general (*global* explanation)

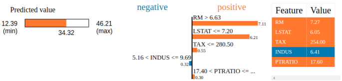
- possibly, select the most relevant features
  - i.e. the ones contributing the most

- represent the importance score accordingly
  - the representation depends on the sort of data at hand

Overview II

Which sorts of data?

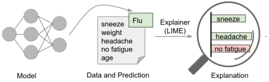
- tabular data → named features — explained via histograms



- images → (super-)pixels — explained via masks / heatmaps

(a) Husky classified as wolf (b) Explanation

- text → bag of words / TD-IDF / Word2Vec — explained via words



Overview III

General Remarks about Feature Importance

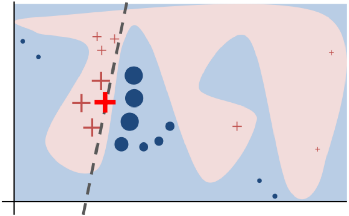
- may be used to explain either the model or the outcome
- in both cases, explanations are provided by model inspection
  → data-specific representations play a crucial role
- feature selection is a by-product of the explanation process
- feature importance computation is commonly
  model agnostic (i.e., it works with any sort of ML predictor)
  post-hoc (i.e., it occurs after predictors' training)

Focus on. . .

Overview I

- LIME = Local Interpretable Model-agnostic Explanations[Ribeiro et al., 2016]
- model-agnostic and post-hoc means for outcome explanation
  - works by constructing a local surrogate model around the prediction to be explained
  - the predictor to be explained acts as an oracle
- may also be exploited as a means for model explanation
  - by averaging multiple outcome explanations

Overview II



To explain a prediction $y = f(\bar{x})$ s.t. $\bar{x} = (x_1, \ldots, x_i, \ldots, x_n)$, LIME:

- trains an interpretable model g
  - approximating f in the surroundings of $\bar{x}$
- uses g to compute how much each x_i contributes to y

Interpretable models could be:

- linear models
- decision trees

Algorithm Overview I

Assumptions and prerequisites

- Input features may be of any sort (numeric, categorical, pixel, etc.)
- Binary interpretable components must be defined for each feature

categorical	feature $\leftrightarrow$ one-hot encoding
numeric	feature $\leftrightarrow$ bin discretization
BOW	feature $\leftrightarrow$ word presence/absence
pixel	feature $\leftrightarrow$ super-pixel presence/absence

  - the mapping among features and components must be reversible
- A measure of proximity / similarity to $\bar{x}$

Algorithm Overview II

About notation

- $\bar{x} \in \mathbb{R}^n \equiv (x_1, \ldots, x_n)$ is the input vector containing the original features
- $\bar{x}' \in \{0,1\}^m \equiv (x'_1, \ldots, x'_m)$ is the corresponding vector of interpretable components
- $f : \mathbb{R}^n \to \mathcal{Y}$ is the predictor to be explained
- $g : \{0,1\}^m \to \mathcal{Y}$ is the interpretable predictor constructed by LIME
- $\pi_{\bar{x}}(\bar{z}) : \mathbb{R}^n \to [0,1]$ is the proximity measure of some input point $\bar{z}$ w.r.t. some pivot point $\bar{x}$

Algorithm Overview III

Algorithm overview

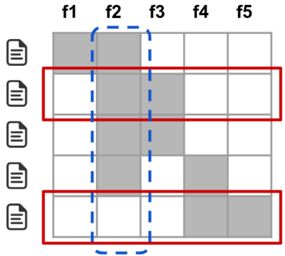
1. Sample N points $\bar{z}_1, \ldots, \bar{z}_N$ around $\bar{x}$ according to $\pi_{\bar{x}}$
2. For each $\bar{z}_i$
   1. compute the corresponding interpretable components $\bar{z}_i'$ ...
   2. ... and prediction $y_i = f(\bar{z}_i)$
3. Use the data items $\langle \bar{z}_i,\ y_i \rangle$ to train g
   - g is trained to perform regularization
4. Repeat the process with different hyper-parameters of g
5. Select the g which
   - maximises the fidelity of g w.r.t. f
   - while minimizing the complexity of g
6. Use the coefficients of g as measures of feature importance
   - select the K-best coefficients

Algorithm Overview IV

Hyper-parameters of LIME

- N: amount of samples generated to explain a single prediction $\bar{x}$
- K: maximum amount of important features to be selected
- g: sort of the interpretable model to be trained (e.g., linear, tree)
    - this commonly implies the sort of *regularization* to be used
- reversible mapping between features and interpretable components
    - essentially, a binarization process

From local to global LIME



1. Select M pivot points X from the input space
2. For each $\bar{x}_i \equiv (x_{i,1}, \ldots, x_{i,j}, \ldots, x_{i,n'}) \in X$ compute K-best feature importance
   - produce a $M \times n'$ matrix W ...
   - ... where cell $w_{i,j}$ is the importance of the j-th component of $\bar{x}_i$
3. Aggregate W column-wise to get global feature importances

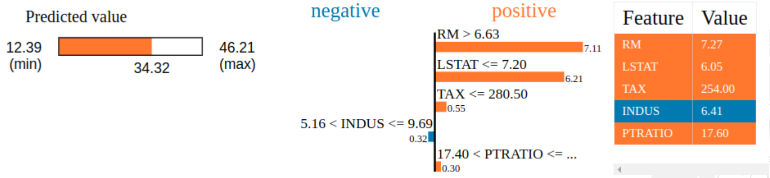
Major issues

- How to select the N pivot points?
- It only works if all instances have the same features

About LIME's outputs I

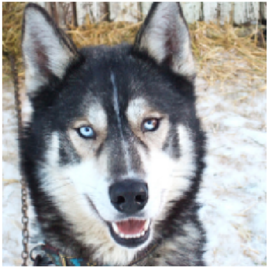
Representation of results is quintessential with feature importance:

- in tabular data, we may represent the contribution of feature intervals:



About LIME's outputs II

- in images, we may highlight the contribution of patches:

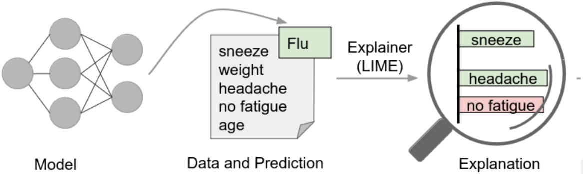


(a) Husky classified as wolf

(b) Explanation

About LIME's outputs III

- in text, we may highlight the contribution of individual tokens:



Focus on...

Discussion

Pros

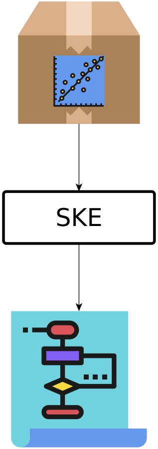
- clear and intuitive interpretation of predictions
- applicable to any sort of supervised predictor
- adaptable to many sorts of data
- computational effort is parametric

Cons

- more a tool for debugging than a means for explanation
- requires a lot of pre-processing
- may *not* fit all sorts of features

Next in Line. . .

Overview I



Insight

- search of a surrogate interpretable model. . .
- . . . consisting of symbolic knowledge

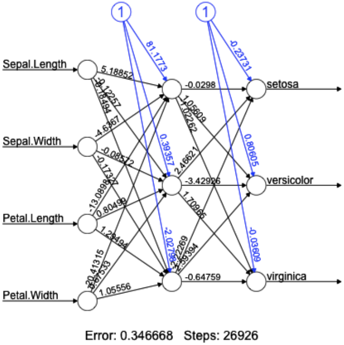
Overview II

Definition

Any algorithmic procedure accepting trained sub-symbolic predictors as input and producing symbolic knowledge as output, in such a way that the extracted knowledge reflects the behaviour of the predictor with high fidelity.

Overview III

Example:



$\rightarrow$

$$Class = \texttt{setosa} \leftarrow PetalWidth \leq 1.0.$$

$$Class = \texttt{versicolor} \leftarrow PetalLength > 4.9 \wedge SepalWidth \in [2.9,\ 3.2].$$
$$Class = \texttt{versicolor} \leftarrow PetalWidth > 1.6.$$

$$Class = \texttt{virginica} \leftarrow SepalWidth \leq 2.9.$$

$$Class = \texttt{virginica} \leftarrow SepalLength \in [5.4,\ 6.3].$$
$$Class = \texttt{virginica} \leftarrow PetalWidth \in [1.0,\ 1.6].$$

What does ‘symbolic’ actually mean? I

Symbolic representations of knowledge[van Gelder, 1990]

- involves a set of symbols,
- which can be combined (e.g., concatenated) in (possibly) infinitely many ways,
- following precise syntactical rules, and
- where both elementary symbols and any admissible combination of them can be assigned with meaning
  - ie each symbol can be mapped into some entity from the domain at hand.

Notable example

- formal logic

What does 'symbolic' actually mean? II

Opposite notion: **distributed** representations

- where symbols alone have no meaning
- unless it is considered along with its neighbourhood
  - ie any other symbol which is close (according to some notion of closeness)

Plenty of SKE methods from the literature I

Table: Summary of the knowledge-extraction algorithms. Symbol $*$ means that the related dimension of the algorithm is not bounded. Symbol † means that the output is a power law.

#	Method	Translucency	Task	Input	Expressiveness	Shape
1	[Breiman et al., 1984]	P	C+R	C+D	P	DT
2	[Quinlan, 1986]	P	C	D	P	DT
3	[Saito and Nakano, 1988]	P	C	D	P	L
4	[Clark and Niblett, 1989]	P	C	C+D	P	L
5	[Masuoka et al., 1990]	D (NN)	C	C	F	L
6	[Hayashi, 1990]	D (NN)	C	B	F	L
7	[Towell and Shavlik, 1991]	D (NN)	C	D	MN	L
8	[Berenji, 1991]	D (NN)	C	C	F	L
9	[Brunk and Pazzani, 1991]	P	C	C+D	P	L
10	[Murphy and Pazzani, 1991]	P	C	D	MN	DT
11	[Horikawa et al., 1992]	D (NN)	C	C	F	L
12	[Tresp et al., 1992]	D (NN)	R	C	P	L
13	[Towell and Shavlik, 1993]	D (NN)	C	D	P	L
14	[Thrun, 1993]	D (NN)	C	C	P+MN	L
15	[Cohen, 1993]	P	C	C+D	P	L

Plenty of SKE methods from the literature II

16	[Quinlan, 1993]	P	C	C+D	P	DT
17	[Fu, 1994]	D (NN)	C	D	P	L
18	[Halgamuge and Glesner, 1994]	D (NN)	C	C	F	L
19	[Mitra, 1994]	D (NN)	C	C+D	F	L
20	[Craven and Shavlik, 1994]	P	C	B	P+MN	L
21	[Fürnkranz and Widmer, 1994]	P	C	D	P	L
22	[Sestito and Dillon, 1994]	P	C	C	P	L
23	[Andrews and Geva, 1995]	D (NN)	C	C+D	P	L
24	[Matthews and Jagielska, 1995]	D (NN)	C	B	F	L
25	[Cohen, 1995]	P	C	C+D	P	L
26	[Pop et al., 1994]	P	C	B	P	L
27	[Setiono and Liu, 1996]	D (NN)	C	B	P	L
28	[Tickle et al., 1996]	P	C	B	P	L
29	[Yuan and Zhuang, 1996]	P	C	D	F	L
30	[Craven and Shavlik, 1996]	P	C	B	P+MN	DT
31	[Hong and Lee, 1996]	P	C	C	F	L
32	[Setiono and Liu, 1997]	D (NN3)	C	C+D	O	L
33	[Setiono, 1997]	D (NN)	C	D	P	L
34	[Nauck and Kruse, 1997]	D (NN)	C	D	F	L

Plenty of SKE methods from the literature III

35	[Saito and Nakano, 1997]	D (NN)	R	C	†	†
36	[Benítez et al., 1997]	D (NN)	C+R	C	F	L
37	[Ishibuchi et al., 1997]	P	C	C	F	L
38	[Taha and Ghosh, 1999]	D (NN)	C	C	P	L
39	[Taha and Ghosh, 1999]	D (NN)	C	C	P	L
40	[Krishnan et al., 1999b]	D (NN)	C	B	P	L
41	[Nauck and Kruse, 1999]	D (NN)	R	D	F	L
42	[Taha and Ghosh, 1999]	P	C	B	P	L
43	[Krishnan et al., 1999a]	P	C	C	P	DT
44	[Schmitz et al., 1999]	P	C+R	C+D	P	DT
45	[Hong and Chen, 1999]	P	C	C	F	L
46	[Setiono, 2000]	D (NN)	C	B	MN	L
47	[Tsukimoto, 2000]	D (NN)	C	C+D	P	L
48	[Kim and Lee, 2000]	D (NN4)	C	C+D	P	DT
49	[Setiono and Leow, 2000]	D (NN)	R	C+D	P+MN+O	DT
50	[Zhou et al., 2000]	P	C	C+D	P	L
51	[Hong and Chen, 2000]	P	C	C	F	L
52	[Sato and Tsukimoto, 2001]	D (NN3)	R	C+D	P	DT
53	[Parpinelli et al., 2001]	P	C	C+D	P	L

Plenty of SKE methods from the literature IV

54	[Castillo et al., 2001]	P	C+R	C+D	F	L
55	[Saito and Nakano, 2002]	D (NN)	R	C+D	P	L
56	[Setiono et al., 2002]	D (NN3)	R	C+D	P	L
57	[Liu et al., 2002]	P	C	C+D	P	L
58	[Boz, 2002]	P	C	C+D	P	DT
59	[Markowska-Kaczmar and Trelak, 2003]	P	C	C+D	F	L
60	[Zhou et al., 2003]	P	C	C+D	P	L
61	[Setiono and Thong, 2004]	D (NN3)	R	C+D	P	L
62	[Fu et al., 2004]	D (SVM)	C	C+D	P	L
63	[Markowska-Kaczmar and Chumieja, 2004]	P	C	C+D	P	L
64	[Rabuñal et al., 2004]	P	C	C+D	P	L
65	[Chen, 2004]	P	C	C	P	L
66	[Liu et al., 2004]	P	C	C+D	P	L
67	[Browne et al., 2004]	P	C	C+D	P+MN	DT
68	[Zhang et al., 2005]	D (SVM)	C	C	P	L
69	[Barakat and Diederich, 2008]	D (SVM)	C+R	*	*	*
70	[Fung et al., 2005]	D (SVM+LC)	C	C	P	L
71	[Chaves et al., 2005]	D (SVM)	C	C	F	L
72	[Torres and Rocco, 2005]	P	C	C+D	P+MN	DT

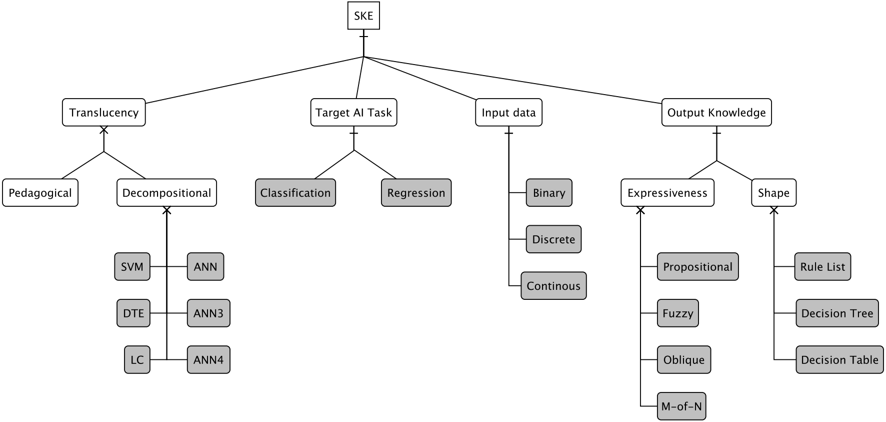
Plenty of SKE methods from the literature V

73	[Etchells and G., 2006]	P	C	C+D	P	L
74	[He et al., 2006]	P	C	C+D	P	DT
75	[Huysmans et al., 2006]	P	R	C	P	L
76	[Bader et al., 2007]	D (NN)	C	B	P	L
77	[Schetinin et al., 2007]	D (DTE)	R	C	P	DT
78	[Chen et al., 2007]	D (SVM)	C	C	P	L
79	[Barakat and Bradley, 2007]	D (SVM)	C	C+D	P	L
80	[Saad and Wunsch II, 2007]	P	C	C+D	O	L
81	[Martens et al., 2007]	P	C	C+D	P	L
82	[Núñez et al., 2008]	D (SVM)	C	C	P+O	L
83	[Setiono et al., 2008]	P	C	C+D	P+O	L
84	[Odajima et al., 2008]	P	C	D	P	L
85	[Konig et al., 2008]	P	C+R	C+D	F	DT
86	[Bader, 2009]	D (NN)	C	B	P	L
87	[Martens et al., 2009]	D (SVM)	C	*	*	*
88	[Lehmann et al., 2010]	P	C	B	P	L
89	[Augasta and Kathirvalavakumar, 2012]	P	C	C+D	P	L
90	[Sethi et al., 2012]	P	C	C+D	P	TA
91	[Zilke et al., 2016]	D (NN)	R	C+D	P	DT

Plenty of SKE methods from the literature VI

92	[Chan and Chan, 2017]	D (NN)	R	C	P	L
93	[Yedjour and Benyettou, 2018]	P	C	B	P	L
94	[Chan and Chan, 2020]	D (NN)	R	C	P	L
95	[Wang et al., 2020]	D (DTE)	C	C	P	L
96	[Sabbatini et al., 2021]	P	R	C	P	L

Taxonomy of SKE methods I



Taxonomy of SKE methods II

target AI task for the predictor undergoing extraction

- **classification** i.e., $f : \mathcal{X} \subseteq \mathbb{R}^n \to \mathcal{Y}$ s.t. $|\mathcal{Y}| = k$
- **regression** i.e., $f : \mathcal{X} \subseteq \mathbb{R}^n \to \mathcal{Y} \subseteq \mathbb{R}^m$

translucency what kind of ML predictor does the SKE method support?

- **pedagogical:** any supervised predictor
- **decompositional:** a particular sort of ML predictor (e.g. NN, SVM, DT)

input data supported by the predictor undergoing extraction

- **binary:** $\mathcal{X} \equiv \{0, 1\}^n$
- **discrete:** $\mathcal{X} \in \{x_1, \ldots, x_n\}^n$
- **continuous:** $\mathcal{X} \subseteq \mathbb{R}^n$

Taxonomy of SKE methods III

shape of the extracted knowledge

- rule list: i.e. ordered sequences of if-then-else rules
- decision tree: hierarchical set of if-then-else rules involving a comparison among a variable and a constant
- decision table: 2D tables summarising decisions for each possible assignment of variables

Taxonomy of SKE methods IV

expressiveness of the extracted knowledge

propositional: boolean statements + logic connectives
- there including arithmetic comparisons among variables and constants

fuzzy: hierarchical set of if-then-else rules involving a comparison among a variable and a constant

oblique: boolean statements + logic connectives + arithmetic comparisons

M-of-N: any of the above + statements like $m - \text{of} - \{\phi_1, \ldots, \phi_n\}$

Examples of methods and their classification – CART I

CART:[Breiman et al., 1984] classification and regression trees

- **translucency:** pedagogical
- **target AI task:** classification OR regression
- **input data:** binary OR discrete OR continuous
- **shape:** decision tree
- **expressiveness:** propositional

Examples of methods and their classification – CART II

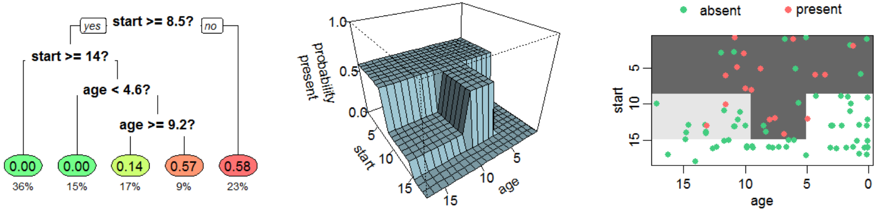


Figure: An example decision tree estimating the probability of kyphosis after spinal surgery, given the *age* of the patient and the vertebra at which surgery was *start*ed[Wikipedia contributors, 2021]. Notice that all decision trees subtend a partition of the input space, and that those trees themselves provide intelligible representations of *how* predictions are attained.

Examples of methods and their classification – CART III

Using CART for SKE

1. generate a ‘fake’ dataset by feeding the predictor undergoing SKE
2. train a decision tree on the ‘fake’ dataset
3. compute fidelity and repeat step 2 until satisfied
4. [opt.] rewrite the tree as a list of rules

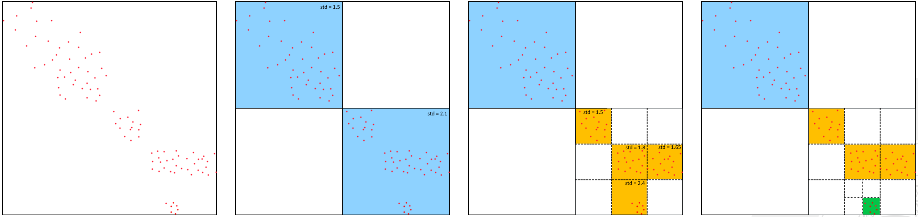
Examples of methods and their classification – GridEx I

GridEx:[Sabbatini et al., 2021] grid extractor

- **translucency:** pedagogical
- **target AI task:** regression
- **input data:** continuous
- **shape:** rule list
- **expressiveness:** propositional

Examples of methods and their classification – GridEx II

Figure: Example of GridEx's hyper-cube partitioning (merging step not reported)



(a) Surrounding cube

(b) Iteration 1 ($p_1 = 2$)

(c) Iteration 2 ($p_2 = 3$).

(d) Iteration 3 ($p_3 = 2$).

Examples of methods and their classification – GridEx III

Using GridEx for SKE

1. partition the input space into p_1^n hypercubes
   - evenly splitting the n dimensions into p_1 bins
2. partition each non empty-region into p_2^n hypercubes
   - evenly splitting the n dimensions into p_2 bins
3. repeat the splitting arbitrarily
4. assign a prediction with each non-empty partition (e.g. average value)
5. write an if-then rule for each non-empty partition:
   - *if*: expressions delimiting the partition
   - *then*: prediction of that partition

Examples of methods and their classification – REFANN I

REFANN:[Setiono et al., 2002] rule extraction from function approximating NN

- **translucency:** decompositional (3-layered NN)
- **target AI task:** regression
- **input data:** continuous OR discrete
- **shape:** rule list
- **expressiveness:** propositional

Examples of methods and their classification – REFANN II

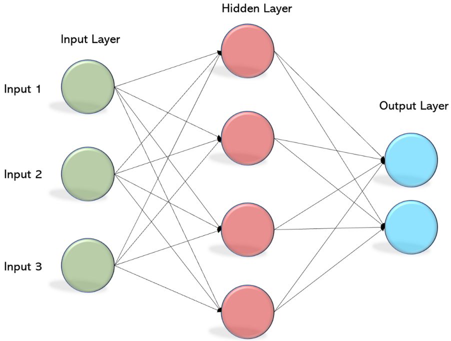


Figure: An example 3-layered multi-layer perceptron (MLP)

Examples of methods and their classification – REFANN III

Using REFANN for SKE

1. prune the network's hidden units and input neurons
2. approximate the hidden units' activation function with a 2-steps-wise linear function
3. approximate the output units' activation function with a 3- or 5-step-wise linear function
4. rewrite each output neuron as a linear combination of the input neuron
5. rewrite the linear combinations as rules
    - hence attaining a list of rules

Examples of methods and their classification – REFANN IV

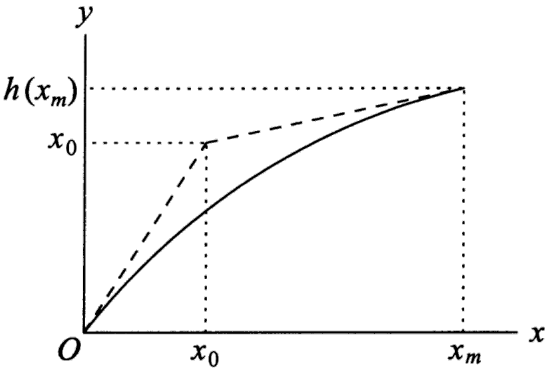


Figure: [Setiono et al., 2002] The *tanh*(x) function (solid curve) for $x \in [0, x_m]$ is approximated by a 2-piece linear function (dashed lines)

Examples of methods and their classification – REFANN V

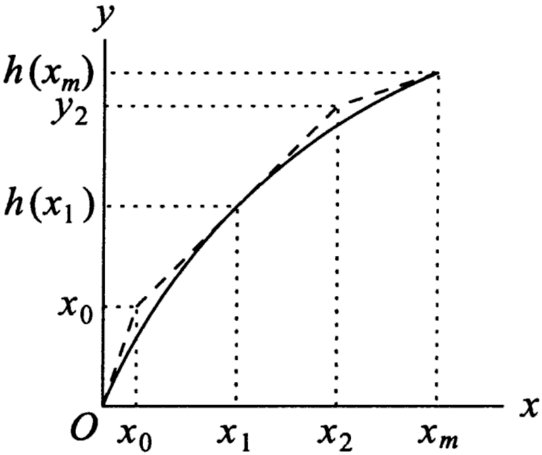


Figure: [Setiono et al., 2002] The *tanh*(x) function (solid curve) for $x \in [0, x_m]$ is approximated by a 3-piece linear function (dashed lines)

Focus on. . .

Notable Remarks

- commitment to a particular output shape / expressiveness
    - to preserve both human- and machine-interpretability
    - other syntaxes may exist
- discretization of the input space
- discretization of the output space
- features should have semantics per se
- further refinements may be applied to rules
- rules constitute global explanations

Current Limitations

- tabular data as input $\rightarrow$ doesn't really work with images
- high dimensional datasets $\rightarrow$ very large, poorly readable rules
- highly variable input spaces $\rightarrow$ many rules $\rightarrow$ poor readability

Future research activities

- target images or highly dimensional data in general
- target reinforcement learning (when based on NN)
- target unsupervised learning
- design and prototype your own extraction algorithm

Next in Line...

Why SKI?

There are several benefits:

- prevent the predictor to become a black-box!;
- reduce learning time;
- reduce the data size needed for training;
- improve predictor's accuracy;
- build a predictor that behave as a logic engine.

Symbolic Knowledge Injection I

Key insights:

- Altering ML predictors. . .
- . . . to make they comply to user-provided knowledge. . .
- . . . which is represented in symbolic form

Symbolic Knowledge Injection II

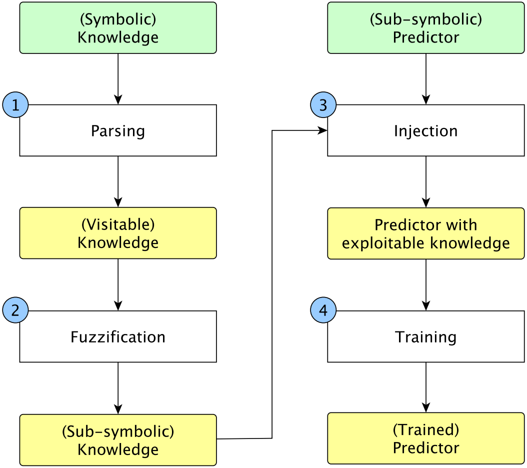
We define SKI as:

any algorithmic procedure affecting how sub-symbolic predictors draw their inferences in such a way that predictions are either computed as a function of, or made consistent with, some given symbolic knowledge*.

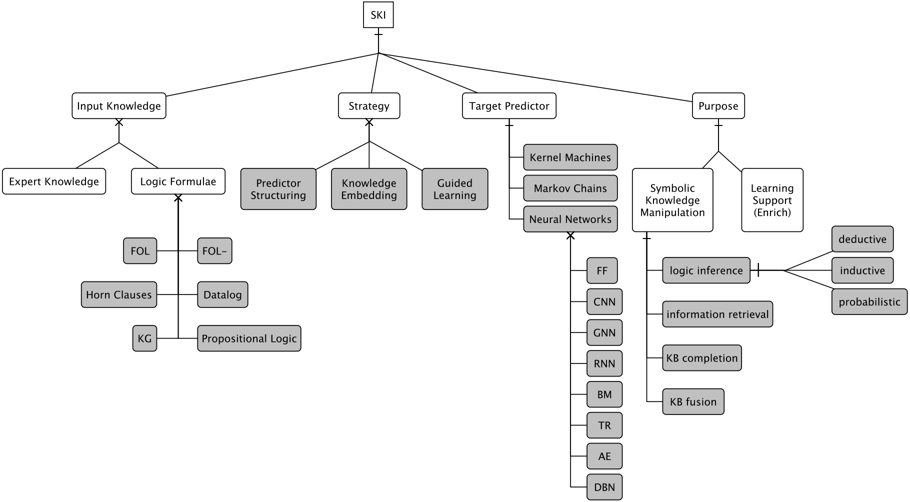
* a wide definition that includes the vast majority of the works in the main surveys [Besold et al., 2017, Xie et al., 2019, Calegari et al., 2020].

Symbolic Knowledge Injection III

General workflow:



Taxonomy of SKI methods I



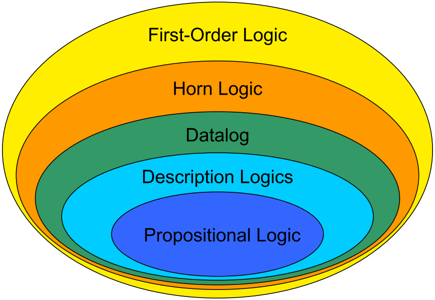
Taxonomy of SKI methods II

- input knowledge how is the knowledge to-be-injected represented?
  - commonly, some sub-set of first-order logic (FOL)
- target predictor which predictors can knowledge be injected into?
  - mostly, neural networks
- strategy how does injection actually work?
  - guided learning the input knowledge is used to guide the training process
  - structuring the internal composition of the predictor is (re-)structured to reflect the input knowledge
  - embedding the input knowledge is converted into numeric array form
- purpose why is knowledge injected in the first place?
  - knowledge manipulation improve / extend / reason about symbol knowledge—subsymbolically
  - learning support improve the sub-symbolic predictor (e.g. speed, size, etc.)

Focus on. . .

About Logic I

How to represent knowledge?



- *expressiveness–tractability* trade-off [Levesque and Brachman, 1987, Brachman and Levesque, 2004]

About Logic II

In practice, virtually all SKI algorithms deal with:

- datalog;
- description logics (a.k.a. knowledge graph, KG);
- propositional logic (PL).

First Order Logic I

Overview

- FOL is extremely flexible and expressive
  - variables, quantifiers, structured terms, negation, logic connectives
- one can use recursion to define recursive structures;
  - possibly, intensionally—i.e. without extensively describing everything
- maybe too "powerful" for canonical NN
  - most NN are essentially DAG
  - training via backpropagation[Baldi and Sadowski, 2016] requires no cycles

$\rightarrow$ recursion not supported

First Order Logic II

Example of FOL knowledge base (Peano numbers)

$$\mathit{natural}(\texttt{zero})$$
$$\forall X : \mathit{natural}(X) \rightarrow \mathit{natural}(\texttt{successorOf}(X))$$

Horn Clauses ($\approx$ Prolog) I

Overview

- sub-set of FOL with:
  - implicit quantifiers
  - limited set of logic connectives
- still supports recursion
- nice expressiveness–tractability trade-off
  - often exploited to design/realise automatic reasoning

Horn Clauses ($\approx$ Prolog) II

Example of Horn clauses (Peano numbers)

$$\begin{aligned}&\mathit{natural}(\mathtt{zero})\\&\mathit{natural}(\mathtt{successorOf}(X)) \leftarrow \mathit{natural}(X)\end{aligned}$$

Datalog I

Overview

- sub-set of Horn clauses with no recursion
- good for SKI!

Peano numbers in Datalog

- cannot be represented!
  - (as they require recursion)

Description Logics ($\approx$ Knowledge Graphs) I

Overview

- Very restricted subset of FOL
    - only constants, variables and n-ary predicates with $n \leq 2$;
- Everything is represented via collections of triplets of the form:

$$\langle \mathtt{a}\ f\ \mathtt{b} \rangle \text{ or } f(\mathtt{a}, \mathtt{b})$$

  where $\mathtt{a}, \mathtt{b}$ are entities, and f is a (binary) relationship
- essentially, directed graph:
    - nodes (i.e. entities) represent individuals
    - edges (i.e. relationships) represent relations among individuals

Description Logics ($\approx$ Knowledge Graphs) II

⟨AlfredHitchcock, DirectorOf, Psycho⟩

Sir Alfred Joseph Hitchcock (13 August 1899 – 29 April 1980) was an English film director and producer, ...

Psycho is a psychological horror film directed and produced by Alfred Hitchcock, and written by Joseph Stefano, ...

Propositional Logic I

Overview

- The simplest subset of FOL
    - no quantifiers, no terms, no n-ary predicates with $n > 0$
    - essentially, just Boolean algebra
- low expressiveness, but easy to work with

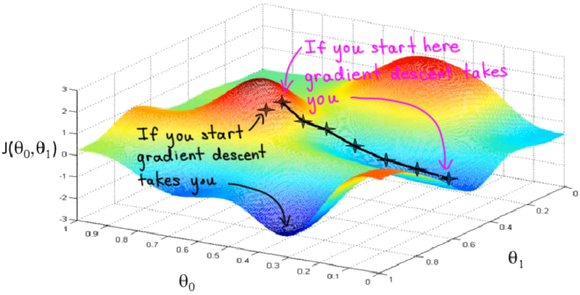
Propositional Logic II

Example

$$
\begin{aligned}
big\_petal \land average\_sepal &\rightarrow \texttt{virginica}. \\
big\_petal \land \neg average\_sepal &\rightarrow \texttt{versicolor}. \\
small\_petal &\rightarrow \texttt{setosa}. \\
average\_sepal &\equiv (3 \leq SepalWidth < 5) \\
big\_petal &\equiv (PetalLength > 3) \\
small\_petal &\equiv \neg big\_petal \equiv (PetalLength \leq 3)
\end{aligned}
$$

Focus on...

Strategy 1: Guided Learning I



- learning is essentially an optimizionation process
- ...often performed via gradient descent
  ie minimising a loss function

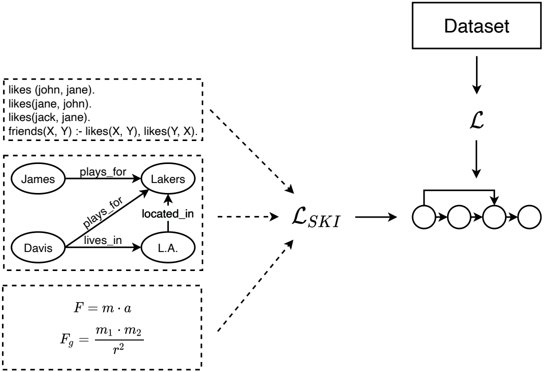
Strategy 1: Guided Learning II

SKI via Guided Learning

1. Input knowledge is converted into a cost factor
   - ie the more the knowledge is violated, the higher the cost
2. The loss function is altered to include that cost factor
   - e.g. as a simple additive regularisation factor
3. The predictor is then trained as usual

→ Training minimises both the predictors' error and inconsistency w.r.t. knowledge

Strategy 1: Guided Learning III



Strategy 2: Structuring I

SKI via Structuring

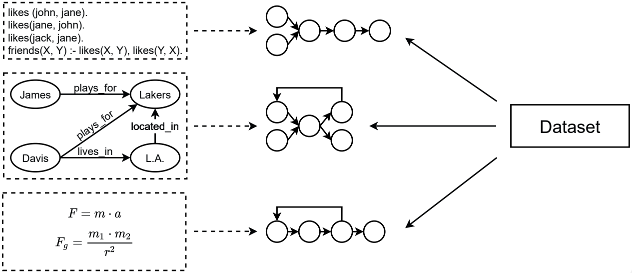
- The predictor's inner architecture is shaped to"mimic" the knowledge
- Shaping is predictor-dependent

  e.g. for neural networks, this means creating ad-hoc layers

  - where small groups of neurons are used to compute pieces of a formula

→ The predictor directly exploits the knowledge during inference

Strategy 2: Structuring II

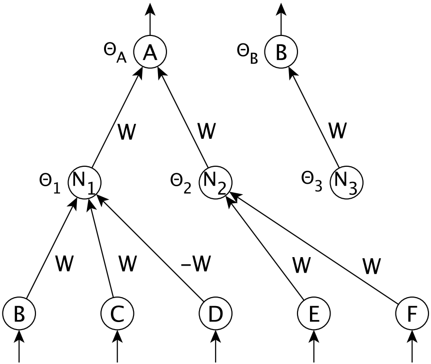


Strategy 2: Structuring III

Example:

$$A \leftarrow B \land C \land \neg D.$$
$$A \leftarrow E \land F.$$
$$B \leftarrow \texttt{true}.$$

$\leftrightarrow$

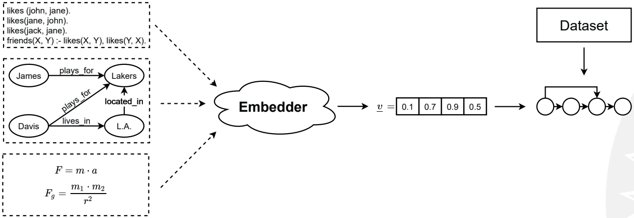


Strategy 3: Embedding I

SKI via Structuring

- Input knowledge is converted into numeric tensor(s)
- These are used as the training set for an ordinary learning process

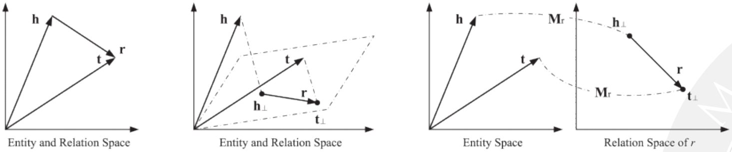
$\rightarrow$ The predictor is trained and used 'as usual'



Strategy 3: Embedding II

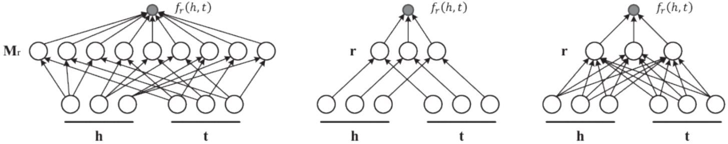
Example: knowledge graph embedding [Wang et al., 2017]

- entities and relations are embedded into continuos vector spaces;
- scoring function $f_r(h, t)$ defined on each fact (h, r, t) to measure its plausibility;

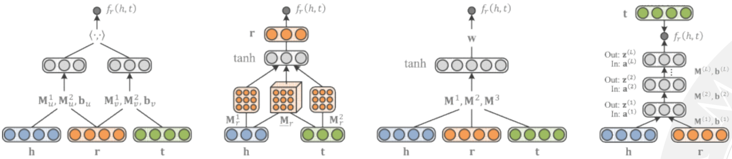


(a) TransE. (b) TransH. (c) TransR.

Strategy 3: Embedding III



(a) RESCAL. (b) DistMult. (c) HolE.



(a) SME. (b) NTN. (c) MLP. (d) NAM.

Focus on. . .

Knowledge Injection via Network Structuring [Magnini et al., 2022a] I

KINS

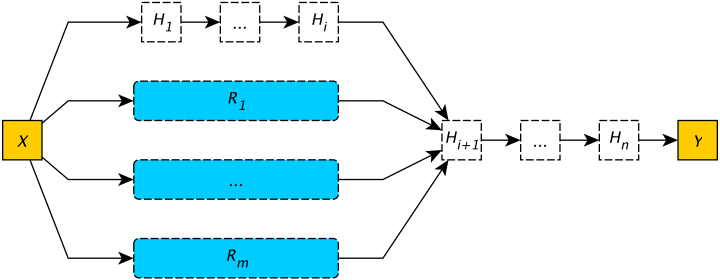
purpose → learning support

target predictor → neural networks

strategy → structuring

input logic → stratified Datalog with negation

Knowledge Injection via Network Structuring[Magnini et al., 2022a] II



Knowledge Injection via Network Structuring [Magnini et al., 2022a] III

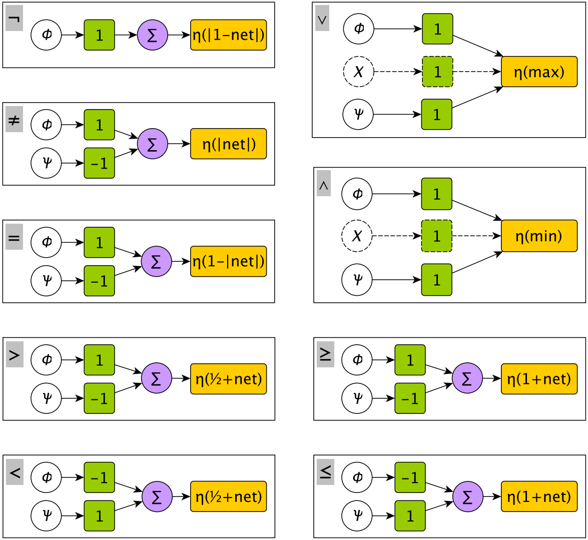
Formula	C. interpretation	Formula	C. interpretation
$[\![\neg\phi]\!]$	$\eta(1 - [\![\phi]\!])$	$[\![\phi \leq \psi]\!]$	$\eta(1 + [\![\psi]\!] - [\![\phi]\!])$
$[\![\phi \wedge \psi]\!]$	$\eta(\mathit{min}([\![\phi]\!], [\![\psi]\!]))$	$[\![\mathit{class}(\bar{X}, \mathtt{y}_i) \leftarrow \psi]\!]$	$[\![\psi]\!]^*$
$[\![\phi \vee \psi]\!]$	$\eta(\mathit{max}([\![\phi]\!], [\![\psi]\!]))$	$[\![\mathrm{expr}(\bar{X})]\!]$	$\mathrm{expr}([\![\bar{X}]\!])$
$[\![\phi = \psi]\!]$	$\eta([\![\neg(\phi \neq \psi)]\!])$	$[\![\mathtt{true}]\!]$	1
$[\![\phi \neq \psi]\!]$	$\eta(\lvert [\![\phi]\!] - [\![\psi]\!] \rvert)$	$[\![\mathtt{false}]\!]$	0
$[\![\phi > \psi]\!]$	$\eta(\frac{1}{2} + [\![\phi]\!] - [\![\psi]\!])$	$[\![X]\!]$	x
$[\![\phi \geq \psi]\!]$	$\eta(1 + [\![\phi]\!] - [\![\psi]\!])$	$[\![\mathtt{k}]\!]$	k
$[\![\phi < \psi]\!]$	$\eta(\frac{1}{2} + [\![\psi]\!] - [\![\phi]\!])$	$[\![p(\bar{X})]\!]^{**}$	$[\![\psi_1 \vee \ldots \vee \psi_k]\!]$

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

** assuming p is defined by k clauses of the form:

$$p(\bar{X}) \leftarrow \psi_1, \ \ldots, \ p(\bar{X}) \leftarrow \psi_k$$

Knowledge Injection via Network Structuring [Magnini et al., 2022a] IV



Knowledge Injection via Lambda Layer [Magnini et al., 2022b] I

KILL

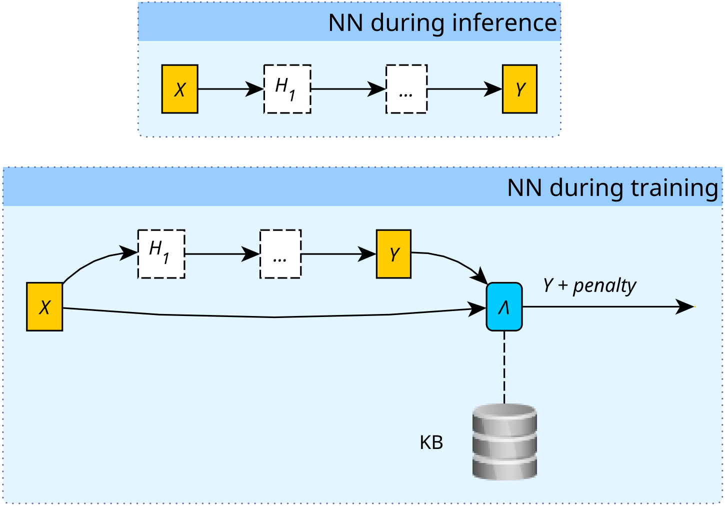
purpose → learning support

target predictor → neural networks

strategy → guided learning

input logic → stratified Datalog with negation

Knowledge Injection via Lambda Layer [Magnini et al., 2022b] II



Knowledge Injection via Lambda Layer[Magnini et al., 2022b] III

Formula	C. interpretation	Formula	C. interpretation
$[\![\neg\phi]\!]$	$\eta(1 - [\![\phi]\!])$	$[\![\phi \leq \psi]\!]$	$\eta([\![\phi]\!] - [\![\psi]\!])$
$[\![\phi \wedge \psi]\!]$	$\eta(max([\![\phi]\!], [\![\psi]\!]))$	$[\![class(\bar{X}, \mathtt{y}_i) \leftarrow \psi]\!]$	$[\![\psi]\!]^*$
$[\![\phi \vee \psi]\!]$	$\eta(min([\![\phi]\!], [\![\psi]\!]))$	$[\![\mathrm{expr}(\bar{X})]\!]$	$\mathrm{expr}([\![\bar{X}]\!])$
$[\![\phi = \psi]\!]$	$\eta(\lvert[\![\phi]\!] - [\![\psi]\!]\rvert)$	$[\![\mathtt{true}]\!]$	0
$[\![\phi \neq \psi]\!]$	$[\![\neg(\phi = \psi)]\!]$	$[\![\mathtt{false}]\!]$	1
$[\![\phi > \psi]\!]$	$\eta(0.5 - [\![\phi]\!] + [\![\psi]\!])$	$[\![X]\!]$	x
$[\![\phi \geq \psi]\!]$	$\eta([\![\psi]\!] - [\![\phi]\!])$	$[\![\mathtt{k}]\!]$	k
$[\![\phi < \psi]\!]$	$\eta(0.5 + [\![\phi]\!] - [\![\psi]\!])$	$[\![p(\bar{X})]\!]^{**}$	$[\![\psi_1 \vee \ldots \vee \psi_k]\!]$

* encodes the penalty for the i^{th} neuron

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

Focus on. . .

Notable Remarks

- knowledge bases should express relations about input–output pairs
- embedding implies extensional representation of knowledge
  - guided learning, and structuring support intensional knowledge
- propositional knowledge implies binarising the I/O spaces

Current Limitations

- support for regression is preliminary
- recursive data structures are not supported
- recursive clauses are not supported
- extensional representation cost storage
  - not always possible
- guided learning works poorly with lacking data

Future research activities

- foundational: address recursion
- practical: address regression
- is SKI possible outside the NN domain?

Next in Line. . .

Focus on. . .

Python Library for LIME I

Key components

LimeTabularExplainer — explainer for predictions on tabular data

- it can be used for both classification and regression tasks

LimeImageExplainer — explainer for predictions on image data

- image classification tasks

LimeTextExplainer — explainer for predictions on text data

- text classification tasks

Python Library for LIME II

Unified API for Explainers

- the explanation for one data sample can be obtained by the `explain_instance` method, it has several parameters

  e.g. `predict_fn`, `num_sample`, `num_features`
- `explain_instance` gives an `Explanation` (or an `ImageExplanation`) object. It contains information about the domain (e.g., features, class, bins) and, of course, about the explanation of the data sample

  e.g. `as_list`, `as_html` to get the explanation as a textual list or an image

Tutorial

Two ways to reproduce the tutorial:

GitHub Repository (long way)

`https://github.com/pikalab-unibo/demo-lime`

DockerHub Images (quick way)

`https://hub.docker.com/r/pikalab/demo-lime/tags`

Focus on. . .

How to set the tutorial up from GitHub I

Enviromental pre-requisites

- Python `3.9.x`
- Git

1. `git clone https://github.com/pikalab-unibo/demo-lime`
2. `cd demo-lime`
3. `pip install -r requirements.txt`
4. `jupyter notebook`

How to set the tutorial up from GitHub II

5. Your browser should automatically open showing the following page:

6. open the `demo-lime.ipynb` notebook
7. listen to the speaker presenting the tutorial =)

Focus on. . .

How to set the tutorial up via Docker I

Enviromental pre-requisites

- Docker

1. $$\texttt{DOCKER\_IMAGE=}\begin{cases}\texttt{pikalab/demo-lime:latest} & \text{on most co} \\ \texttt{pikalab/demo-lime:latest-apple-m1} & \text{on Apple M}\end{cases}$$
2. `docker pull $DOCKER_IMAGE`
   - in case of lacking Internet access:

     `docker image load -i /path/to/local/image/file.tar`
3. `docker run -it -rm -name demo-lime -p 8888:8888 $DOCKER_IMAGE`
4. Some textual output such as the following one should appear:

How to set the tutorial up via Docker II

```
[I 09:51:46.940 NotebookApp] Writing notebook server cookie secret to /root/.local/
    share/jupyter/runtime/notebook_cookie_secret
[I 09:51:47.159 NotebookApp] Serving notebooks from local directory: /notebook
[I 09:51:47.159 NotebookApp] Jupyter Notebook 6.5.2 is running at:
[I 09:51:47.159 NotebookApp] http://cb0a3641caf0:8888/?token=2
    b02d31671c6ad9e9cf8e036eb6962d3592af9cfdd5e60bd
[I 09:51:47.159 NotebookApp]  or http://127.0.0.1:8888/?token=2
    b02d31671c6ad9e9cf8e036eb6962d3592af9cfdd5e60bd
[I 09:51:47.160 NotebookApp] Use Control-C to stop this server and shut down all
    kernels (twice to skip confirmation).
[C 09:51:47.162 NotebookApp]

To access the notebook, open this file in a browser:
    file:///root/.local/share/jupyter/runtime/nbserver-7-open.html
Or copy and paste one of these URLs:
    http://cb0a3641caf0:8888/?token=2
        b02d31671c6ad9e9cf8e036eb6962d3592af9cfdd5e60bd
or http://127.0.0.1:8888/?token=2b02d31671c6ad9e9cf8e036eb6962d3592af9cfdd5e60bd
```

How to set the tutorial up via Docker III

5. Copy-paste into your browser any link of the form:

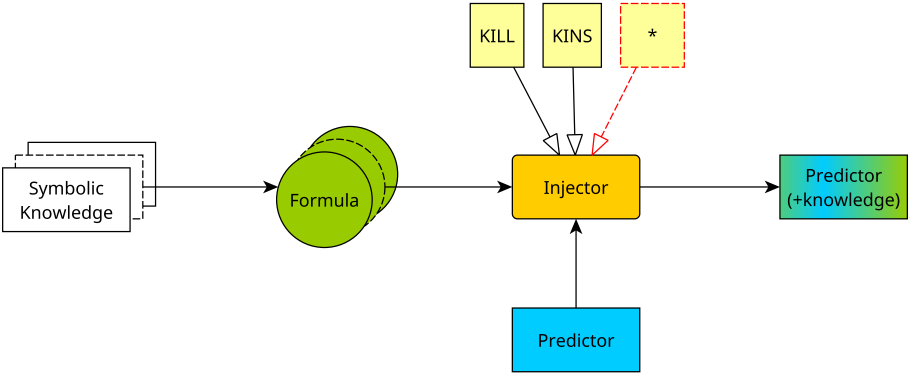
   `http://cb0a3641caf0:8888/?token=`*`TOKEN`*

6. Your browser should now be showing the following page:

7. open the `demo-lime.ipynb` notebook
8. listen to the speaker presenting the tutorial =)

Focus on. . .

Overall Design I



Overall Design II

Key components:

injector: any entity capable of injecting knowledge into a sub-symbolic predictor

- it simply alters/reconfigures the predictor…
- …which should be trained after the injector operates

predictor: the partially-trained classifier/regressor where knowledge should be injected into

- untrained is ok too

formula: formal representation of the symbolic knowledge to be injected

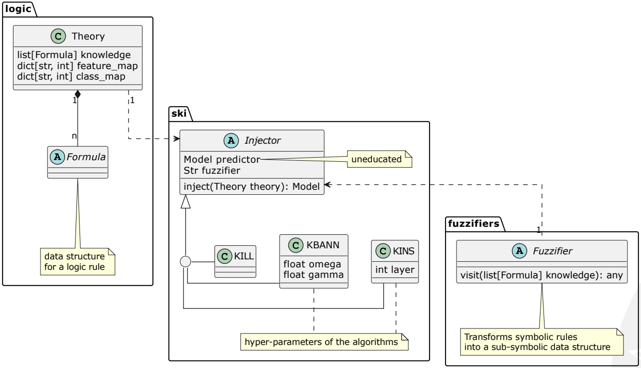
- e.g. in Prolog or FOL syntax

Overall Design III

Unified API for SKI

- 1 interface for `Injector`, several implementations
  e.g. KBANN, KINS, KILL, etc.
- 1 interface for `Formula`, several implementations
  e.g. Datalog, Propositional, etc.
- 1 interface for `Predictor`, currently a TF model
  e.g. different kinds of NN

API Design I



logic
Theory
list[Formula] knowledge
dict[str, int] feature_map
dict[str, int] class_map
1
n
Formula
data structure
for a logic rule
1
ski
Injector
Model predictor
Str fuzzifier
uneducated
inject(Theory theory): Model
KILL
KBANN
float omega
float gamma
KINS
int layer
hyper-parameters of the algorithms
fuzzifiers
1
Fuzzifier
visit(list[Formula] knowledge): any
Transforms symbolic rules
into a sub-symbolic data structure

API Design II

Remarks

- The user only needs to know:
  - the particular injector to exploit (and its parameters)
  - the particular parser to decode logic rules

API Design III

Underlying symbolic AI library (e.g. 2P-Kt[Ciatto et al., 2021]), providing:

`Rule` a semantic, intelligible representation of the function mapping `Predictor`'s inputs into the corresponding outputs, for a particular portion of the input space;

`Theory` an ordered collection of rules.

Tutorial

Two ways to reproduce the tutorial:

GitHub Repository (long way)

https://github.com/psykei/demo-psyki-python

DockerHub Images (quick way)

https://hub.docker.com/r/pikalab/prima-tutorial-2022/tags

Focus on. . .

How to set the tutorial up from GitHub I

Enviromental pre-requisites

- Python `3.9.x`
- JDK $\geq$ `11`
- Git

1. `git clone https://github.com/psykei/demo-psyki-python`
2. `cd demo-psyki-python`
3. `pip install -r requirements.txt`
4. `export PYTHONPATH="$(pwd)"`
5. `jupyter notebook`

How to set the tutorial up from GitHub II

6. Your browser should automatically open showing the following page:

7. open the `*.ipynb` notebooks in the notebook folder
8. listen to the speaker presenting the tutorial =)

Focus on. . .

How to set the tutorial up via Docker I

Enviromental pre-requisites

- Docker

1. $$\texttt{DOCKER\_IMAGE=}\begin{cases}\texttt{pikalab/demo-psyki-python:latest} \\ \text{(on most computers)} \\ \texttt{pikalab/demo-psyki-python:latest-apple-m1} \\ \text{(on Apple M1 computers)}\end{cases}$$
2. `docker pull $DOCKER_IMAGE`
   - in case of lacking Internet access:

     `docker image load -i /path/to/local/image/file.tar`
3. `docker run -it -rm -name demo-psyki-python -p 8888:8888 $DOCKER_IMAGE`

How to set the tutorial up via Docker II

4. Some textual output such as the following one should appear:

```
[I 09:51:46.940 NotebookApp] Writing notebook server cookie secret to /root/.local/
    share/jupyter/runtime/notebook_cookie_secret
[I 09:51:47.159 NotebookApp] Serving notebooks from local directory: /notebook
[I 09:51:47.159 NotebookApp] Jupyter Notebook 6.5.2 is running at:
[I 09:51:47.159 NotebookApp] http://cb0a3641caf0:8888/?token=2
    b02d31671c6ad9e9cf8e036eb6962d3592af9cfdd5e60bd
[I 09:51:47.159 NotebookApp]  or http://127.0.0.1:8888/?token=2
    b02d31671c6ad9e9cf8e036eb6962d3592af9cfdd5e60bd
[I 09:51:47.160 NotebookApp] Use Control-C to stop this server and shut down all
    kernels (twice to skip confirmation).
[C 09:51:47.162 NotebookApp]

To access the notebook, open this file in a browser:
    file:///root/.local/share/jupyter/runtime/nbserver-7-open.html
Or copy and paste one of these URLs:
    http://cb0a3641caf0:8888/?token=2
        b02d31671c6ad9e9cf8e036eb6962d3592af9cfdd5e60bd
or http://127.0.0.1:8888/?token=2b02d31671c6ad9e9cf8e036eb6962d3592af9cfdd5e60bd
```

How to set the tutorial up via Docker III

5. Copy-paste into your browser any link of the form:

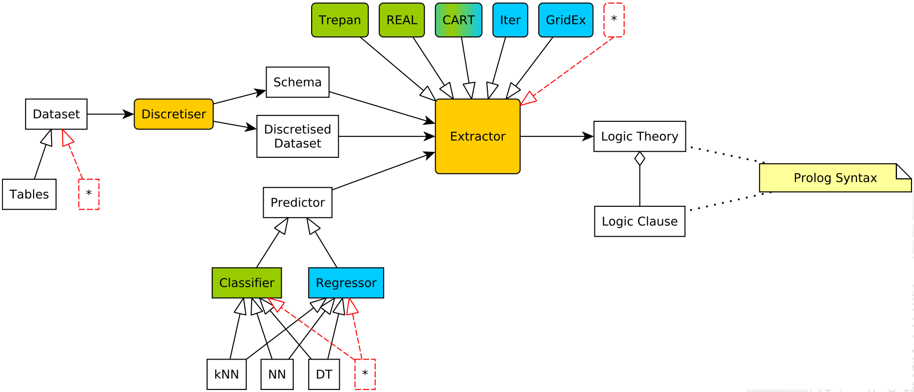
   `http://cb0a3641caf0:8888/?token=TOKEN`

6. Your browser should now be showing the following page:

7. open the `*.ipynb` notebooks
8. listen to the speaker presenting the tutorial =)

Focus on. . .

Overall Design I



Overall Design II

Key components:

extractor: any entity capable of extracting symbolic knowledge out of sub-symbolic predictors

- possibly, in the form of logic knowledge bases
- possibly, leveraging upon the dataset the predictor was trained upon ...
  - possibly, after a discretization step
- ...and its schema

predictor: some trained classifier/regressor from which knowledge should be extracted

discretiser: any component capable to turn continuous datasets into discrete form, following some strategy

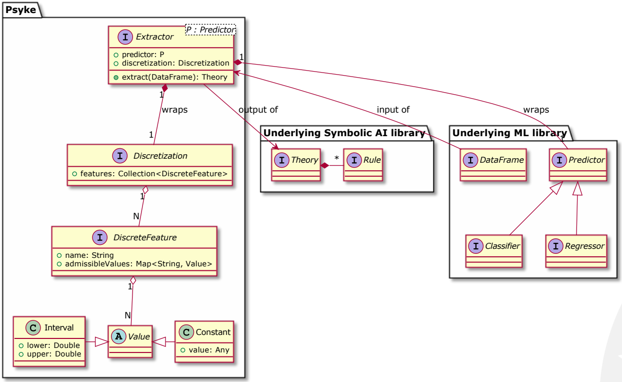
logic theory: outcome of the extraction process

Overall Design III

Unified API for SKE

- 1 interface for `Extractor`, several implementations
  e.g. CART, REAL, GridEx
- 1 interface for `Discretiser`, several implementations
- 1 interface for `Predictor`, several implementations (scikit-learn method convention)
  e.g. NN, kNN, DT

API Design I


Psyke
Extractor
P : Predictor
predictor: P
discretization: Discretization
extract(DataFrame): Theory
wraps
Discretization
features: Collection<DiscreteFeature>
DiscreteFeature
name: String
admissibleValues: Map<String, Value>
Interval
lower: Double
upper: Double
Value
Constant
value: Any
output of
input of
wraps
Underlying Symbolic AI library
Theory
Rule
Underlying ML library
DataFrame
Predictor
Classifier
Regressor

API Design II

General assumptions:

- underlying ML library (e.g. Scikit-Learn[Pedregosa et al., 2011]), providing:
  - `DataFrame` a container of tabular data
  - `Predictor<R>` a computational entity which can be trained (a.k.a. fitted) against a `DataFrame` and used to draw predictions of type `R`;
  - `Classifier<R>` a particular case of predictor where `R` represents a type having a finite amount of admissible values;
  - `Regressor<R>` a particular case of predictor where `R` represents a type having a potentially infinite (possibly continuous) amount of admissible values.

API Design III

- underlying symbolic AI library (e.g. 2P-Kt[Ciatto et al., 2021]), providing:
  - **Rule** a semantic, intelligible representation of the function mapping `Predictor`'s inputs into the corresponding outputs, for a particular portion of the input space;
  - **Theory** an ordered collection of rules.

About the Extracted Knowledge I

Knowledge extracted from **classifiers**

$$
\begin{array}{lcl}
\langle task \rangle(X_1, \ldots, X_n, \mathrm{y}_1) & :- & p_{1,1}(\bar{X}),\ \ldots,\ p_{n,1}(\bar{X}). \\
\langle task \rangle(X_1, \ldots, X_n, \mathrm{y}_2) & :- & p_{1,2}(\bar{X}),\ \ldots,\ p_{n,2}(\bar{X}). \\
 & \vdots & \\
\langle task \rangle(X_1, \ldots, X_n, \mathrm{y}_m) & :- & p_{1,m}(\bar{X}),\ \ldots,\ p_{n,m}(\bar{X}).
\end{array}
$$

About the Extracted Knowledge II

Knowledge extracted from **regressors**

$$
\begin{aligned}
\langle task \rangle(X_1, \ldots, X_n, Y) \quad &\texttt{:-} \quad p_{1,1}(\bar{X}),\ \ldots,\ p_{n,1}(\bar{X}), \\
&\qquad Y \texttt{ is } f_1(\bar{X}). \\
\langle task \rangle(X_1, \ldots, X_n, Y) \quad &\texttt{:-} \quad p_{1,2}(\bar{X}),\ \ldots,\ p_{n,2}(\bar{X}), \\
&\qquad Y \texttt{ is } f_2(\bar{X}). \\
&\vdots \\
\langle task \rangle(X_1, \ldots, X_n, Y) \quad &\texttt{:-} \quad p_{1,m}(\bar{X}),\ \ldots,\ p_{n,m}(\bar{X}), \\
&\qquad Y \texttt{ is } f_m(\bar{X}).
\end{aligned}
$$

About the Extracted Knowledge III

...where:

- *task* is the $(n+1)$-ary relation representing the classification or regression task at hand,
- each X_i is a logic variable named after the i^{th} input attribute of the currently available data set,
- $\bar{X}$ is the n-nuple $X_1, \ldots, X_n$,
- each $p_{i,j}$ is either a n-ary predicate expressing some constraint about one, two or more variables, or the `true` literal—which can be omitted,
- y_i is the output of the i^{th} prediction rule,
- f_j is an n-ary function computing the output value for the regression task in the particular portion of the input space handled by the j^{th} rule, and
- $is/2$ is the well-known Prolog predicate aimed at evaluating functions.

About the Extracted Knowledge IV

Underlying assumptions

1. the input space is partitioned into a finite set of regions
2. each region is assigned with a particular outcome, namely:
    - a class, for classification problems
    - a constant, or a simpler function, for regression problems
3. one rule generated describing for each region and its corresponding outcome

Tutorial

Two ways to reproduce the tutorial:

GitHub Repository (long way)

https://github.com/pikalab-unibo/prima-tutorial-2022

DockerHub Images (quick way)

https://hub.docker.com/r/pikalab/prima-tutorial-2022/tags

Focus on...

How to set the tutorial up from GitHub I

Enviromental pre-requisites

- Python `3.9.x`
- JDK $\geq$ `11`
- Git

1. `git clone https://github.com/pikalab-unibo/prima-tutorial-2022`
2. `cd prima-tutorial-2022`
3. `pip install -r requirements.txt`
4. `jupyter notebook`

How to set the tutorial up from GitHub II

5. Your browser should automatically open showing the following page:

6. open the `psyke-tutorial.ipynb` notebook
7. listen to the speaker presenting the tutorial =)

Focus on. . .

How to set the tutorial up via Docker I

Enviromental pre-requisites

- Docker

1. DOCKER_IMAGE= { `pikalab/prima-tutorial-2022:latest` | `pikalab/prima-tutorial-2022:latest-apple-m1` }
2. `docker pull $DOCKER_IMAGE`
   - in case of lacking Internet access:

     `docker image load -i /path/to/local/image/file.tar`
3. `docker run -it -rm -name prima-tutorial-ske-ski -p 8888:8888 $DOCKER_IMAGE`
4. Some textual output such as the following one should appear:

How to set the tutorial up via Docker II

```
[I 09:51:46.940 NotebookApp] Writing notebook server cookie secret to /root/.local/
    share/jupyter/runtime/notebook_cookie_secret
[I 09:51:47.159 NotebookApp] Serving notebooks from local directory: /notebook
[I 09:51:47.159 NotebookApp] Jupyter Notebook 6.5.2 is running at:
[I 09:51:47.159 NotebookApp] http://cb0a3641caf0:8888/?token=2
    b02d31671c6ad9e9cf8e036eb6962d3592af9cfdd5e60bd
[I 09:51:47.159 NotebookApp]  or http://127.0.0.1:8888/?token=2
    b02d31671c6ad9e9cf8e036eb6962d3592af9cfdd5e60bd
[I 09:51:47.160 NotebookApp] Use Control-C to stop this server and shut down all
    kernels (twice to skip confirmation).
[C 09:51:47.162 NotebookApp]

To access the notebook, open this file in a browser:
    file:///root/.local/share/jupyter/runtime/nbserver-7-open.html
Or copy and paste one of these URLs:
    http://cb0a3641caf0:8888/?token=2
        b02d31671c6ad9e9cf8e036eb6962d3592af9cfdd5e60bd
or http://127.0.0.1:8888/?token=2b02d31671c6ad9e9cf8e036eb6962d3592af9cfdd5e60bd
```

How to set the tutorial up via Docker III

5. Copy-paste into your browser any link of the form:

   `http://cb0a3641caf0:8888/?token=`*`TOKEN`*

6. Your browser should now be showing the following page:

7. open the `psyke-tutorial.ipynb` notebook
8. listen to the speaker presenting the tutorial =)

eXplainable Artificial Intelligence (XAI)
A Gentle Introduction

Matteo Magnini Giovanni Ciatto Andrea Omicini

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

Advanced School in Artificial Intelligence – 17-28 July 2023

References I

[Andrews and Geva, 1995] Andrews, R. and Geva, S. (1995).
Rulex & cebp networks as the basis for a rule refinement system.
In Hallam, J., editor, *Hybrid Problems, Hybrid Solutions*, pages 1–12. IOS Press
.

[Anjomshoae et al., 2019] Anjomshoae, S., Najjar, A., Calvaresi, D., and Främling, K. (2019).
Explainable agents and robots: Results from a systematic literature review.
In Elkind, E., Veloso, M., Agmon, N., and Taylor, M. E., editors, *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, pages 1078–1088. International Foundation for Autonomous Agents and Multiagent Systems
http://dl.acm.org/citation.cfm?id=3331806.

[Augasta and Kathirvalavakumar, 2012] Augasta, M. G. and Kathirvalavakumar, T. (2012).
Reverse engineering the neural networks for rule extraction in classification problems.
Neural Processing Letters, 35(2):131–150
DOI:10.1007/s11063-011-9207-8.

[Bader, 2009] Bader, S. (2009).
Extracting propositional rules from feedforward neural networks by means of binary decision diagrams.
In d'Avila Garcez, A. S. and Hitzler, P., editors, *Proceedings of the Fifth International Workshop on Neural-Symbolic Learning and Reasoning, NeSy 2009, Pasadena, CA, USA, July 11, 2009*, volume 481 of *CEUR Workshop Proceedings*. CEUR-WS.org
http://ceur-ws.org/Vol-481/paper-5.pdf.

References II

[Bader et al., 2007] Bader, S., Hölldobler, S., and Mayer-Eichberger, V. (2007).
Extracting propositional rules from feed-forward neural networks – A new decompositional approach.
In d'Avila Garcez, A. S., Hitzler, P., and Tamburrini, G., editors, *Proceedings of the 3rd International Workshop on Neural-Symbolic Learning and Reasoning, NeSy'07, held at IJCAI-07, Hyderabad, India, January 8, 2007*, volume 230 of *CEUR Workshop Proceedings*. CEUR-WS.org
http://ceur-ws.org/Vol-230/04-bader.pdf.

[Baldi and Sadowski, 2016] Baldi, P. and Sadowski, P. J. (2016).
A theory of local learning, the learning channel, and the optimality of backpropagation.
Neural Networks, 83:51–74
DOI:10.1016/j.neunet.2016.07.006.

[Barakat and Bradley, 2007] Barakat, N. H. and Bradley, A. P. (2007).
Rule extraction from support vector machines: A sequential covering approach.
IEEE Transactions on Knowledge and Data Engineering, 19(6):729–741
DOI:10.1109/TKDE.2007.190610.

[Barakat and Diederich, 2008] Barakat, N. H. and Diederich, J. (2008).
Eclectic rule-extraction from support vector machines.
International Journal of Computer and Information Engineering, 2(5):1672–1675
DOI:10.5281/zenodo.1055511.

[Benítez et al., 1997] Benítez, J. M., Castro, J. L., and Requena, I. (1997).
Are artificial neural networks black boxes?
IEEE Transactions on Neural Networks, 8(5):1156–1164
DOI:10.1109/72.623216.

References III

[Berenji, 1991] Berenji, H. R. (1991).
Refinement of approximate reasoning-based controllers by reinforcement learning.
In Birnbaum, L. and Collins, G., editors, *Proceedings of the Eighth International Workshop (ML91), Northwestern University, Evanston, Illinois, USA*, pages 475–479. Morgan Kaufmann
DOI:10.1016/b978-1-55860-200-7.50097-0.

[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.

[Boz, 2002] Boz, O. (2002).
Converting a trained neural network to a decision tree DecText - decision tree extractor.
In Wani, M. A., Arabnia, H. R., Cios, K. J., Hafeez, K., and Kendall, G., editors, *Proceedings of the 2002 International Conference on Machine Learning and Applications - ICMLA 2002, June 24-27, 2002, Las Vegas, Nevada, USA*, pages 110–116. CSREA Press
.

[Brachman and Levesque, 2004] Brachman, R. J. and Levesque, H. J. (2004).
The tradeoff between expressiveness and tractability.
In Brachman, R. J. and Levesque, H. J., editors, *Knowledge Representation and Reasoning*, The Morgan Kaufmann Series in Artificial Intelligence, pages 327–348. Morgan Kaufmann, San Francisco
DOI:https://doi.org/10.1016/B978-155860932-7/50101-1.

References IV

[Breiman et al., 1984] Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984).
Classification and Regression Trees.
CRC Press
.

[Brooks, 1991a] Brooks, R. A. (1991a).
Intelligence without reason.
In Mylopoulos, J. and Reiter, R., editors, *12th International Joint Conference on Artificial Intelligence (IJCAI 1991)*, volume 1, pages 569–595, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
http://dl.acm.org/citation.cfm?id=1631258.

[Brooks, 1991b] Brooks, R. A. (1991b).
Intelligence without representation.
Artificial Intelligence, 47:139–159
DOI:10.1016/0004-3702(91)90053-M.

[Browne et al., 2004] Browne, A., Hudson, B. D., Whitley, D. C., Ford, M. G., and Picton, P. (2004).
Biological data mining with neural networks: implementation and application of a flexible decision tree extraction algorithm to genomic problem domains.
Neurocomputing, 57:275–293
DOI:10.1016/j.neucom.2003.10.007.

[Brunk and Pazzani, 1991] Brunk, C. and Pazzani, M. J. (1991).
An investigation of noise-tolerant relational concept learning algorithms.
In Birnbaum, L. and Collins, G., editors, *Proceedings of the Eighth International Workshop (ML91), Northwestern University, Evanston, Illinois, USA*, pages 389–393. Morgan Kaufmann
DOI:10.1016/b978-1-55860-200-7.50080-5.

References V

[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.

[Castillo et al., 2001] Castillo, L. A., González Muñoz, A., and Pérez, R. (2001).
Including a simplicity criterion in the selection of the best rule in a genetic fuzzy learning algorithm.
Fuzzy Sets Syst., 120(2):309–321
DOI:10.1016/S0165-0114(99)00095-0.

[Chan and Chan, 2017] Chan, V. and Chan, C. W. (2017).
Towards developing the piece-wise linear neural network algorithm for rule extraction.
International Journal of Cognitive Informatics and Natural Intelligence, 11(2):57–73
DOI:10.4018/IJCINI.2017040104.

[Chan and Chan, 2020] Chan, V. K. and Chan, C. W. (2020).
Towards explicit representation of an artificial neural network model: Comparison of two artificial neural network rule extraction approaches.
Petroleum, 6(4):329–339.
SI: Artificial Intelligence (AI), Knowledge-based Systems (KBS), and Machine Learning (ML)
DOI:https://doi.org/10.1016/j.petlm.2019.11.005.

[Chaves et al., 2005] Chaves, A. d. C. F., Vellasco, M. M. B. R., and Tanscheit, R. (2005).
Fuzzy rule extraction from support vector machines.
In Nedjah, N., de Macedo Mourelle, L., Abraham, A., and Köppen, M., editors, *5th International Conference on Hybrid Intelligent Systems (HIS 2005), 6-9 November 2005, Rio de Janeiro, Brazil*, pages 335–340. IEEE Computer Society
DOI:10.1109/ICHIS.2005.51.

References VI

[Chen, 2004] Chen, F. (2004).
Learning accurate and understandable rules from SVM classifiers.
Master's thesis, Simon Fraser University, Vancouver, Canada
.

[Chen et al., 2007] Chen, Z., Li, J., and Wei, L. (2007).
A multiple kernel support vector machine scheme for feature selection and rule extraction from gene expression data of cancer tissue.
Artif. Intell. Medicine, 41(2):161–175
DOI:10.1016/j.artmed.2007.07.008.

[Ciatto et al., 2021] Ciatto, G., Calegari, R., and Omicini, A. (2021).
2P-Kt: A logic-based ecosystem for symbolic AI.
SoftwareX, 16:100817:1–7
DOI:10.1016/j.softx.2021.100817.

[Ciatto et al., 2019] Ciatto, G., Calegari, R., Omicini, A., and Calvaresi, D. (2019).
Towards XMAS: eXplainability through Multi-Agent Systems.
In Savaglio, C., Fortino, G., Ciatto, G., and Omicini, A., editors, *AI&IoT 2019 – Artificial Intelligence and Internet of Things 2019*, volume 2502 of *CEUR Workshop Proceedings*, pages 40–53. CEUR WS
http://ceur-ws.org/Vol-2502/paper3.pdf.

[Ciatto et al., 2020] Ciatto, G., Schumacher, M. I., Omicini, A., and Calvaresi, D. (2020).
Agent-based explanations in AI: Towards an abstract framework.
In Calvaresi, D., Najjar, A., Winikoff, M., and Främling, K., editors, *Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, volume 12175 of *LNCS*, pages 3–20. Springer, Cham
DOI:10.1007/978-3-030-51924-7_1.

References VII

[Clark and Niblett, 1989] Clark, P. and Niblett, T. (1989).
The CN2 induction algorithm.
Mach. Learn., 3:261–283
DOI:10.1007/BF00116835.

[Cohen, 1993] Cohen, W. W. (1993).
Efficient pruning methods for separate-and-conquer rule learning systems.
In Bajcsy, R., editor, *Proceedings of the 13th International Joint Conference on Artificial Intelligence. Chambéry, France, August 28 - September 3, 1993*, pages 988–994. Morgan Kaufmann
.

[Cohen, 1995] Cohen, W. W. (1995).
Fast effective rule induction.
In Prieditis, A. and Russell, S. J., editors, *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, 1995*, pages 115–123. Morgan Kaufmann
DOI:10.1016/b978-1-55860-377-6.50023-2.

[Craven and Shavlik, 1994] Craven, M. W. and Shavlik, J. W. (1994).
Using sampling and queries to extract rules from trained neural networks.
In *Machine Learning Proceedings 1994*, pages 37–45. Elsevier
DOI:10.1016/B978-1-55860-335-6.50013-1.

[Craven and Shavlik, 1996] Craven, M. W. and Shavlik, J. W. (1996).
Extracting tree-structured representations of trained networks.
In Touretzky, D. S., Mozer, M. C., and Hasselmo, M. E., editors, *Advances in Neural Information Processing Systems 8. Proceedings of the 1995 Conference*, pages 24–30. The MIT Press
http://papers.nips.cc/paper/1152-extracting-tree-structured-representations-of-trained-networks.pdf.

References VIII

[Crawford, 2016] Crawford, K. (2016).
Artificial intelligence's white guy problem.
The New York Times, 25
.

[D'Amore, 2005] D'Amore, B. (2005).
Noetica e semiotica nell'apprendimento della matematica.
In Laura, A. R., Eleonora, F., Antonella, M., and Rosa, P., editors, *Insegnare la matematica nella scuola di tutti e di ciascuno*, Milano, Italy. Ghisetti & Corvi Editore
http://www.dm.unibo.it/rsddm/it/articoli/damore/676noeticaesemioticaBari.pdf.

[De Rijk, 2002] De Rijk, L. M. (2002).
Aristotle: Semantics and Ontology. Volume I: General Introduction. The Works on Logic, volume 91 of *Philosophia Antiqua*.
Brill Academic Publishers
https://brill.com/view/title/7491.

[Dean et al., 2012] Dean, L. G., Kendal, R. L., Schapiro, S. J., Thierry, B., and Laland, K. N. (2012).
Identification of the social and cognitive processes underlying human cumulative culture.
Science, 335(6072):1114–1118
DOI:10.1126/science.1213969.

[Etchells and G., 2006] Etchells, T. A. and G., L. P. J. (2006).
Orthogonal search-based rule extraction (OSRE) for trained neural networks: a practical and efficient approach.
IEEE Transactions on Neural Networks, 17(2):374–384
DOI:10.1109/TNN.2005.863472.

References IX

[Fu, 1994] Fu, L. (1994).
Rule generation from neural networks.
IEEE Transactions on Systems, Man, and Cybernetics, 24(8):1114–1124
DOI:10.1109/21.299696.

[Fu et al., 2004] Fu, X., Ong, C., Keerthi, S., Hung, G. G., and Goh, L. (2004).
Extracting the knowledge embedded in support vector machines.
In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, volume 1, pages 291–296
DOI:10.1109/IJCNN.2004.1379916.

[Fung et al., 2005] Fung, G., Sandilya, S., and Rao, R. B. (2005).
Rule extraction from linear support vector machines.
In Grossman, R., Bayardo, R. J., and Bennett, K. P., editors, *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August 21-24, 2005*, pages 32–40. ACM
DOI:10.1145/1081870.1081878.

[Fürnkranz and Widmer, 1994] Fürnkranz, J. and Widmer, G. (1994).
Incremental reduced error pruning.
In Cohen, W. W. and Hirsh, H., editors, *Machine Learning, Proceedings of the Eleventh International Conference, Rutgers University, New Brunswick, NJ, USA, July 10-13, 1994*, pages 70–77. Morgan Kaufmann
DOI:10.1016/b978-1-55860-335-6.50017-9.

[Goodman and Flaxman, 2017] Goodman, B. and Flaxman, S. (2017).
European Union regulations on algorithmic decision-making and a “right to explanation”.
AI Magazine, 38(3):50–57
DOI:10.1609/aimag.v38i3.2741.

References X

[Gozalo-Brizuela and Garrido-Merchan, 2023] Gozalo-Brizuela, R. and Garrido-Merchan, E. C. (2023).
ChatGPT is not all you need. a state of the art review of large generative AI models
DOI:10.48550/ARXIV.2301.04655.

[Guidotti et al., 2018] Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2018).
A survey of methods for explaining black box models.
ACM Computing Surveys, 51(5):1–42
DOI:10.1145/3236009.

[Gunning, 2016a] Gunning, D. (2016a).
Explainable artificial intelligence.
https://www.darpa.mil/attachments/XAIIndustryDay_Final.pptx
.

[Gunning, 2016b] Gunning, D. (2016b).
Explainable artificial intelligence (XAI).
Funding Program DARPA-BAA-16-53, Defense Advanced Research Projects Agency (DARPA)
http://www.darpa.mil/program/explainable-artificial-intelligence.

[Halgamuge and Glesner, 1994] Halgamuge, S. K. and Glesner, M. (1994).
Neural networks in designing fuzzy systems for real world applications.
Fuzzy Sets and Systems, 65(1):1–12
DOI:https://doi.org/10.1016/0165-0114(94)90242-9.

References XI

[Hayashi, 1990] Hayashi, Y. (1990).
A neural expert system with automated extraction of fuzzy if-then rules.
In Lippmann, R., Moody, J. E., and Touretzky, D. S., editors, *Advances in Neural Information Processing Systems 3, [NIPS Conference, Denver, Colorado, USA, November 26-29, 1990]*, pages 578–584. Morgan Kaufmann
http://papers.nips.cc/paper/355-a-neural-expert-system-with-automated-extraction-of-fuzzy-if-then-rules.

[He et al., 2006] He, J., Hu, H.-J., Harrison, R., Tai, P., and Pan, Y. (2006).
Rule generation for protein secondary structure prediction with support vector machines and decision tree.
IEEE Transactions on NanoBioscience, 5(1):46–53
DOI:10.1109/TNB.2005.864021.

[Hoffmann and Magazzeni, 2019] Hoffmann, J. and Magazzeni, D. (2019).
Explainable AI planning (XAIP): overview and the case of contrastive explanation (extended abstract).
In Krötzsch, M. and Stepanova, D., editors, *Reasoning Web. Explainable Artificial Intelligence - 15th International Summer School 2019, Bolzano, Italy, September 20-24, 2019, Tutorial Lectures*, volume 11810 of *Lecture Notes in Computer Science*, pages 277–282. Springer
DOI:10.1007/978-3-030-31423-1_9.

[Hong and Chen, 1999] Hong, T. and Chen, J. (1999).
Finding relevant attributes and membership functions.
Fuzzy Sets Syst., 103(3):389–404
DOI:10.1016/S0165-0114(97)00187-5.

[Hong and Chen, 2000] Hong, T. and Chen, J. (2000).
Processing individual fuzzy attributes for fuzzy rule induction.
Fuzzy Sets Syst., 112(1):127–140
DOI:10.1016/S0165-0114(98)00179-1.

References XII

[Hong and Lee, 1996] Hong, T. and Lee, C. (1996).
Induction of fuzzy rules and membership functions from training examples.
Fuzzy Sets Syst., 84(1):33–47
DOI:10.1016/0165-0114(95)00305-3.

[Horikawa et al., 1992] Horikawa, S., Furuhashi, T., and Uchikawa, Y. (1992).
On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm.
IEEE Transactions on Neural Networks, 3(5):801–806
DOI:10.1109/72.159069.

[Huysmans et al., 2006] Huysmans, J., Baesens, B., and Vanthienen, J. (2006).
ITER: An algorithm for predictive regression rule extraction.
In *Data Warehousing and Knowledge Discovery (DaWaK 2006)*, pages 270–279. Springer
DOI:10.1007/11823728_26.

[Ishibuchi et al., 1997] Ishibuchi, H., Nii, M., and Murata, T. (1997).
Linguistic rule extraction from neural networks and genetic-algorithm-based rule selection.
In *Proceedings of International Conference on Neural Networks (ICNN'97), Houston, TX, USA, June 9-12, 1997*, pages 2390–2395. IEEE
DOI:10.1109/ICNN.1997.614441.

[Kim and Lee, 2000] Kim, D. and Lee, J. (2000).
Handling continuous-valued attributes in decision tree with neural network modeling.
In López de Mántaras, R. and Plaza, E., editors, *Machine Learning: ECML 2000*, pages 211–219, Berlin, Heidelberg. Springer Berlin Heidelberg
.

References XIII

[Konig et al., 2008] Konig, R., Johansson, U., and Niklasson, L. (2008).
G-REX: A versatile framework for evolutionary data mining.
In *2008 IEEE International Conference on Data Mining Workshops (ICDM 2008 Workshops)*, pages 971–974
DOI:10.1109/ICDMW.2008.117.

[Krishnan et al., 1999a] Krishnan, R., Sivakumar, G., and Bhattacharya, P. (1999a).
Extracting decision trees from trained neural networks.
Pattern Recognition, 32(12):1999–2009
DOI:10.1016/S0031-3203(98)00181-2.

[Krishnan et al., 1999b] Krishnan, R., Sivakumar, G., and Bhattacharya, P. (1999b).
A search technique for rule extraction from trained neural networks.
Pattern Recognition Letters, 20(3):273–280
DOI:10.1016/S0167-8655(98)00145-7.

[Lehmann et al., 2010] Lehmann, J., Bader, S., and Hitzler, P. (2010).
Extracting reduced logic programs from artificial neural networks.
Applied Intelligence, 32(3):249–266
DOI:10.1007/s10489-008-0142-y.

[Levesque and Brachman, 1987] Levesque, H. J. and Brachman, R. J. (1987).
Expressiveness and tractability in knowledge representation and reasoning.
Comput. Intell., 3:78–93
DOI:10.1111/j.1467-8640.1987.tb00176.x.

References XIV

[Lipton, 2018] Lipton, Z. C. (2018).
The mythos of model interpretability.
Queue, 16(3):31–57
DOI:10.1145/3236386.3241340.

[Liu et al., 2002] Liu, B., Abbass, H. A., and McKay, R. I. (2002).
Density-based heuristic for rule discovery with ant-miner.
In *The 6th Australia-Japan joint workshop on intelligent and evolutionary system*, volume 184
.

[Liu et al., 2004] Liu, B., Abbass, H. A., and McKay, R. I. (2004).
Classification rule discovery with ant colony optimization.
IEEE Intell. Informatics Bull., 3(1):31–35
http://www.comp.hkbu.edu.hk/%7Ecib/2004/Feb/2004/Feb/cib_vol3no1_article4.pdf.

[Lundberg and Lee, 2017] Lundberg, S. M. and Lee, S.-I. (2017).
A unified approach to interpreting model predictions.
In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

[Magnini et al., 2022a] Magnini, M., Ciatto, G., and Omicini, A. (2022a).
KINS: Knowledge injection via network structuring.
In Calegari, R., Ciatto, G., and Omicini, A., editors, *CILC 2022 – Italian Conference on Computational Logic*, volume 3204 of *CEUR Workshop Proceedings*, pages 254–267. CEUR-WS
http://ceur-ws.org/Vol-3204/paper_25.pdf.

References XV

[Magnini et al., 2022b] Magnini, M., Ciatto, G., and Omicini, A. (2022b).
A view to a KILL: Knowledge injection via lambda layer.
In Ferrando, A. and Mascardi, V., editors, *WOA 2022 – 23rd Workshop "From Objects to Agents"*, volume 3261 of *CEUR Workshop Proceedings*, pages 61–76. Sun SITE Central Europe, RWTH Aachen University
http://ceur-ws.org/Vol-3261/paper5.pdf.

[Markowska-Kaczmar and Chumieja, 2004] Markowska-Kaczmar, U. and Chumieja, M. (2004).
Discovering the mysteries of neural networks.
Int. J. Hybrid Intell. Syst., 1(3-4):153–163
http://content.iospress.com/articles/international-journal-of-hybrid-intelligent-systems/his016.

[Markowska-Kaczmar and Trelak, 2003] Markowska-Kaczmar, U. and Trelak, W. (2003).
Extraction of fuzzy rules from trained neural network using evolutionary algorithm.
In *ESANN 2003, 11th European Symposium on Artificial Neural Networks, Bruges, Belgium, April 23-25, 2003, Proceedings*, pages 149–154
https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2003-9.pdf.

[Martens et al., 2009] Martens, D., Baesens, B., and Van Gestel, T. (2009).
Decompositional rule extraction from support vector machines by active learning.
IEEE Transactions on Knowledge and Data Engineering, 21(2):178–191
DOI:10.1109/TKDE.2008.131.

[Martens et al., 2007] Martens, D., De Backer, M., Haesen, R., Vanthienen, J., Snoeck, M., and Baesens, B. (2007).
Classification with ant colony optimization.
IEEE Transactions on Evolutionary Computation, 11(5):651–665
DOI:10.1109/TEVC.2006.890229.

References XVI

[Masuoka et al., 1990] Masuoka, R., Watanabe, N., Kawamura, A., Owada, Y., and Asakawa, K. (1990).
Neurofuzzy systems – Fuzzy inference using a structured neural network.
In *Proceedings of International Conference on Fuzzy Logic and Neural Networks, Iizuka Japan, July, 1990*, pages 173–177
.

[Matthews and Jagielska, 1995] Matthews, C. and Jagielska, I. (1995).
Fuzzy rule extraction from a trained multilayer neural network.
In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 2, pages 744–748 vol.2
DOI:10.1109/ICNN.1995.487510.

[McCarthy, 1981] McCarthy, J. (1981).
History of LISP.
In Wexelblat, R. L., editor, *History of Programming Languages I*, pages 173–185. ACM, New York, NY, USA
DOI:10.1145/800025.1198360.

[Milani et al., 2022] Milani, S., Topin, N., Veloso, M., and Fang, F. (2022).
A survey of explainable reinforcement learning.
CoRR, abs/2202.08434
https://arxiv.org/abs/2202.08434.

[Mitra, 1994] Mitra, S. (1994).
Fuzzy mlp based expert system for medical diagnosis.
Fuzzy Sets and Systems, 65(2):285–296.
Fuzzy Methods for Computer Vision and Pattern Recognition
DOI:https://doi.org/10.1016/0165-0114(94)90025-6.

References XVII

[Murphy, 2022] Murphy, K. P. (2022).
Probabilistic Machine Learning: An Introduction.
MIT Press
https://mitpress.mit.edu/9780262046824/.

[Murphy and Pazzani, 1991] Murphy, P. M. and Pazzani, M. J. (1991).
Id2-of-3: Constructive induction of m-of-n concepts for discriminators in decision trees.
In *Machine Learning Proceedings 1991*, pages 183–187. Elsevier
.

[Nauck and Kruse, 1997] Nauck, D. D. and Kruse, R. (1997).
A neuro-fuzzy method to learn fuzzy classification rules from data.
Fuzzy Sets Syst., 89(3):277–288
DOI:10.1016/S0165-0114(97)00009-2.

[Nauck and Kruse, 1999] Nauck, D. D. and Kruse, R. (1999).
Neuro-fuzzy systems for function approximation.
Fuzzy Sets Syst., 101(2):261–271
DOI:10.1016/S0165-0114(98)00169-9.

[Newell and Simon, 1956] Newell, A. and Simon, H. (1956).
The logic theory machine–a complex information processing system.
IRE Transactions on Information Theory, 2(3):61–79
DOI:10.1109/TIT.1956.1056797.

References XVIII

[Núñez et al., 2008] Núñez, H., Angulo, C., and Català, A. (2008).
Rule extraction based on support and prototype vectors.
In Diederich, J., editor, *Rule Extraction from Support Vector Machines*, volume 80 of *Studies in Computational Intelligence*, pages 109–134. Springer
DOI:10.1007/978-3-540-75390-2_5.

[Odajima et al., 2008] Odajima, K., Hayashi, Y., Tianxia, G., and Setiono, R. (2008).
Greedy rule generation from discrete data and its use in neural network rule extraction.
Neural Networks, 21(7):1020–1028
DOI:10.1016/j.neunet.2008.01.003.

[Parliament and Council, 2016] Parliament, E. and Council, E. (2016).
Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec.
http://data.europa.eu/eli/reg/2016/679/oj.
Online; accessed on October 11, 2019
.

[Parpinelli et al., 2001] Parpinelli, R. S., Lopes, H. S., and Freitas, A. A. (2001).
An ant colony based system for data mining: applications to medical data.
In *Proceedings of the genetic and evolutionary computation conference (GECCO-2001)*, pages 791–797. Citeseer
.

[Pascal, 1669] Pascal, B. (1669).
Pensées.
Guillaume Desprez, Paris, France
.

References XIX

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011).
Scikit-learn: Machine learning in Python.
Journal of Machine Learning Research (JMLR), 12:2825–2830
https://dl.acm.org/doi/10.5555/1953048.2078195.

[Pop et al., 1994] Pop, E., Hayward, R., and Diederich, J. (1994).
RULENEG: Extracting rules from a trained ANN by stepwise negation.
Technical report, Neurocomputing Research Centre, Queensland University of Technology
.

[Popper, 2002] Popper, K. R. (2002).
The Logic of Scientific Discovery.
Routledge, London, UK, 2nd edition.
1st English Edition: 1959
DOI:10.4324/9780203994627.

[Quinlan, 1986] Quinlan, J. R. (1986).
Induction of decision trees.
Mach. Learn., 1(1):81–106
DOI:10.1023/A:1022643204877.

[Quinlan, 1993] Quinlan, J. R. (1993).
C4.5: Programming for machine learning.
Morgan Kauffmann
https://dl.acm.org/doi/10.5555/152181.

References XX

[Rabuñal et al., 2004] Rabuñal, J. R., Dorado, J., Pazos, A., Pereira, J., and Rivero, D. (2004).
A new approach to the extraction of ANN rules and to their generalization capacity through GP.
Neural Computation, 16(7):1483–1523
DOI:10.1162/089976604323057461.

[Ribeiro et al., 2016] Ribeiro, M. T., Singh, S., and Guestrin, C. (2016).
"why should I trust you?": Explaining the predictions of any classifier.
In Krishnapuram, B., Shah, M., Smola, A. J., Aggarwal, C. C., Shen, D., and Rastogi, R., editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144. ACM
DOI:10.1145/2939672.2939778.

[Saad and Wunsch II, 2007] Saad, E. W. and Wunsch II, D. C. (2007).
Neural network explanation using inversion.
Neural Networks, 20(1):78–93
DOI:10.1016/j.neunet.2006.07.005.

[Sabbatini and Calegari, 2022] Sabbatini, F. and Calegari, R. (2022).
Clustering-based approaches for symbolic knowledge extraction.
In *XLoKR 2022 - Third Workshop on Explainable Logic-Based Knowledge Representation*, Haifa, Israel
https://arxiv.org/abs/2211.00234.

[Sabbatini et al., 2021] Sabbatini, F., Ciatto, G., and Omicini, A. (2021).
GridEx: An algorithm for knowledge extraction from black-box regressors.
In Calvaresi, D., Najjar, A., Winikoff, M., and Främling, K., editors, *Explainable and Transparent AI and Multi-Agent Systems. Third International Workshop, EXTRAAMAS 2021, Virtual Event, May 3–7, 2021, Revised Selected Papers*, volume 12688 of *LNCS*, pages 18–38. Springer Nature, Basel, Switzerland
DOI:10.1007/978-3-030-82017-6_2.

References XXI

[Saito and Nakano, 1988] Saito, K. and Nakano, R. (1988).
Medical diagnostic expert system based on PDP model.
In *Proceedings of International Conference on Neural Networks (ICNN'88), San Diego, CA, USA, July 24-27, 1988*, pages 255–262. IEEE
DOI:10.1109/ICNN.1988.23855.

[Saito and Nakano, 1997] Saito, K. and Nakano, R. (1997).
Law discovery using neural networks.
In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, IJCAI 97, Nagoya, Japan, August 23-29, 1997, 2 Volumes*, pages 1078–1083. Morgan Kaufmann
http://ijcai.org/Proceedings/97-2/Papers/042.pdf.

[Saito and Nakano, 2002] Saito, K. and Nakano, R. (2002).
Extracting regression rules from neural networks.
Neural Networks, 15(10):1279–1288
DOI:10.1016/S0893-6080(02)00089-8.

[Sato and Tsukimoto, 2001] Sato, M. and Tsukimoto, H. (2001).
Rule extraction from neural networks via decision tree induction.
In *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*, volume 3, pages 1870–1875. IEEE
DOI:10.1109/IJCNN.2001.938448.

[Schetinin et al., 2007] Schetinin, V., Fieldsend, J. E., Partridge, D., Coats, T. J., Krzanowski, W. J., Everson, R. M., Bailey, T. C., and Hernandez, A. (2007).
Confident interpretation of bayesian decision tree ensembles for clinical applications.
IEEE Trans. Inf. Technol. Biomed., 11(3):312–319
DOI:10.1109/TITB.2006.880553.

References XXII

[Schmitz et al., 1999] Schmitz, G. P. J., Aldrich, C., and Gouws, F. S. (1999).
ANN-DT: an algorithm for extraction of decision trees from artificial neural networks.
IEEE Transactions on Neural Networks, 10(6):1392–1401
DOI:10.1109/72.809084.

[Selbst and Powles, 2017] Selbst, A. D. and Powles, J. (2017).
Meaningful information and the right to explanation.
International Data Privacy Law, 7(4):233–242
DOI:10.1093/idpl/ipx022.

[Sestito and Dillon, 1994] Sestito, S. and Dillon, T. S. (1994).
Automated knowledge acquisition.
Prentice Hall International series in computer science and engineering. Prentice Hall
.

[Sethi et al., 2012] Sethi, K. K., Mishra, D. K., and Mishra, B. (2012).
KDRuleEx: A novel approach for enhancing user comprehensibility using rule extraction.
In *2012 Third International Conference on Intelligent Systems Modelling and Simulation*, pages 55–60
DOI:10.1109/ISMS.2012.116.

[Setiono, 1997] Setiono, R. (1997).
Extracting rules from neural networks by pruning and hidden-unit splitting.
Neural Computation, 9(1):205–225
DOI:10.1162/neco.1997.9.1.205.

References XXIII

[Setiono, 2000] Setiono, R. (2000).
Extracting M-of-N rules from trained neural networks.
IEEE Transactions on Neural Networks, 11(2):512–519
DOI:10.1109/72.839020.

[Setiono et al., 2008] Setiono, R., Baesens, B., and Mues, C. (2008).
Recursive neural network rule extraction for data with mixed attributes.
IEEE Transactions on Neural Networks, 19(2):299–307
DOI:10.1109/TNN.2007.908641.

[Setiono and Leow, 2000] Setiono, R. and Leow, W. K. (2000).
FERNN: An algorithm for fast extraction of rules from neural networks.
Applied Intelligence, 12(1-2):15–25
DOI:10.1023/A:1008307919726.

[Setiono et al., 2002] Setiono, R., Leow, W. K., and Zurada, J. M. (2002).
Extraction of rules from artificial neural networks for nonlinear regression.
IEEE Transactions on Neural Networks, 13(3):564–577
DOI:10.1109/TNN.2002.1000125.

[Setiono and Liu, 1996] Setiono, R. and Liu, H. (1996).
Symbolic representation of neural networks.
Computer, 29(3):71–77
DOI:10.1109/2.485895.

References XXIV

[Setiono and Liu, 1997] Setiono, R. and Liu, H. (1997).
Neurolinear: A system for extracting oblique decision rules from neural networks.
In van Someren, M. and Widmer, G., editors, *Machine Learning: ECML-97, 9th European Conference on Machine Learning, Prague, Czech Republic, April 23-25, 1997, Proceedings*, volume 1224 of *Lecture Notes in Computer Science*, pages 221–233. Springer
DOI:10.1007/3-540-62858-4_87.

[Setiono and Thong, 2004] Setiono, R. and Thong, J. Y. L. (2004).
An approach to generate rules from neural networks for regression problems.
European Journal of Operational Research, 155(1):239–250
DOI:10.1016/S0377-2217(02)00792-0.

[Silver et al., 2016] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016).
Mastering the game of Go with deep neural networks and tree search.
Nature, 529:484–489
DOI:10.1038/nature16961.

[Skinner, 1985] Skinner, B. F. (1985).
Cognitive science and behaviourism.
British Journal of Psychology, 76(3):291–301
DOI:10.1111/j.2044-8295.1985.tb01953.x.

[Taha and Ghosh, 1999] Taha, I. A. and Ghosh, J. (1999).
Symbolic interpretation of artificial neural networks.
IEEE Transactions on Knowledge and Data Engineering, 11(3):448–463
DOI:10.1109/69.774103.

References XXV

[Thrun et al., 2006] Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L., Koelen, C., Markey, C., Rummel, C., Niekerk, J. v., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., and Mahoney, P. (2006).
Stanley: The robot that won the DARPA Grand Challenge.
Journal of Field Robotics, 23(9):661–692
DOI:10.1002/rob.20147.

[Thrun, 1993] Thrun, S. B. (1993).
Extracting provably correct rules from artificial neural networks.
Technical report, University of Bonn
.

[Tickle et al., 1996] Tickle, A. B., Orlowski, M., and Diederich, J. (1996).
DEDEC: A methodology for extracting rules from trained artificial neural networks.
In Andrews, R. and Diederich, J., editors, *Rules and Networks: Proceedings of the Rule Extraction from Trained Artificial Neural Networks Workshop*, pages 90–102. Neurocomputing Research Centre, Queensland University of Technology
.

[Torres and Rocco, 2005] Torres, D. E. D. and Rocco, C. M. S. (2005).
Extracting trees from trained SVM models using a TREPAN based approach.
In Nedjah, N., de Macedo Mourelle, L., Abraham, A., and Köppen, M., editors, *5th International Conference on Hybrid Intelligent Systems (HIS 2005), 6-9 November 2005, Rio de Janeiro, Brazil*, pages 353–360. IEEE Computer Society
DOI:10.1109/ICHIS.2005.41.

References XXVI

[Towell and Shavlik, 1991] Towell, G. G. and Shavlik, J. W. (1991).
Interpretation of artificial neural networks: Mapping knowledge-based neural networks into rules.
In Moody, J. E., Hanson, S. J., and Lippmann, R., editors, *Advances in Neural Information Processing Systems 4, [NIPS Conference, Denver, Colorado, USA, December 2-5, 1991]*, pages 977–984. Morgan Kaufmann
http://papers.nips.cc/paper/546-interpretation-of-artificial-neural-networks-mapping-knowledge-based-neural-networks-into-rules.

[Towell and Shavlik, 1993] Towell, G. G. and Shavlik, J. W. (1993).
Extracting refined rules from knowledge-based neural networks.
Machine Learning, 13(1):71–101
DOI:10.1007/BF00993103.

[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.

[Tsukimoto, 2000] Tsukimoto, H. (2000).
Extracting rules from trained neural networks.
IEEE Transactions on Neural Networks, 11(2):377–389
DOI:10.1109/72.839008.

References XXVII

[van Gelder, 1990] van Gelder, T. (1990).
Why distributed representation is inherently non-symbolic.
In Dorffner, G., editor, *Konnektionismus in Artificial Intelligence und Kognitionsforschung. Proceedings 6. Österreichische Artificial Intelligence-Tagung (KONNAI), Salzburg, Österreich, 18. bis 21. September 1990*, volume 252 of *Informatik-Fachberichte*, pages 58–66. Springer
DOI:10.1007/978-3-642-76070-9_6.

[Voigt and von dem Bussche, 2017] Voigt, P. and von dem Bussche, A. (2017).
The EU General Data Protection Regulation (GDPR). A Practical Guide.
Springer
DOI:10.1007/978-3-319-57959-7.

[Wang et al., 2017] Wang, Q., Mao, Z., Wang, B., and Guo, L. (2017).
Knowledge graph embedding: A survey of approaches and applications.
IEEE Transactions on Knowledge and Data Engineering, 29(12):2724–2743
DOI:10.1109/TKDE.2017.2754499.

[Wang et al., 2020] Wang, S., Wang, Y., Wang, D., Yin, Y., Wang, Y., and Jin, Y. (2020).
An improved random forest-based rule extraction method for breast cancer diagnosis.
Applied Soft Computing, 86
DOI:10.1016/j.asoc.2019.105941.

[Wexler, 2017] Wexler, R. (2017).
When a computer program keeps you in jail: How computers are harming criminal justice.
New York Times
https://www.nytimes.com/2017/06/13/opinion/how-computers-are-harming-criminal-justice.html.

References XXVIII

[Wikipedia contributors, 2021] Wikipedia contributors (2021).
Decision tree learning — Wikipedia, the free encyclopedia.
[Online; accessed 17-September-2021]
https://en.wikipedia.org/w/index.php?title=Decision_tree_learning.

[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.

[Yedjour and Benyettou, 2018] Yedjour, D. and Benyettou, A. (2018).
Symbolic interpretation of artificial neural networks based on multiobjective genetic algorithms and association rules mining.
Applied Soft Computing, 72:177–188
DOI:10.1016/j.asoc.2018.08.007.

[Yuan and Zhuang, 1996] Yuan, Y. and Zhuang, H. (1996).
A genetic algorithm for generating fuzzy classification rules.
Fuzzy Sets Syst., 84(1):1–19
DOI:10.1016/0165-0114(95)00302-9.

References XXIX

[Zhang et al., 2005] Zhang, Y., Su, H., Jia, T., and Chu, J. (2005).
Rule extraction from trained support vector machines.
In Ho, T. B., Cheung, D. W., and Liu, H., editors, *Advances in Knowledge Discovery and Data Mining, 9th Pacific-Asia Conference, PAKDD 2005, Hanoi, Vietnam, May 18-20, 2005, Proceedings*, volume 3518 of *Lecture Notes in Computer Science*, pages 61–70. Springer
DOI:10.1007/11430919_9.

[Zhou et al., 2000] Zhou, Z., Chen, S., and Chen, Z. (2000).
A statistics based approach for extracting priority rules from trained neural networks.
In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, IJCNN 2000, Neural Computing: New Challenges and Perspectives for the New Millennium, Como, Italy, July 24-27, 2000, Volume 3*, pages 401–406. IEEE Computer Society
DOI:10.1109/IJCNN.2000.861337.

[Zhou et al., 2003] Zhou, Z., Jiang, Y., and Chen, S. (2003).
Extracting symbolic rules from trained neural network ensembles.
AI Communications, 16(1):3–15
http://content.iospress.com/articles/ai-communications/aic272.

[Zilke et al., 2016] Zilke, J. R., Mencía, E. L., and Janssen, F. (2016).
DeepRED – Rule extraction from deep neural networks.
In Calders, T., Ceci, M., and Malerba, D., editors, *Discovery Science - 19th International Conference, DS 2016, Bari, Italy, October 19-21, 2016, Proceedings*, volume 9956 of *Lecture Notes in Computer Science*, pages 457–473
DOI:10.1007/978-3-319-46307-0_29.