Towards Deep and Interpretable Rule Learning

Johannes Fürnkranz
Johannes Kepler University, Linz
Institute for Application-Oriented Knowledge Processing
Computational Data Analytics Group

juffi@faw.jku.at

Joint Work with
Florian Beck, Van Quoc Phuong Hyunh, Tomas Kliegr et al.

Towards Deep (and Interpretable?) Rule Learning

Johannes Fürnkranz
Johannes Kepler University, Linz
Institute for Applied Knowledge Processing
Computational Data Analytics Group

juffi@faw.jku.at

Joint Work with
Florian Beck, Van Quoc Phuong Hyunh, Tomas Kliegr et al.

AI and (Lack of) Interpretability

- Many AI systems can produce good performance
  - but cannot explain their decisions (→ "black-box models")
- Example:

  - When Kasparov lost a crucial game against Deep Blue in 1997, he demanded to see „the printouts"
    - meaning: explain to me how the computer derived its move
  - Impossible demand (→ complexity of chess)
  - Chess programs play extremely well
    - but cannot explain their moves

Deep Blue...

- built 1985 to 1997
  - first at CMU, later at IBM
  - by Feng-hsiung Hsu
- chess engine relying on
  - brute-force exhaustive search
  - chess-specific hardware
  - comparably simple evaluation function
  - (almost) no machine learning

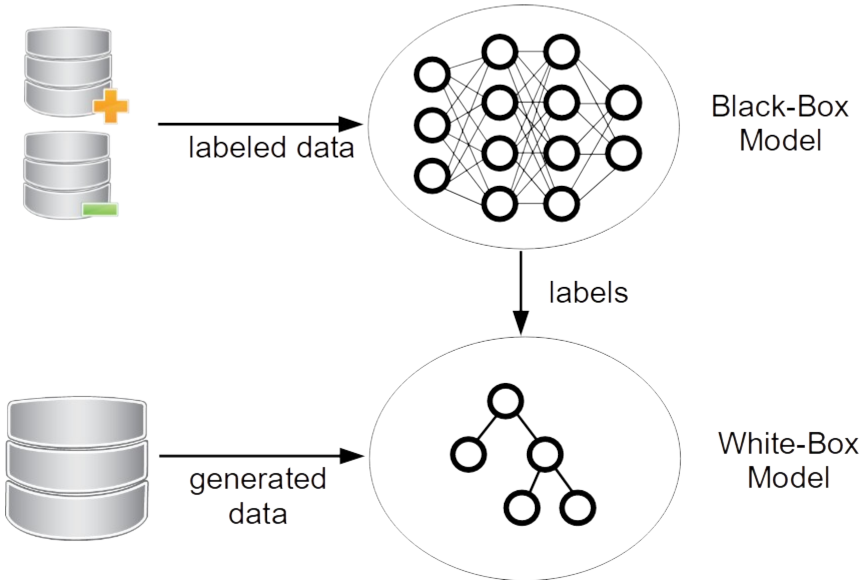
→ symbolic AI

… but is obviously a black-box model.

Two Roads Towards Explainable AI

1. Interpreting Black-Box Models

- Typical set-up:
  - use the BB model as an oracle for training an interpretable model
- variants are possible
  - e.g., only approximate a local region (LIME, etc.)

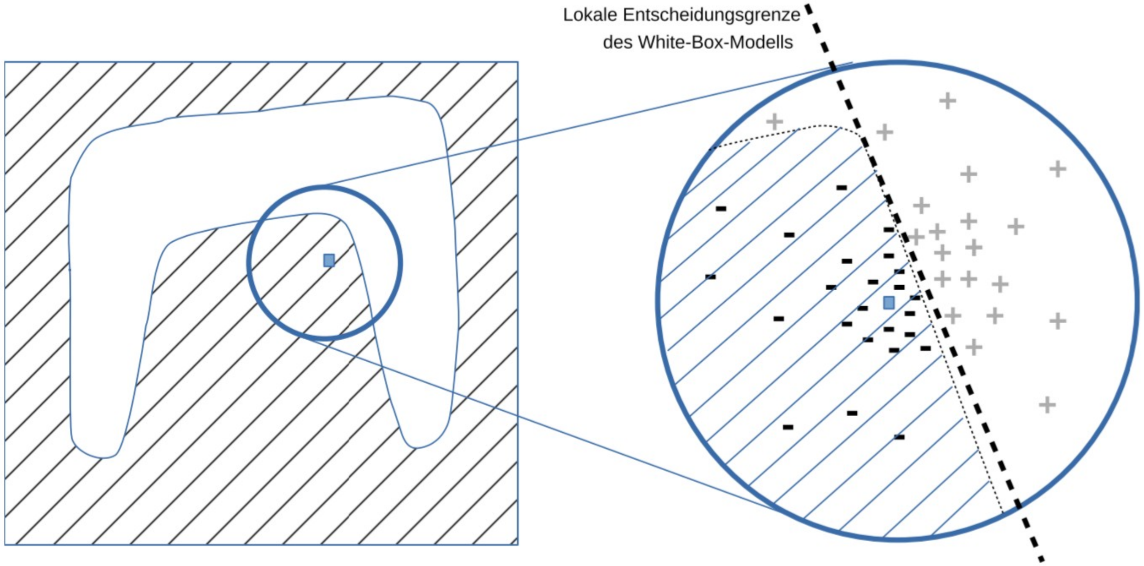


Finding Local Post-Hoc Explanations

JOHANNES KEPLER UNIVERSITÄT LINZ

- Local Interpretable Model-Agnostic Explanations (LIME) (Ribeiro, Singh, Guestrin 2017)
  - finds local explanations for a given example

- Key steps:
  1) generate examples that are close to a given test example
  2) use the black-box model for labeling these examples
  3) train a white-box model from this smaller dataset



Two Roads Towards Explainable AI

JOHANNES KEPLER
UNIVERSITÄT LINZ

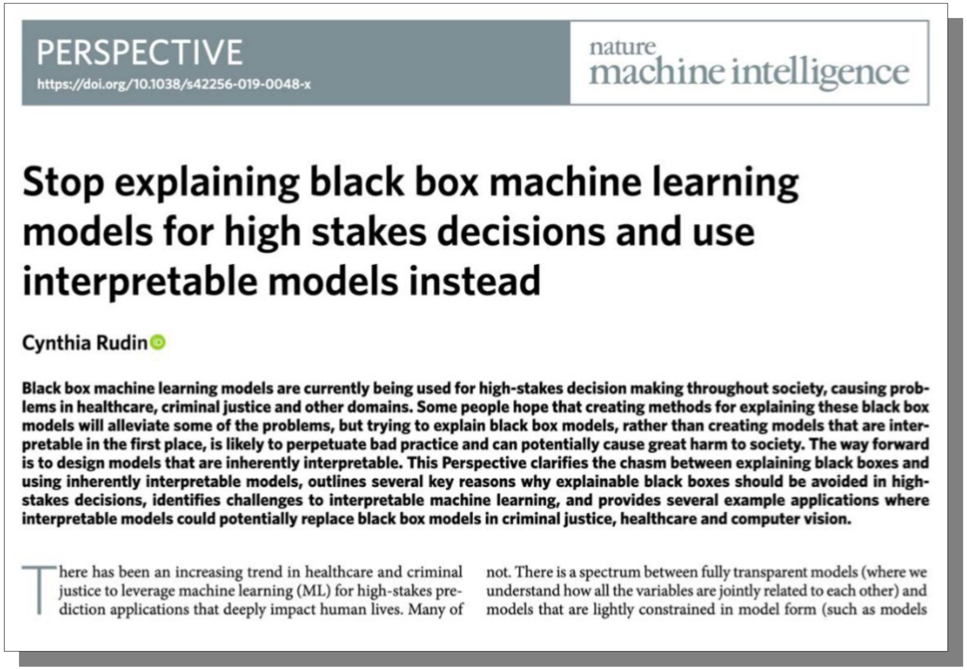
2. Direct learning of Interpretable Models

Pros:

- post-hoc explanations only approximate the BB model
- instead, the same model is used for explaining and for predicting

Cons:

- current interpretable models often do not reach the same performance
- they are not able to detect and use regularities that do not directly relate to the target concept.
- are often not as interpretable as they seem



PERSPECTIVE

https://doi.org/10.1038/s42256-019-0048-x

nature machine intelligence

Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead

Cynthia Rudin

Black box machine learning models are currently being used for high-stakes decision making throughout society, causing problems in healthcare, criminal justice and other domains. Some people hope that creating methods for explaining these black box models will alleviate some of the problems, but trying to explain black box models, rather than creating models that are interpretable in the first place, is likely to perpetuate bad practice and can potentially cause great harm to society. The way forward is to design models that are inherently interpretable. This Perspective clarifies the chasm between explaining black boxes and using inherently interpretable models, outlines several key reasons why explainable black boxes should be avoided in high-stakes decisions, identifies challenges to interpretable machine learning, and provides several example applications where interpretable models could potentially replace black box models in criminal justice, healthcare and computer vision.

There has been an increasing trend in healthcare and criminal justice to leverage machine learning (ML) for high-stakes prediction applications that deeply impact human lives. Many of not. There is a spectrum between fully transparent models (where we understand how all the variables are jointly related to each other) and models that are lightly constrained in model form (such as models

A Sample Database

No.	Education	Marital S.	Sex.	Children?	**Approved?**
1	Primary	Single	M	N	-
2	Primary	Single	M	Y	-
3	Primary	Married	M	N	+
4	University	Divorced	F	N	+
5	University	Married	F	Y	+
6	Secondary	Single	M	N	-
7	University	Single	F	N	+
8	Secondary	Divorced	F	N	+
9	Secondary	Single	F	Y	+
10	Secondary	Married	M	Y	+
11	Primary	Married	F	N	+
12	Secondary	Divorced	M	Y	-
13	University	Divorced	F	Y	-
14	Secondary	Divorced	M	N	+

Property of Interest
("class variable")

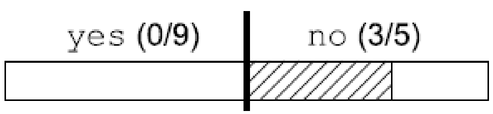
Subgroup Discovery

- Definition

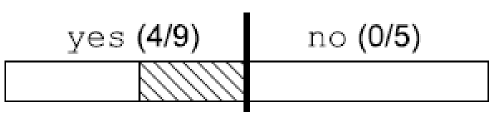
> "Given a population of individuals and a property of those individuals that we are interested in, **find population subgroups** that are statistically 'most interesting', e.g., are as large as possible and have the most unusual distributional characteristics with respect to the property of interest"
>
> (Klösgen 1996; Wrobel 1997)

- Examples

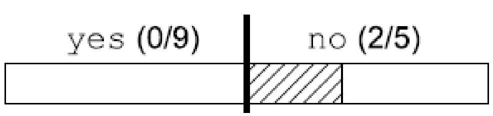
```
IF   MaritalStatus = single
 AND Sex = male
THEN Approved = no
```




```
IF   MaritalStatus = married
THEN Approved = yes
```




```
IF   MaritalStatus = divorced
 AND HasChildren = yes
THEN Approved = no
```



Rule-Based Models and Explanations

Rule-based allow to seamlessly move between global models and individual predictions

- Individual Rules as Local Explanations:
  - each rule provides an explanation for a local neighborhood ($\rightarrow$ subgroup discovery)
- Rule Sets as Interpretable Global Models:
  - the rules are combined into a rule set that provides a global explanation

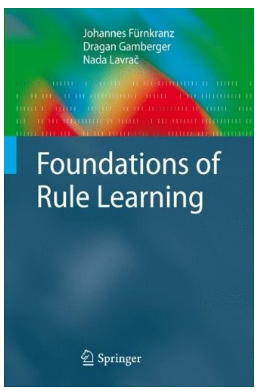
Nevertheless, interpretability of rules should not be taken for granted!

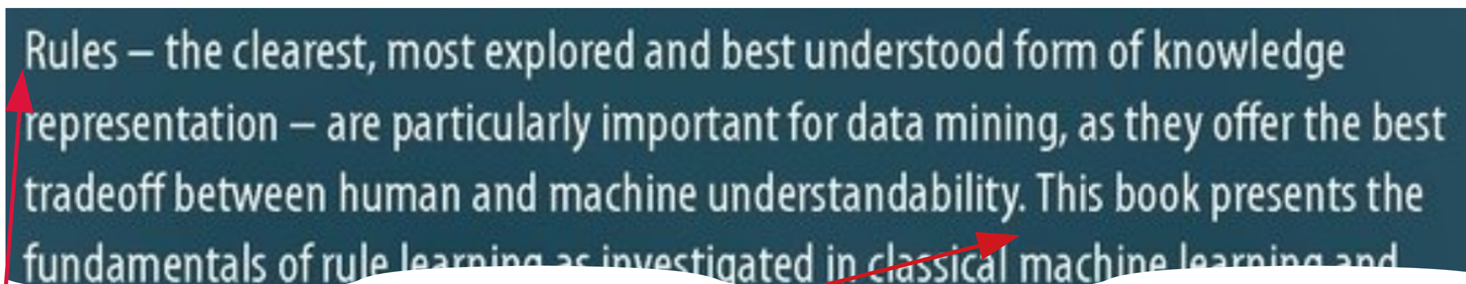
Interpretability and Rule Learning

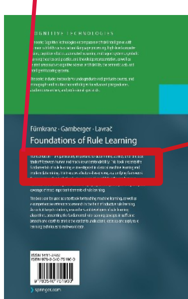
JOHANNES KEPLER
UNIVERSITÄT LINZ

Rules (and decision trees) are often equated with interpretable concepts

- If we learn rules, then we are interpretable
- Shorter models are more interpretable than longer models




Rules – the clearest, most explored and best understood form of knowledge representation – are particularly important for data mining, as they offer the best tradeoff between human and machine understandability. This book presents the fundamentals of rule learning as investigated in classical machine learning and



Note: The book has a 13-page index, which does not contain entries for understandability, interpretability, comprehensibility, or similar...

Understandability vs. Rule Length

Conventional Rule learning algorithms tend to learn short rules

- They favor to add conditions that exclude many negative examples

Typical intuition: Short rules are better

- long rules are less understandable, therefore short rules are preferable
- short rules are more general, therefore (statistically) more reliable and would have been easier to falsify on the training data

Claim: Shorter rules are not always better

- **Predictive Performance:** Longer rules often cover the same number of examples than shorter rules so that (statistically) there is no preference for choosing one over the other
- **Understandability:** In many cases, longer rules may be much more intuitive than shorter rules

→ *we need to understand understandability*!

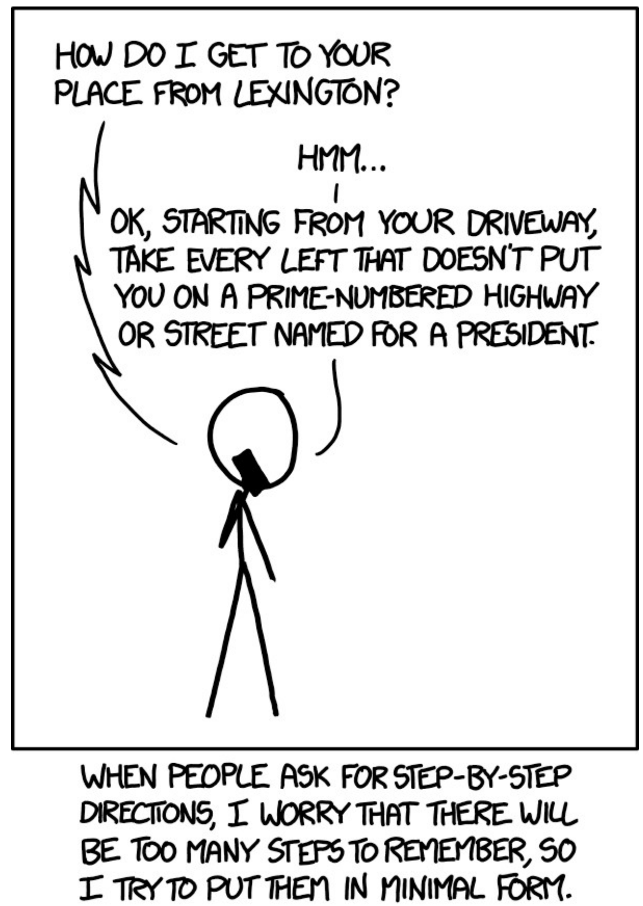
Are Shorter Explanations better?

- Shorter explanations are often more predictive than longer ones
- but do not need to be interpretable

Other dimensions:

- Representativeness
- Redundancy
- Coherence
- Structure
- ...

Kolmogorov Directions

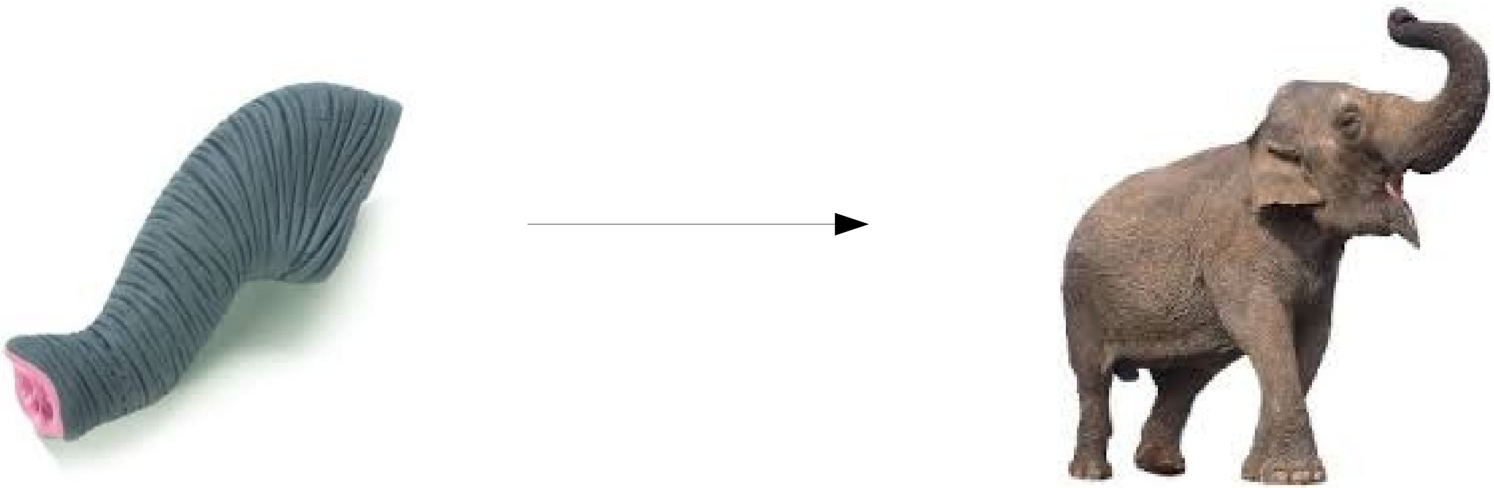


Source: https://www.xkcd.com/1155/
(Thanks to Jilles Vreeken for the pointer)

Discriminative Rules

- Allow to quickly **discriminate an object** of one category from objects of other categories
- Typically a few properties suffice

- Example:



Characteristic Rules

- Allow to characterize an object of a category
- Focus is on all properties that are **representative** for objects of that category

- Example:

Example Rules – Mushroom dataset

- The best three rules learned with conventional heuristics

```
poisonous :- odor = foul.              (2160,0)
poisonous :- gill-color = buff.        (1152,0)
poisonous :- odor = pungent.            (256,0)
```

- The best three rules learned with inverted heuristics

```
poisonous :- veil-color = white, gill-spacing = close,
             no bruises, ring-number = one,
             stalk-surface-above-ring = silky.   (2192,0)
poisonous :- veil-color = white, gill-spacing = close,
             gill-size = narrow, population = several,
             stalk-shape = tapering.              (864,0)
poisonous :- stalk-color-below-ring = white,
             ring-type = pendant, ring-number = one,
             stalk-color-above-ring = white,
             cap-surface = smooth, stalk-root = bulbuous,
             gill-spacing = close.                (336,0)
```

(Stecher, Janssen, Fürnkranz 2016)

Example Rules – Brain Ischemia

```
[149| 0] ischemia :- b.i. < 60.0001.
[140| 0] ischemia :- b.i. < 70.0001, fibrin. >= 3.8.
[137| 0] ischemia :- b.i. < 75.0001, fibrin. >= 3.9.
```

Regular heuristics find **Barthel index** and **fibrinogen value** as relevant for a brain stroke.

Inverted heuristics in addition refer to **age**, **diastolic blood pressure**, and **cholesterol**

```
[147| 0] ischemia :- rrrrdyast.. >= 70, fibrin. >= 2.8, b.i. < 60.0001.
[139| 0] ischemia :- age >= 58, rrrrdyast.. >= 80, b.i. < 60.0001.
[107| 0] ischemia :- rrrrdyast.. >= 80, fibrin. >= 3.5, b.i. < 65.0001, chol. >= 5.2.
```

Is Rule Length an Indicator for Interpretability?

(Fürnkranz, Kliegr, Paulheim 2020)

JOHANNES KEPLER UNIVERSITÄT LINZ

Result of a crowd-sourcing experiment in 4 domains

- in two out of four domains there was no correlation
- in the other two longer rules were considered to be more plausible

dataset	units	judg	qfr [%]	Kendall's τ		Spearman's ρ	
Traffic	80	412	12	0.05	(0.226)	0.06	(0.230)
Quality	36	184	11	**0.20**	(0.002)	**0.23**	(0.002)
Movies	32	156	14	−0.01	(0.837)	−0.02	(0.828)
Mushroom	10	250	14	**0.37**	(0.000)	**0.45**	(0.000)
total	158	962	13				

→ no evidence that shorter rules are better understood

(Fürnkranz, Kliegr 2018)

The Need for Interpretability Biases

- Understandability is currently mostly defined via rule length
  - Occam's Razor: Shorter rules are better
- On the other hand, longer rules are often more convincing
  - Characteristic rules, closed itemsets, formal concepts, rules learned with inverted heuristics, ...
- To define interpretability biases we need to understand human cognitive biases
  - **Representativeness**: a rule that is more typical to what we expect is more convincing
  - **Semantic coherence**: rules that have semantically similar conditions are better
  - **Recognition**: rules with well-recognized conditions are better
  - **Structure:** flat rules are not very natural

AQ-Style Rule Induction

- Oldest type of rule induction algorithm (Michalski 1969)
  - e.g., also used in Progol

Algorithm 1 AQ-type rule induction

function AQ(E,F,h)
  $R = \emptyset$
  while $E \neq \emptyset$ **do**
    randomly select example $\langle \mathbf{x}, y \rangle \in E$
    $r \leftarrow \arg\max_{r'=(B \rightarrow y), B \subset F_{\mathbf{x}}} h(r')$
    $R \leftarrow R \cup \{r\}$
    $E \leftarrow E \setminus E_r$
  end while
  return R
end function

(Greedily) find a subset B of all features $F_{\mathbf{x}}$ that cover a randomly selected example $\mathbf{x}$ so that some quality function h is optimized

Covering: Repeat until all examples are covered by one (or more) rule

CN2-style rule induction

- Most popular type of rule induction (Clark & Niblett, 1989)
  - used in most covering rule learning algorithms

Algorithm 2 CN2-type rule induction

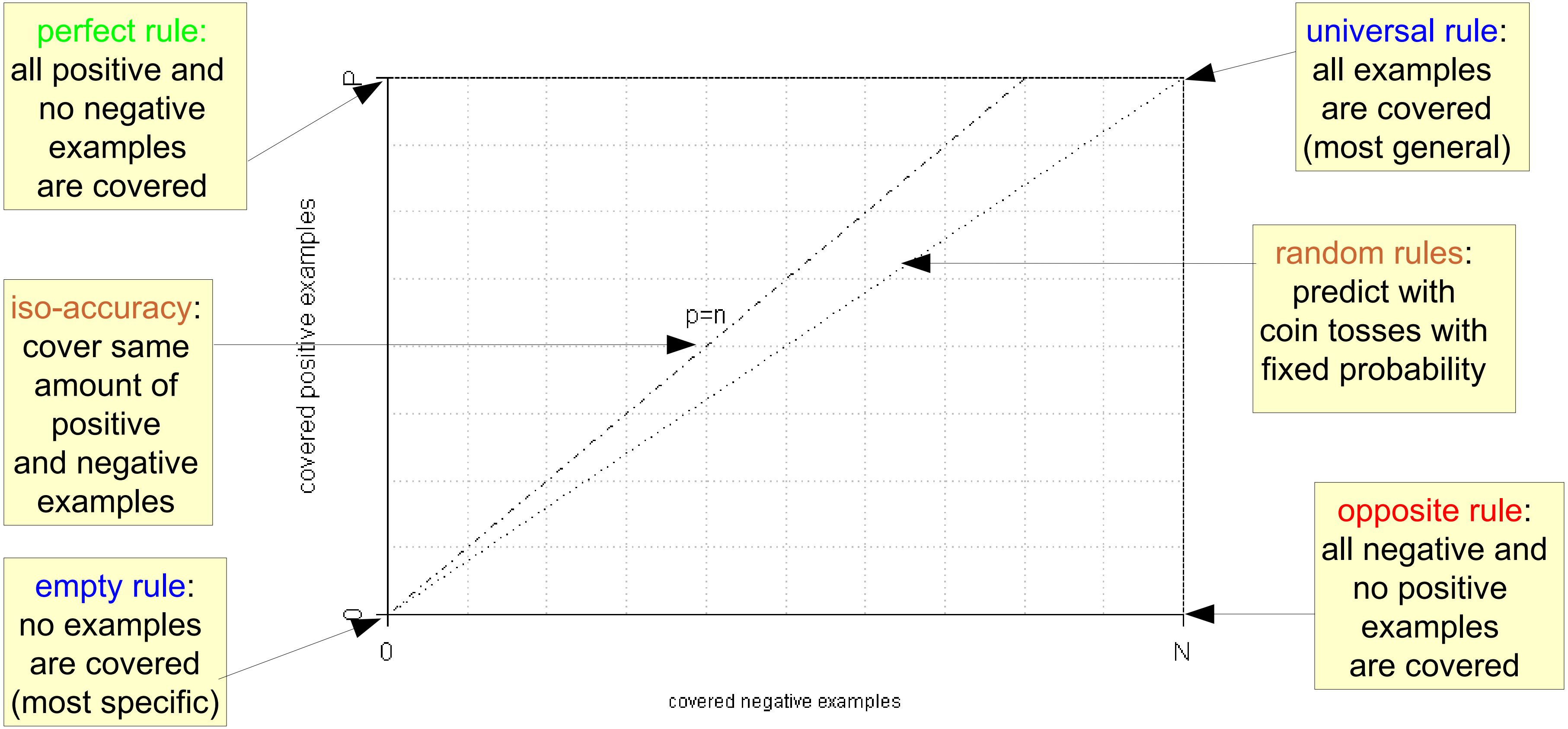
function CN2(E,F,h)
  $R = \emptyset$
  while $E \neq \emptyset$ **do**
    $r \leftarrow \arg\max_{r'=(B \to y), B \subset F, y \in C} h(r')$
    $R \leftarrow R \cup \{r\}$
    $E \leftarrow E \setminus E_r$
  end while
  return R
end function

(Greedily) find a subset B of all features F so that some quality function h is optimized

Covering: Repeat until all examples are covered by one (or more) rule

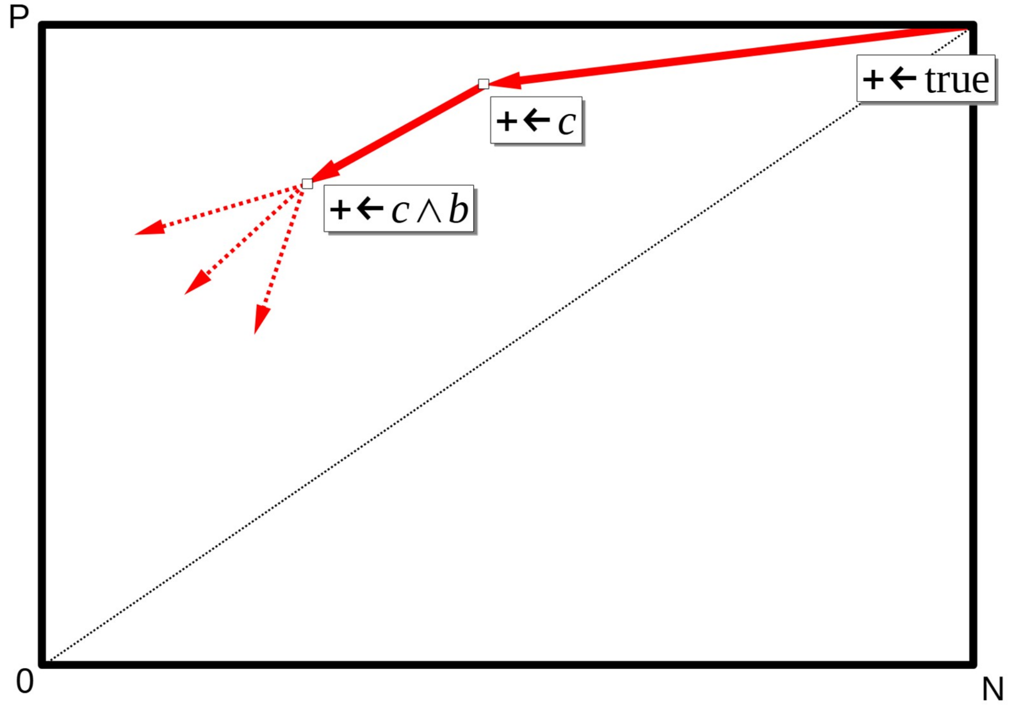
Coverage Spaces

- good tool for visualizing properties of rule evaluation heuristics
  - each point is a rule covering p positive and n negative examples



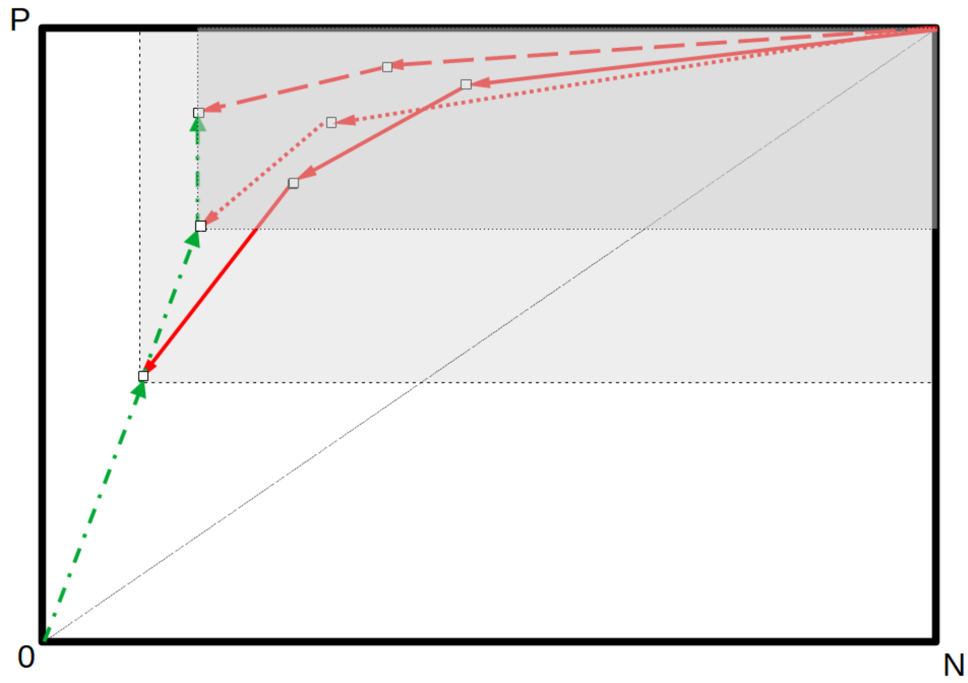
Learning Conjunctive Rules

- Most rule learning algorithms learns conjunctive rule bodies
- Learning a single conjunctive rule in coverage space
  - in a greedy top-down (general-to-specific) search



Learning DNFs via Covering

- successive refinement of individual rules (red)
- reductions in coverage space by removing covered examples (shades of grey)
- bulding up the DNF by adding conjunctive rules (green)



(Huynh, Fürnkranz, Beck 2023)

Locally Optimal Rule Induction

- Try to combine the best of AQ-style and CN2-style induction
  - no dependence on random example selection
  - efficient reduction of feature subsets
  - strive for the best rule for each example

Algorithm 3 locally optimal rule induction

function LORD(E,F,h)
  $R = \emptyset$
  for all $\langle \mathbf{x}, y \rangle \in E$ **do**
    $r \leftarrow \arg\max_{r'=(B \rightarrow y), B \subset F_{\mathbf{x}}} h(r')$
    $R \leftarrow R \cup \{r\}$
  end for
  return R
end function

No random selection: learn a rule for every example x

No covering: Stop when every example has its best rule.

(Huynh, Fürnkranz, Beck 2023)

The LORD Rule Learner

JOHANNES KEPLER UNIVERSITÄT LINZ

- **Key idea**
  - aim at learning the best rule for each training example
    - local optimum in a local neighborhood around the training example
    - motivated by the XAI idea of providing explanations for each example
  - the result is one rule for each training example
    - almost, because suboptimal and duplicate rules are removed
- **Implementation characteristics**

https://github.com/vqphuynh/LORD

  - Make use of efficient data structures known from association rule mining like PPC-trees and N-lists
    - can efficiently summarize the dataset in one pass
  - Use a rule learning heuristic for guiding its greedy search
    - e.g. the m-estimate
  - Inherently parallel search for locally optimal rules
    - LORD can efficiently tackle very large example sets

LORD Evaluation

JOHANNES KEPLER UNIVERSITÄT LINZ

- 24 datasets with various sizes
- the largest with 5 million examples and 19 attributes

#	Datasets	# Exs.	# Attr.	Attr. Types	Missing Values	Class Distributions (%)
1	*lymph*	148	19	categorical	no	54.7; 41.2; 2.7; 1.3
2	*wine*	178	14	numeric	no	33.2; 39.9; 26.9
3	*vote*	435	17	categorical	yes	54.8; 45.2
4	*breast-cancel*	699	10	numeric	yes	65.5; 34.5
5	*tic-tac-toe*	958	10	categorical	no	65.3; 34.7
6	*german*	1,000	21	mix	no	70; 30
7	*car-eval*	1,728	7	categorical	no	22.3; 3.9; 70; 3.8
8	*hypo*	3,163	26	mix	yes	95.2; 4.8
9	*kr-vs-kp*	3,196	37	categorical	no	52.2; 47.8
10	*waveform*	5,000	22	numeric	no	33.2; 32.9; 33.9
11	*mushroom*	8,124	23	categorical	yes	51.7; 48.3
12	*nursery*	12,960	9	categorical	no	33.3; 32.9; 31.2; 2.5; 0.01
13	*adult*	48,842	14	mix	yes	76; 24
14	*bank*	45,211	17	mix	no	11.7; 88.3
15	*skin*	245,057	4	numeric	no	20.7; 79.3
16	*s-mushroom*	61,069	21	mix	yes	44.5; 55.5
17	*connect-4*	67,557	42	categorical	no	65.8; 24.6; 9.6
18	*PUC-Rio*	165,632	19	mix	no	28.6; 26.2; 7.5; 7.1; 30.6
19	*census*	299,285	41	mix	yes	93.8; 6.2
20	*gas-sensor-11*	919,438	11	numeric	no	32.9; 29.8; 37.3
21	*gas-sensor-12*	919,438	12	numeric	no	32.9; 29.8; 37.3
22	*cover-type*	581,012	55	mix	no	36.4; 48.8; 6.2; 0.5; 1.6; 3; 3.5
23	*pamap2*	1,942,872	33	numeric	yes	9.9; 6; 9.5; 5.4; 2.5; 9.8; 12.3; 9; 5.1; 12.3; 8.5; 9.7
24	*susy*	5,000,000	19	numeric	no	54.2; 45.8

Results

JOHANNES KEPLER UNIVERSITÄT LINZ

- Accuracy
  - better than Ripper and other modern rule learner (not ensembles)

# Datasets	Lord (m = 0.1)	Lord (best m)	Lord* (m = 0.1)	Ripper (o = 0)	Ripper (o = 2)	CMAR	CG	PyIDS (k = 50)	PyIDS (k = 150)
Avg. acc. (3-6,8-9,11,13-16)	**0.9416**	**0.9436**	**0.9415**	0.9365	0.9374	0.916	0.9222	0.8137	0.8312
Avg. acc. (1-22)	**0.9268**	**0.9311**	**0.9266**	0.9073	0.9152	0.8056	//	0.7077	0.7287
Avg. ranks (1-22)	**3.14**	**1.84**	**3.3**	4.48	3.59	5.2	//	7.57	6.89

- Run-time
  - only few algorithms could tackle the largest datasets

#	Datasets	Lord (m = 0.1)	Lord (best m)	Lord* (m = 0.1)	Ripper (o = 0)	Ripper (o = 2)	CMAR	CG	PyIDS (k = 50)	PyIDS (k = 150)
23	pamap2	6063	6044	386	Out of memory	Out of memory	50.4	//	Out of memory	Out of memory
24	susy	52592	51218	15350	Out of memory	Out of memory	97.4	Out of time	9435	29109
Avg. runtime (1-22)		**94**	**95.1**	**31.5**	342	8642.8	116	//	274.7	2568.6
Avg. ranks (1-22)		3.5	3.75	**1.73**	2.95	5.09	4.89	//	6.27	7.82

Results

- Number of learned rules
  - enormous, e.g., 1.6 million rules for the susy dataset

#	Datasets	Lord (m = 0.1)		Lord (best m)		Lord* (m = 0.1)		Ripper (o = 0)		Ripper (o = 2)		CMAR		CG	PyIDS (k = 50)		PyIDS (k = 150)	
23	pamap2	16827	3.07	14137	3.09	15824	3.05	Out of memory		Out of memory		486	2.54	//	Out of memory		Out of memory	
24	susy	1611856	4.30	1201338	4.10	976522	4.30	Out of memory		Out of memory		637	1.40	Out of time	18	2.0	63	2.0
Avg. values (1-22)		8390.6	3.54	8261.4	3.5	6361.2	3.49	104.6	4.12	111.5	3.74	1945.1	3.06	//	**16.8**	**2.06**	**50.4**	**2.1**
Avg. ranks (1-22)		6.82	5.86	6.39	5.82	5.25	4.95	2.73	5.16	2.23	3.95	6.45	4.86	//	**1.91**	**2.45**	4.23	**2.93**

- This is certainly not interpretable
  - However, each rule is the perfect explanation for one of the training examples
- Ongoing Work:
  - LORD as a post-hoc XAI tool
  - transductive learning of rules (this is harder than you may think…)

Example: Parity / XOR

- Consider the parity / XOR problem
  - $n + r$ binary attributes sampled with an equal distribution of 0/1
  - n relevant binary attributes (the first n w.l.o.g.)
  - r irrelevant binary attributes

- Target concept:
  - is there an even number of 1's in the relevant attributes?

Encoding Parity with a Flat Rule Set

JOHANNES KEPLER UNIVERSITÄT LINZ

Most rule learning algorithms learn flat theories

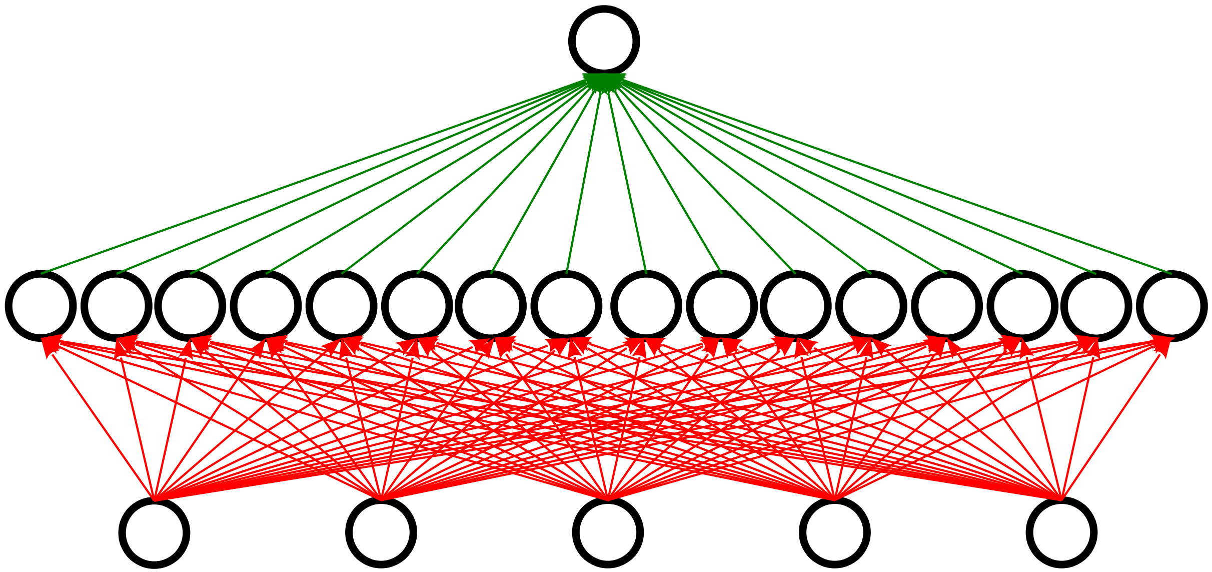
- n-bit parity needs 2^{n-1} flat rules, no shorter encoding is possible
- each rule encoding one positive case in the truth table

```
parity :-     x1,     x2,     x3,     x4, not x5.
parity :-     x1,     x2, not x3, not x4, not x5.
parity :-     x1, not x2,     x3, not x4, not x5.
parity :-     x1, not x2, not x3,     x4, not x5.
parity :- not x1,     x2, not x3,     x4, not x5.
parity :- not x1,     x2,     x3, not x4, not x5.
parity :- not x1, not x2,     x3,     x4, not x5.
parity :- not x1, not x2, not x2, not x4, not x5.
parity :-     x1,     x2,     x3, not x4,     x5.
parity :-     x1,     x2, not x3,     x4,     x5.
parity :-     x1, not x2,     x3,     x4,     x5.
parity :- not x1,     x2,     x3,     x4,     x5.
parity :- not x1, not x2, not x3,     x4,     x5.
parity :- not x1, not x2,     x3, not x4,     x5.
parity :- not x1,     x2, not x3, not x4,     x5.
parity :-     x1, not x2, not x2, not x4,     x5.
```

DNF formula with 2^{n-1} literals, each having n variables

Network View of a Flat Rule Set

- Flat Rule Sets can be converted into a network using a single AND and a single OR layer ($\rightarrow$ a DNF expression)



- Each node in the hidden layer corresponds to one rule
  - typically it is a local pattern, covering part of the target

The Sucess of Deep Learning

- **Hypothesis:**
  Most of the success of deep learning is due to the fact that it allows to learn **deep structures** in which auxiliary concepts develop which will facilitate the learning process

- **Problem:**
  No state-of-the-art *rule learning* algorithm is able to learn such structured, purely declarative rule bases

Encoding Parity with a Structured Rule Base

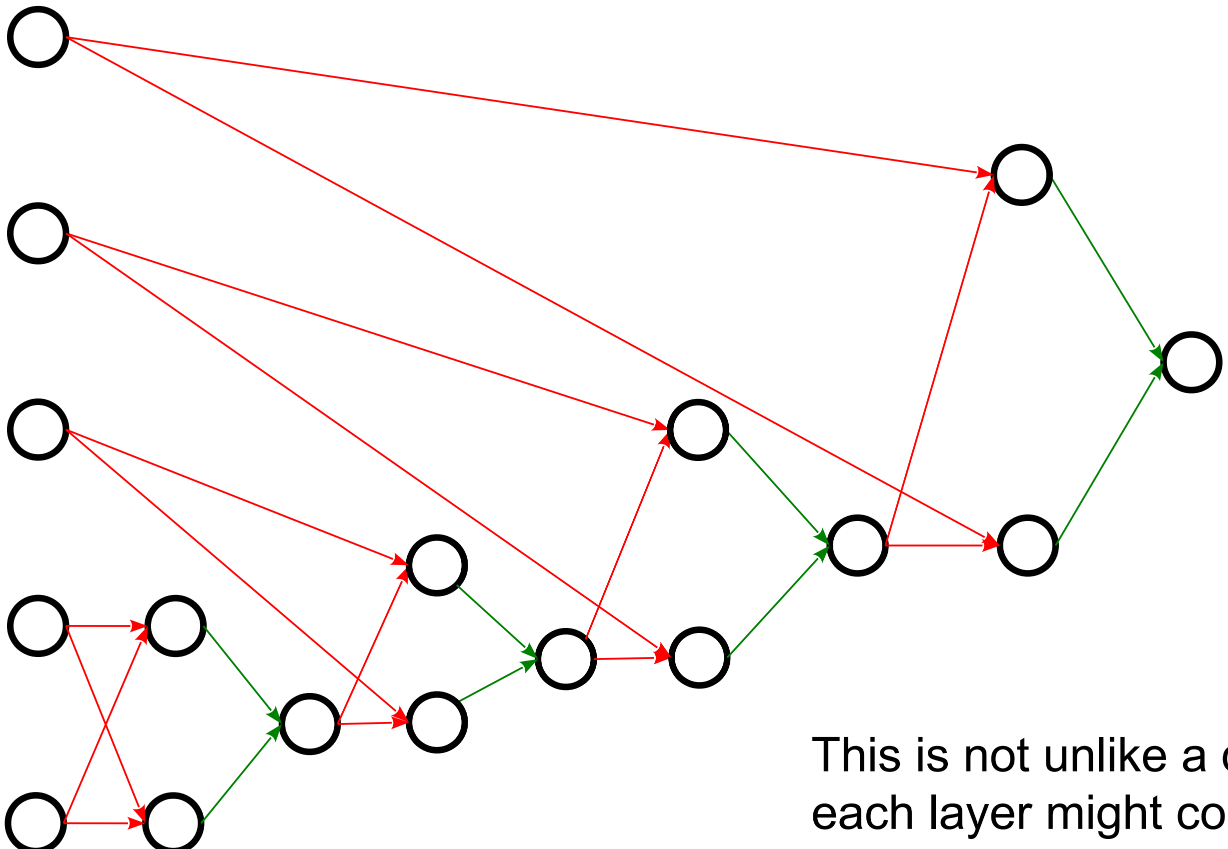
But structured concepts are often more interpretable

- in parity we need only $O(n)$ rules with intermediate concepts

```
parity45    :-     x4,     x5.
parity45    :- not x4, not x5.

parity345   :-     x3, not parity45.
parity345   :- not x3,     parity45.

parity2345  :-     x2, not parity345.
parity2345  :- not x2,     parity345.

parity      :-     x1, not parity2345.
parity      :- not x1,     parity2345.
```

Network View of a Structured Rule Base

- This is encodes a deep network structure



This is not unlike a deep network:
each layer might contain more nodes,
which eventually are not needed

Why is it good to learn deep rule sets?

- **Expressivity?** It does not necessarily increase expressivity
  - any structured rule base can be converted into an equivalent DNF expression, i.e., a flat set of rules
    - but this is also true for NNs → universal approximation theorem (one layer is sufficient; Hornik et al. 1989)
  - in both cases the number of terms (size of hidden layers, conjuncts in the DNF) is unbounded
    - Note that a disjunction of all examples is also a DNF expression

Why is it good to learn deep rule sets?

- **Interpretability?**
  - structured rule sets may be more compact
  - are they more interpretable?

- **Example**: Why is $\mathbf{x} = (1,1,1,0,1,0,0,1,0,0,...)$ in parity?

```
parity :-     x1,     x2,     x3,     x4, not x5.
parity :-     x1,     x2, not x3, not x4, not x5.
parity :-     x1, not x2,     x3, not x4, not x5.
parity :-     x1, not x2, not x3,     x4, not x5.
parity :- not x1,     x2, not x3,     x4, not x5.
parity :- not x1,     x2,     x3, not x4, not x5.
parity :- not x1, not x2,     x3,     x4, not x5.
parity :- not x1, not x2, not x2, not x4, not x5.
parity :-     x1,     x2,     x3, not x4,     x5.
parity :-     x1,     x2, not x3,     x4,     x5.
parity :-     x1, not x2,     x3,     x4,     x5.
parity :- not x1,     x2,     x3,     x4,     x5.
parity :- not x1, not x2, not x3,     x4,     x5.
parity :- not x1, not x2,     x3, not x4,     x5.
parity :- not x1,     x2, not x3, not x4,     x5.
parity :-     x1, not x2, not x2, not x4,     x5.
```

Even though the rule set is quite complex, we only need a single rule for giving a good explanation.

Why is it good to learn deep rule sets?

JOHANNES KEPLER UNIVERSITÄT LINZ

- **Interpretability?**
  - structured rule sets may be more compact
  - are they more interpretable?

    → Only if all subconcepts are easily interpretable!
  - **Example**: Why is $\mathbf{x} = (1,1,1,0,1,0,0,1,0,0,...)$ in parity?

```
parity    :-      x1, not parity2345.
parity    :- not x1,     parity2345.
```

Even though the rule set is more compact, we need to understand every subconcept in order to interpret the explanation.

(Fürnkranz et al. 2020)

Why is it good to learn structured rule bases?

- **Explicit representation of all aspects of the decision function**
  - rule sets are typically **not declarative**, require some sort of **tie breaking**
- two main approaches
  - **weighted rules** / probabilistic rules

$$\begin{array}{ll} \boldsymbol{r}_1(0.8) : a \wedge b & \rightarrow x \\ \boldsymbol{r}_2(0.9) : b \wedge c & \rightarrow y \\ \boldsymbol{r}_3(0.7) : c \wedge d & \rightarrow x \\ \boldsymbol{d} : & \rightarrow z \end{array}$$

max: y (0.9)

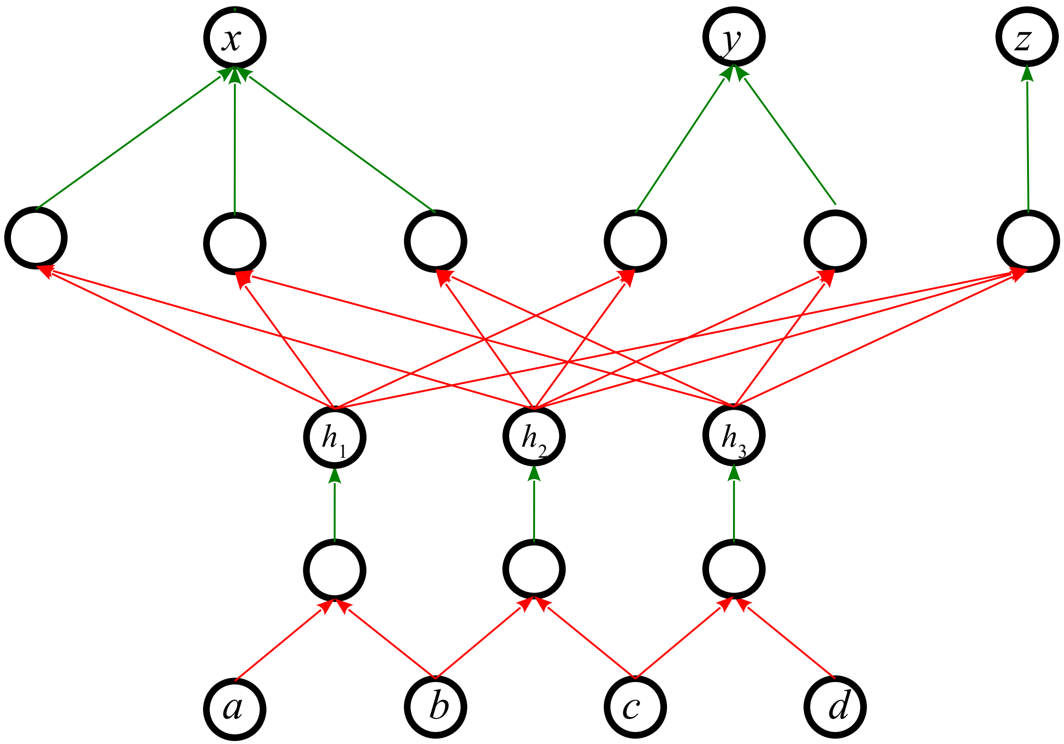
sum: x (0.7+0.8 > 0.9)

  - **decision lists** $\mathcal{D} = (\boldsymbol{r}_2, \boldsymbol{r}_1, \boldsymbol{r}_3, \boldsymbol{d})$
    - sort the rules according to some criterion
      - e.g., order in which they are learned
      - e.g., order according to weight (effectively equivalent to using weighted max)
    - use the first rule that fires

Declarative Version of Weighted Rule Sets

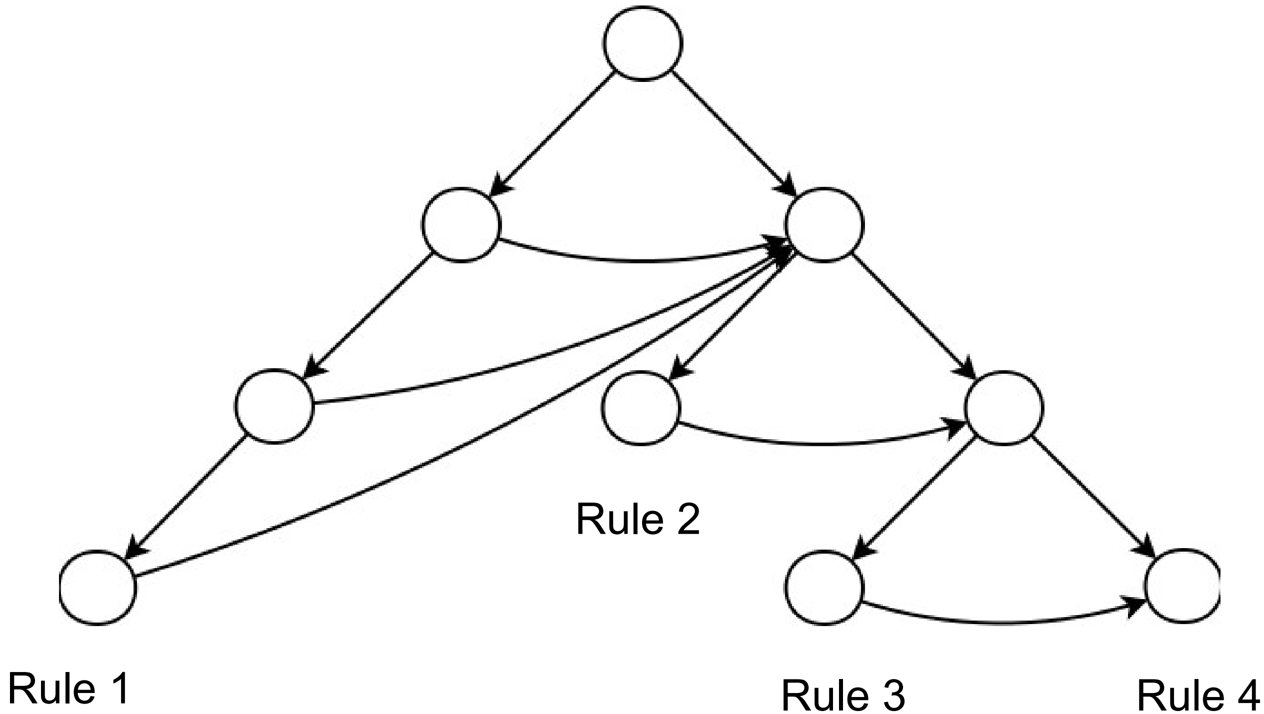
- Tie Breaking with Majority vote

$$
\begin{array}{r}
\hline
a \wedge b \rightarrow h_1 \\
b \wedge c \rightarrow h_2 \\
c \wedge d \rightarrow h_3 \\
h_1 \wedge h_3 \rightarrow x \\
h_1 \wedge \neg h_2 \rightarrow x \\
h_3 \wedge \neg h_2 \rightarrow x \\
h_2 \wedge \neg h_1 \rightarrow y \\
h_2 \wedge \neg h_3 \rightarrow y \\
\neg h_1 \wedge \neg h_2 \wedge \neg h_3 \rightarrow z \\
\hline
\end{array}
$$



Declarative Version of Decision List

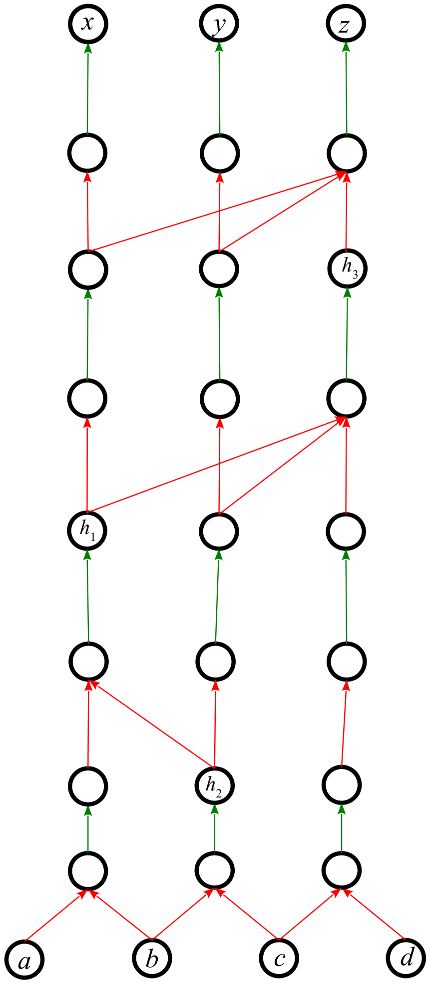
- A decision list is a decision graph, where not satisfied condition takes you to the start of the next rule
- Example of a decision list with 4 rules with 4, 2, 2, 1 conditions



Declarative Version of Decision List

- In our example

$$
\begin{array}{r}
\hline
b \wedge c \rightarrow h_2 \\
h_2 \rightarrow y \\
\neg h_2 \wedge a \wedge b \rightarrow h_1 \\
h_1 \rightarrow x \\
\neg h_1 \wedge \neg h_2 \wedge c \wedge d \rightarrow h_3 \\
h_3 \rightarrow x \\
\neg h_1 \wedge \neg h_2 \wedge \neg h_3 \rightarrow z \\
\hline
\end{array}
$$

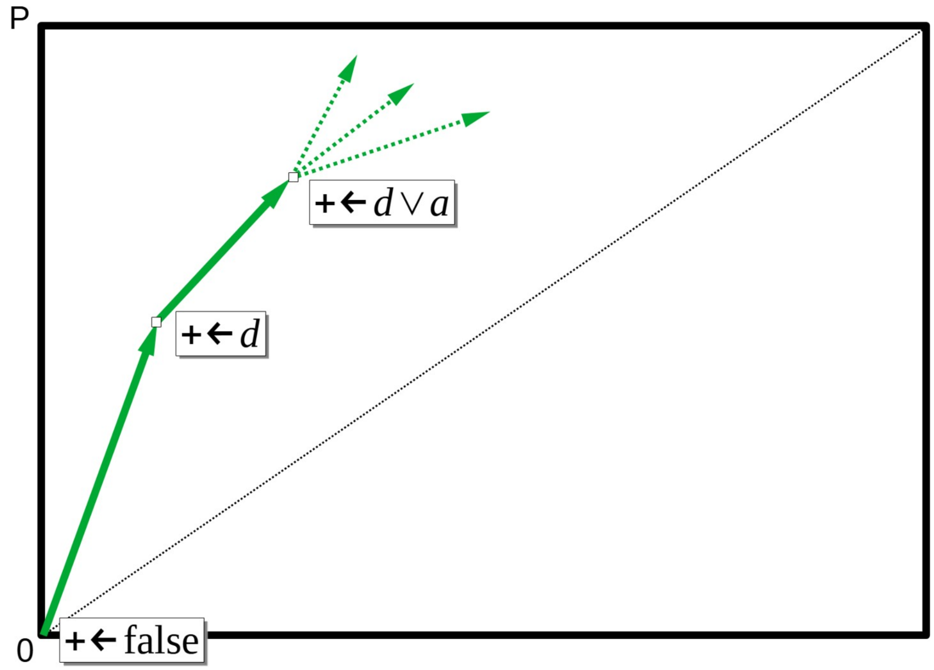


Why is it good to learn structured rule bases?

- **Learning Efficiency**
  - the hope is that deeper structures might be easier to learn
  - possibly contain fewer “parameters” that need to be found

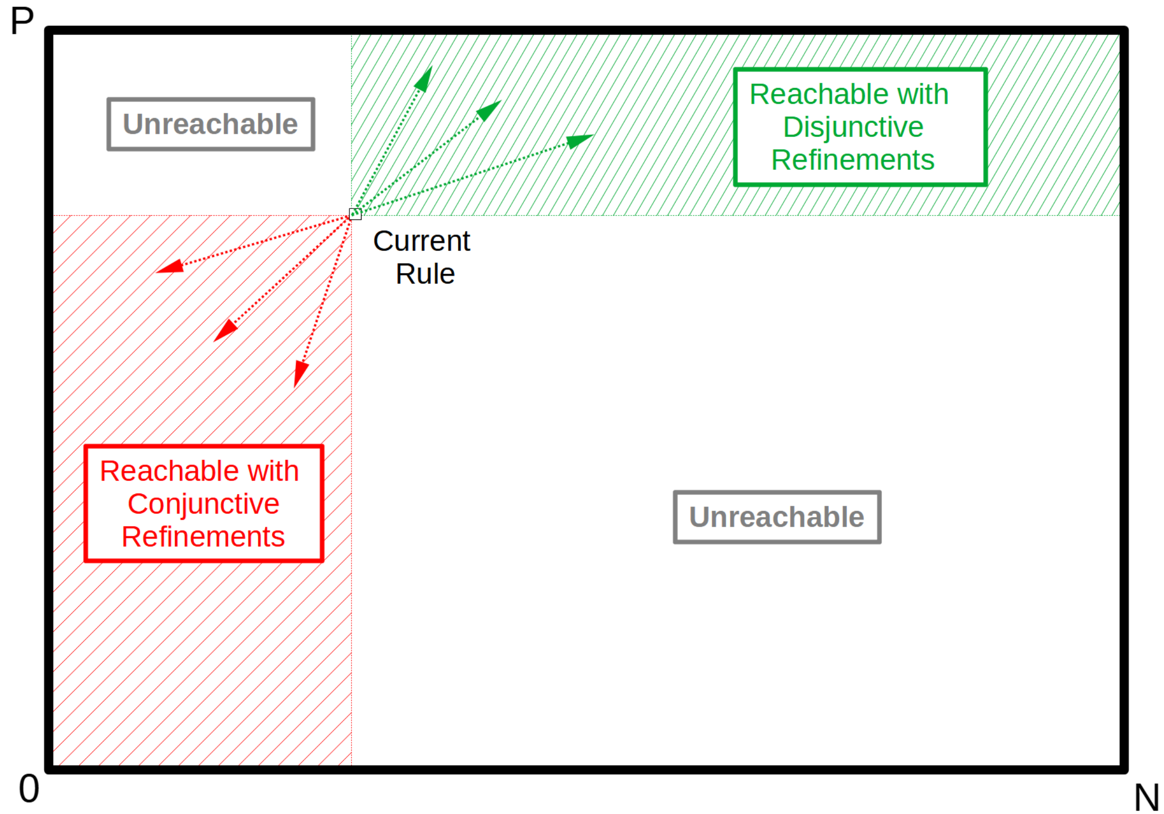
Learning Disjunctive Rules

- Disjunctive rules can be learned analogously to conjunctive ones
  - when these are combined conjunctively, it effective learns a CNF definition for the concept
- Learning a disjunctive single rule in coverage space:



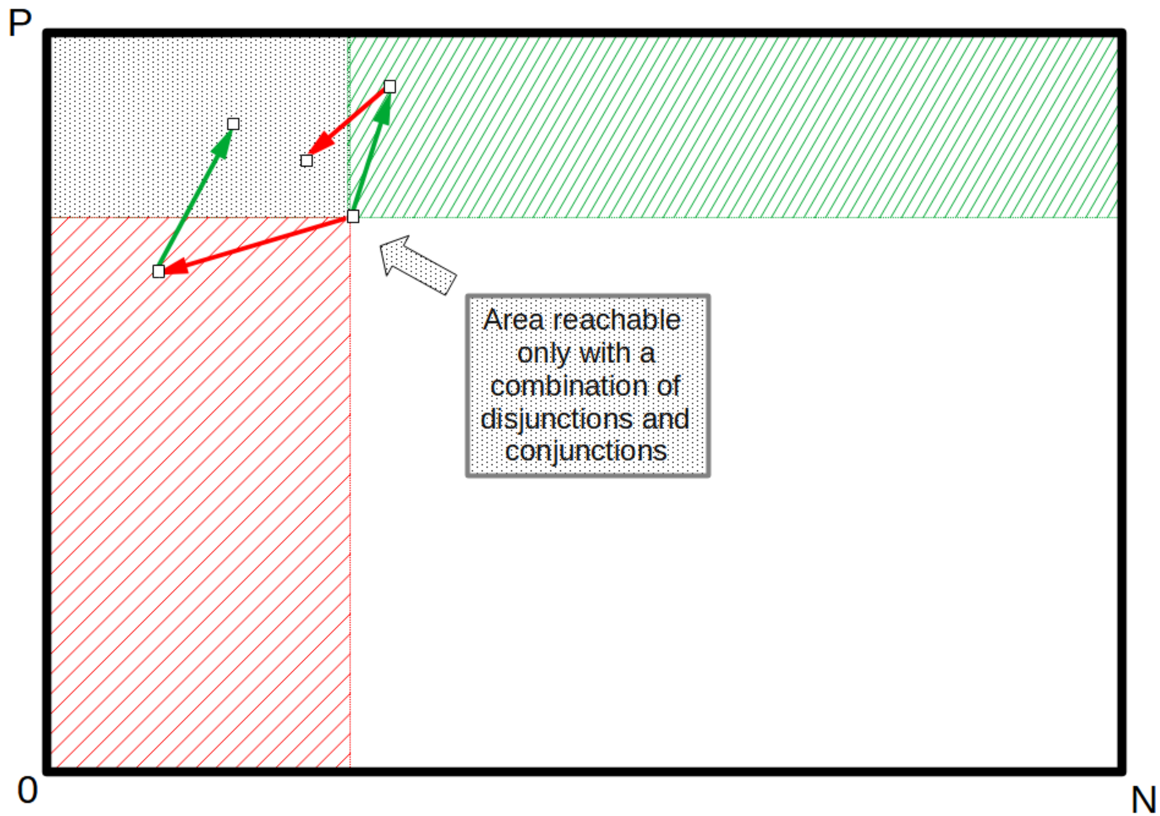
Limitations of Uni-Directional Refinements

→ The regions in coverage space that can be reached with successive (conjunctive or disjunctive) refinements are limited



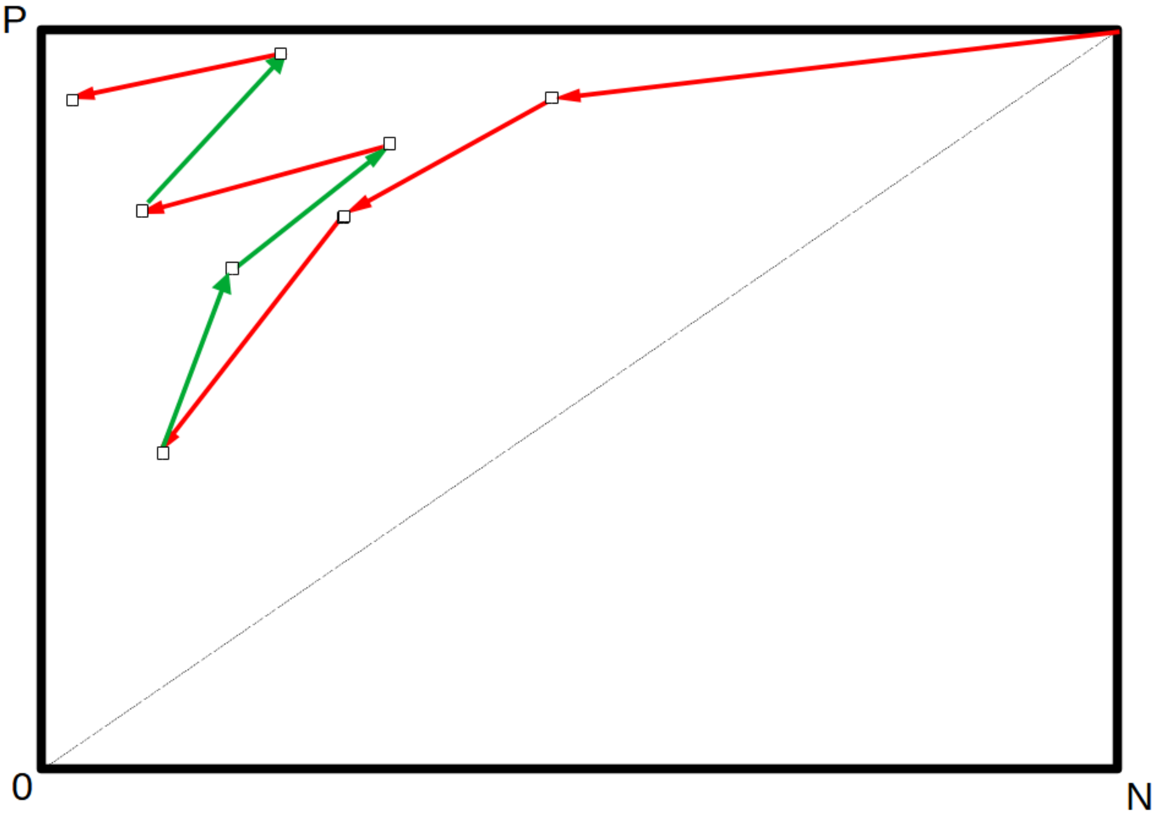
Bi-Directional Refinements

- This can be overcome with by allowing successive alternations of conjunctions and disjunctions



Bi-Directional Refinements

- ...which essentially corresponds to multiple alternating AND/OR layers



How to Learn Deep Rule Sets

1. The Neural Network Approach
   - fix a network structure and optimize its parameters

   a) Binary/Ternary Neural Networks
      - most of the works focus on (memory) efficiency, not on logic interpretability
      - Work in Progress: Incremental Freezing of Neural Network Weights

   b) Differentiable Logic
      - most of the works focus on first-order logic
      - diff-logic is an interesting exception

   c) Sum/Product Networks
      - focus on probabilities

→ We did a study in order to compare deep and shallow structure with a simple optimization algorithm (randomized hill-climbing)

(Beck & Fürnkranz 2020)

Does a Deep Structure help?

JOHANNES KEPLER UNIVERSITÄT LINZ

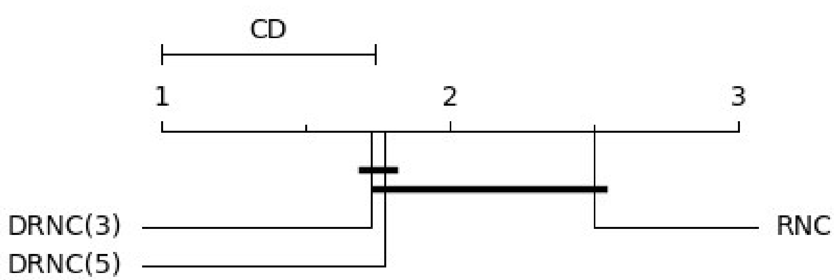
- To answer this empirically, we need to **compare** a powerful **shallow** rule learner with a powerful **deep** rule learner
  - But we do not have a powerful deep rule learner… (yet)
- Instead, we use a **simple optimization algorithm** to learn both, deep and shallow representations
  1) Fix a network architecture
     - Shallow, single layer network RNC: [20]
     - Deep 3-layer network DRNC(3): [32, 8, 2]
     - Deep 5-layer network DRNC(5): [32, 16, 8, 4, 2]
  2) Initialize Boolean weights probabilistically
  3) Use stochastic local search to find best weight „flip“ on a mini-batch of data until convergence
  4) Optimize finally on whole training set

(Beck & Fürnkranz 2020)

Results on Artificial Datasets

- 20 artificial datasets with 10 Boolean inputs, 1 Boolean output
  - generated from a randomly initialized (deep) Boolean network

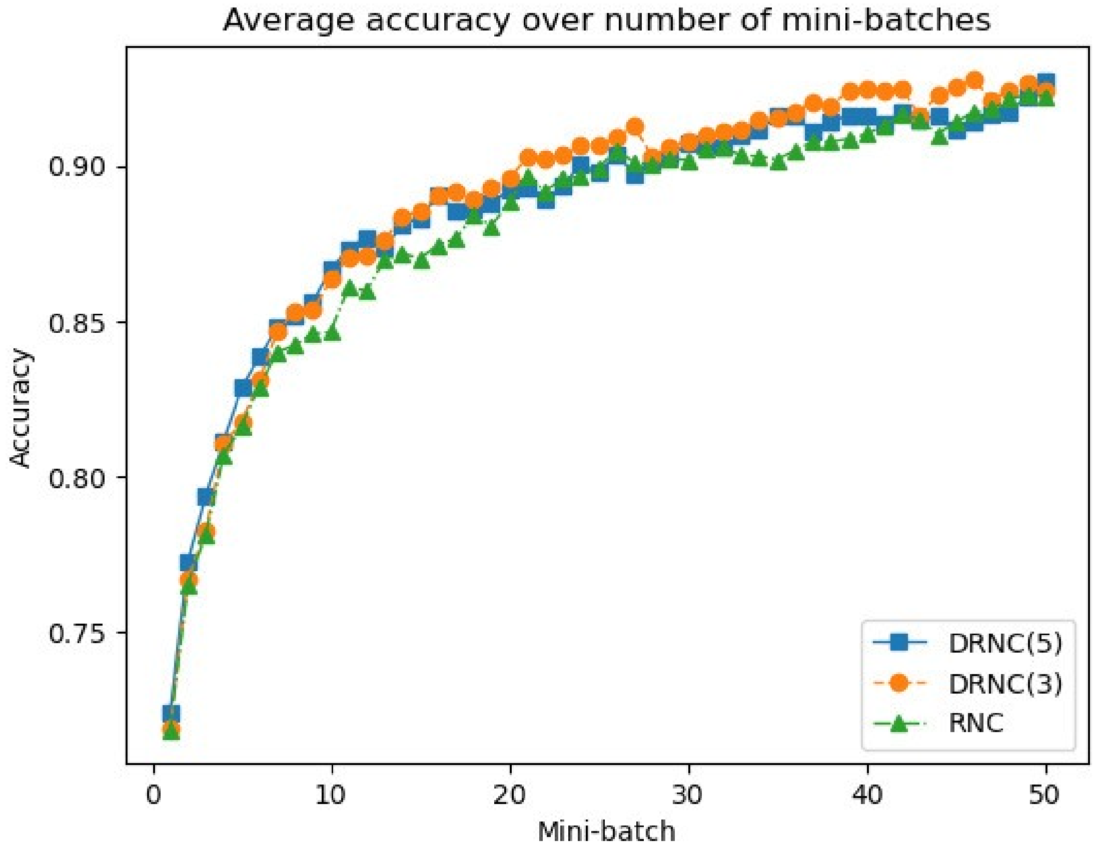
seed %(+)	DRNC(5)	DRNC(3)	RNC	RIPPER	CART
Ø Accuracy	0.9467	0.9502	0.9386	0.9591	*0.9644*
Ø Rank	1.775	1.725	2.5		



- DRNC(3) [DRNC(5)] outperforms RNC on a significance level of more than 95% [90%]

(Beck & Fürnkranz 2021)

Learning Curves (Artificial Datasets)



- DRNC(3) and DRNC(5) converge faster than RNC

Results on Real-World (UCI) Datasets

dataset	%(+)	DRNC(5)	DRNC(3)	RNC	RIPPER	CART
car-evaluation	0.7002	0.8999	**0.9022**	0.8565	*0.9838*	0.9821
connect-4	0.6565	**0.7728**	0.7712	0.7597	0.7475	*0.8195*
kr-vs-kp	0.5222	0.9671	0.9643	**0.9725**	0.9837	*0.989*
monk-1	0.5000	***1***	0.9982	0.9910	0.9478	0.8939
monk-2	0.3428	0.7321	**0.7421**	0.7139	0.6872	*0.7869*
monk-3	0.5199	**0.9693**	0.9603	0.9567	0.9386	*0.9729*
mushroom	0.784	***1***	0.978	0.993	0.9992	*1*
tic-tac-toe	0.6534	0.8956	0.9196	**0.9541**	*1*	0.9217
vote	0.6138	***0.9655***	0.9288	0.9264	0.9011	0.9287
Ø Rank		1.556	2	2.444		

- DRNC(5) has the best performance on these real-world datasets, followed by DRNC(3)

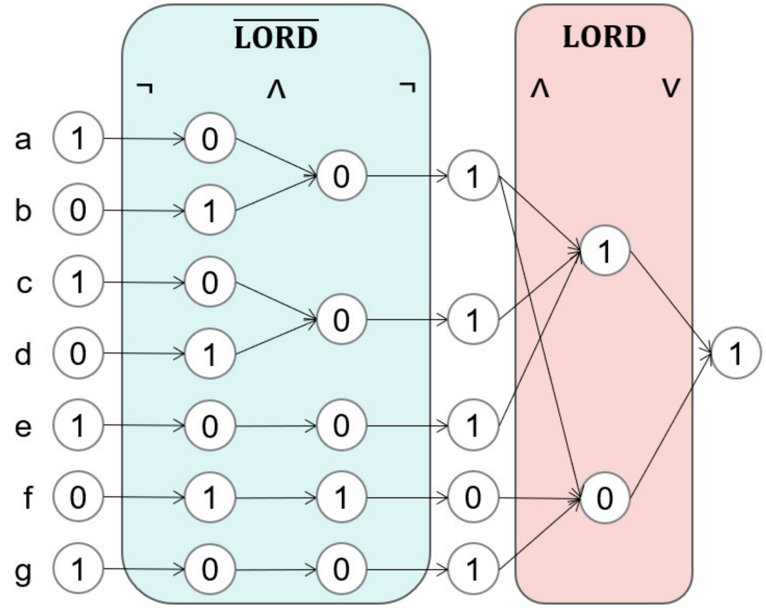
How to Learn Deep Rule Sets

1. The Neural Network Approach
   - fix a network structure and optimize its parameters
2. The Rule Learning Approach
   - layerwise learning of multiple layers of conjunctive and disjunctive rules
     - use conjunctions as input features for CNF learner, and vice versa
   - DNF learners can be used for learning CNF layers

(Beck, Fürnkranz, Huynh 2023)

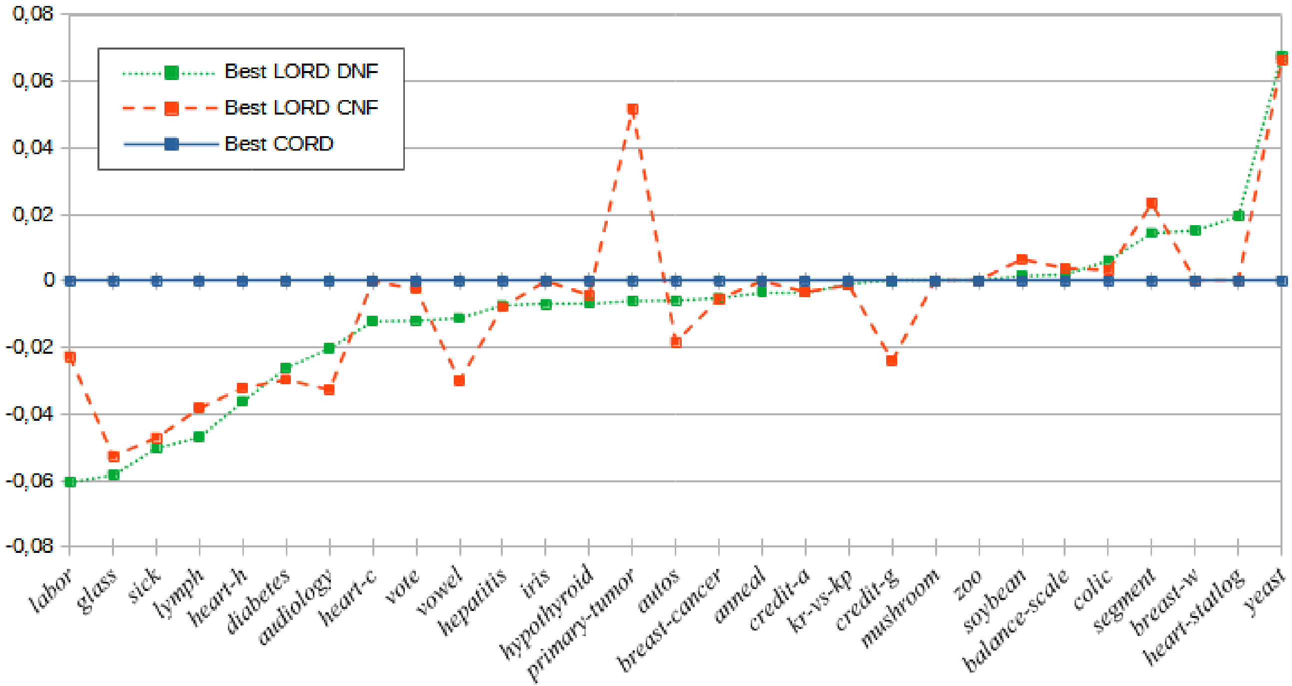
Learning Mixed Conjunctive and Disjunctive Rules

- LORD: A (powerful) conventional rule learner (i.e., DNF learner)
- NegLORD: Learn a CNF by inverting the problem to learn a DNF on the negated classes and negated inputs
- CORD: Allow a combination of conjunctive and disjunctive layers to potentially learn the best of both worlds



Results

- As known from previous works, some concepts can be better learned in CNF, some in DNF
- CORD is in most (but not all) cases better than either



Going Deeper

JOHANNES KEPLER UNIVERSITÄT LINZ

- CORD has 3 layers by default (disj./conj./disj.)
- More layers could be added with the same setup
- Results show modest but not consistent improvements for carefully tuned networks

FROM → TO	2 → 3	2 → 4	2 → 5	3 → 4	3 → 5	4 → 5
# IMPR.	6219	6189	6788	4407	4877	3189
# DET.	5274	5301	6057	4452	5007	3289
% IMPR.	24.75	24.63	27.01	17.54	19.41	12.69
% DET.	20.99	21.09	24.10	17.72	19.92	13.09
VALUES FOR BEST FIVE-LAYERED CORD:						
# IMPR.	126	139	144	86	97	40
# DET.	48	53	52	62	56	17
% IMPR.	43.45	47.93	49.66	29.66	33.45	13.79
% DET.	16.55	18.28	17.93	21.38	19.31	5.86

Analysis of Deeper Networks

- positive and negative correlation of various properties in the conjunctive and disjunctive layers of 5-layer networks with overall accuracy

	CORD				DORC			
	D_1	C_2	D_3	C_4	C_1	D_2	C_3	D_4
m	0.154	0.020	-0.101	-0.131	0.081	0.175	0.019	-0.098
# Rules	-0.189	-0.145	-0.092	-0.043	-0.084	-0.253	-0.134	-0.081
# Concepts	-	0.095	0.045	0.008	-	0.060	0.151	0.074
Avg. Depth	-	0.111	0.057	-0.018	-	0.117	0.159	0.107
Accuracy	0.203	0.520	0.690	-	-0.041	0.342	0.564	-

  - e.g., higher values of the m-parameter (yielding more general rules) are good in early layers, wheras lower values are better in later layers
  - accuracy increases in later layers

How to Learn Deep Rule Sets

JOHANNES KEPLER UNIVERSITÄT LINZ

1. The Neural Network Approach
   - fix a network structure and optimize its parameters
2. The Rule Learning Approach
   - layerwise learning of multiple layers of conjunctive and disjunctive rules
   - DNF learners can be used for learning CNF layers
3. Dedicated Search Algorithm
   - bidirectional search of multiple specializations (selecting conditions) and generalizations (pruning conditions) for learning individual rules did not bring much improvement in the LORD rule learner
     - one layer of specializations + one layer of generalizations is enough
   - ongoing work:
     - evaluate this for incremental constructions of AND/OR networks
       - similar to → (fuzzy) pattern trees (Hüllermeier 2015)

Conclusions

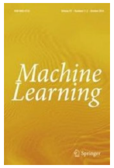
- There are some reasons to believe that deep rule networks may outperform shallow ones (at least in some cases)
- … but there is no convincing evidence yet

→ Deep Rule Learning is a promising topic for further research

- Challenges:
  - Efficient learning algorithms for training intermediate concepts
  - Learning bias for compact structured rule sets
  - Are structured rule sets more interpretable than unstructured rule sets?
  - What would be a killer application for deep rule sets?

References

- Huynh P. V. Q., Fürnkranz, J., Beck, F.: Efficient learning of large sets of locally optimal classification rules. *Machine Learning* 112(2): 571-610 (2023). doi:10.1007/s10994-022-06290-w.
- Beck F., Fürnkranz J.: An Empirical Investigation into Deep and Shallow Rule Learning. *Frontiers in Artificial Intelligence* 4 (2021). doi:10.3389/frai.2021.689398
- Fürnkranz J., Kliegr T., Paulheim H.: On cognitive preferences and the plausibility of rule-based models. *Machine Learning* 109(4): 853-898 (2020) doi:10.1007/s10994-019-05856-5
- Kliegr T., Bahník S., Fürnkranz: A review of possible effects of cognitive biases on interpretation of rule-based machine learning models. *Artificial Intelligence* 295:103458 (2021) doi:10.3389/frai.2021.689398



- Beck F., Fürnkranz J., Huynh P.V.C.: Layerwise Learning of Mixed Conjunctive and Disjunctive Rule Sets. Proceedings of the 7th International Conference on Rules and Reasoning (RuleML) 2023:95-109
- Beck F., Fürnkranz J.: Beyond DNF: On the Incremental Construction of Deep Rule Theories, in *Proceedings of the 22nd Conference Information Technologies - Applications and Theory (ITAT)*, pp. 61--68, 2022.
- Beck F., Fürnkranz J.: An Investigation into Mini-Batch Rule Learning, in *Proceedings of the 2nd Workshop on Deep Continuous-Discrete Machine Learning (DeCoDeML)*, 2020.
- Fürnkranz J., Hüllermeier E., Loza Mencía E., Rapp M.: Learning Structured Declarative Rule Sets – A Challenge for Deep Discrete Learning, in *Proceedings of the 2nd Workshop on Deep Continuous-Discrete Machine Learning (DeCoDeML)*, 2020.
- Fürnkranz J., Kliegr T.: The Need for Interpretability Biases. Proc. IDA 2018: 15-27