Tampereen yliopisto
Tampere University

Explaining with Short Boolean Formulas in Practice

Antti Kuusisto and Tomi Janhunen

(Joint work with Reijo Jaakkola, Masood Rankooh, Miikka Vilander)

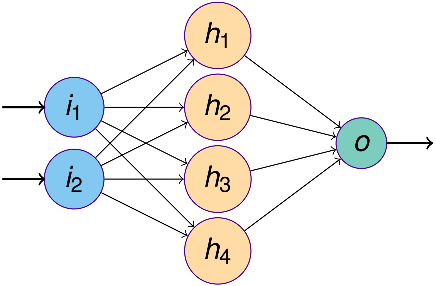
Tampere University, Finland

TU Wien, November 27, 2023

Motivation for the XAILOG Project

- Explainability is a hot topic, e.g., the EU Data Protection Regulation notes the right of individuals to get explanations of automated decisions made about them.
- Explainability via **logic** is underrepresented in the literature.
- In the project, we study explainability
  1. with *explanations* being formulas of different logics while
  2. the targets of explanation, i.e., *classifiers*, may vary:

     formulas, *logic programs*, *neural nets*, *automata*, etc.



Recent Results from XAILOG

- We have concentrated on the Boolean case and studied the effect of *formula length* on the accuracy of explanations / classifiers.
- Following Occam's razor principle (see, e.g., A. Blumer et al., 1987), finding **short** and intuitively **simple** explanations is of great interest.
- This talk is based on two publications (also available at arXiv):

1. R. Jaakkola, T. Janhunen, A. Kuusisto, M. F. Rankooh, and M. Vilander: *Explainability via Short Formulas: the Case of Propositional Logic with Implementation*. In RCRA, 2022.
2. R. Jaakkola, T. Janhunen, A. Kuusisto, M. F. Rankooh, and M. Vilander: *Short Boolean Formulas as Explanations in Practice*. In JELIA, 2023.

Let us start by an introduction to *general* and *special* explanations [1].

Motivation: General Explanations

- The goal is to explain entire classifiers—hence the term “general”.

Data: Boolean assignments in variables $p_1, \ldots, p_k$.

Classifier: A Boolean formula φ.

Explanation: A shortest equivalent formula ψ.

Examples

Classifier	Explanation
$\neg\neg\neg\neg\neg\neg\neg\neg\neg\neg\neg\neg\neg\neg\neg\neg\neg\neg p_1$	p_1
$(p_1 \to p_2) \to ((p_2 \to (p_3 \to p_4)) \to ((p_1 \wedge p_3) \to p_4))$	$\top$

Motivation: Special Explanations

Data: Boolean assignments in variables $p_1, \ldots, p_k$.

Classifier: A Boolean formula φ.

What needs to be explained:

$$\text{the truth value } b \in \{0, 1\} \text{ of } \varphi \text{ under assignment } s.$$

Explanation: the shortest formula ψ such that

1. $\mathit{Truthvalue}(\psi, s) = b$.
2. For any assignment t,

$$\mathit{Truthvalue}(\psi, t) = b \quad \Rightarrow \quad \mathit{Truthvalue}(\varphi, t) = b.$$

(The definition above can be tailored for various logic-based settings.)

Motivation: Special Explanations

Example

Input: the assignment s with $p_1 = 1, p_2 = 0, p_3 = 1$ and the formula

$$p_1 \vee p_2 \vee (p_3 \wedge \neg p_3)$$

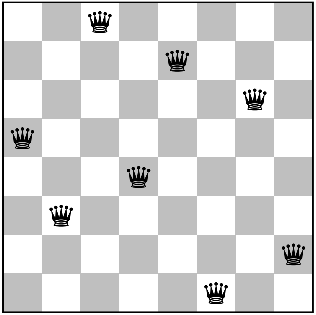
which evaluates to $b = 1$ under s.

Output: p_1

Motivation: Explaining *n*-Queens

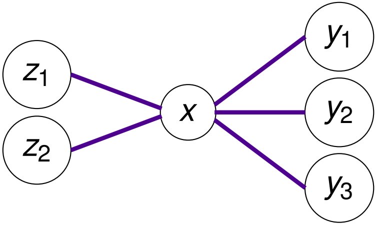
- *Positive instances* are random solutions to the *n*-Qs problem for increasing *n*.
- *Negative instances* are obtained by moving a random queen to a wrong row.

- *Positive explanations* reconstruct solutions.
- *Negative explanations* are misplaced or pairs of threatening queens.



Motivation: Dominating Sets

- Instances are based on *random planar graphs* of varying sizes and their *minimum-size* dominating sets.
- *Negative instances* remove one random vertex from a generated DS; explained by a vertex and its neighbors.
- *Positive instances* mark all vertices in, thus *reconstructing* solutions.



Specification of the DS problem for SATGRND (Gebser et al., JELIA, 2016):

```
vertex(X) :- edge(X,Y).
vertex(Y) :- edge(X,Y).
in(X) | in(Y): edge(X,Y) | in(Z): edge(Z,X) :- vertex(X).
```

Outline

- Main concepts and definitions
- Theoretical results: Ideal classifiers and biases
- Experimental results
- Implementation in ASP
- Related work
- Conclusion

Our Approach in a Nutshell

- We study **explainability** via **short** Boolean formulas.
- The short length guarantees that formulas obtained as explanations are **immediately interpretable** based on the variables involved.
- Our results characterize *quantitatively* why and how **overfitting** arises.
- To find explanations in practice, we
  1. scan through formulas of increasing length ℓ,
  2. for each value ℓ, find a formula that **minimizes the error** among formulas up to the given length, and
  3. stop at the length where overfitting begins.
- The implementation is based on **Answer Set Programming** (ASP).

Tampereen yliopisto
Tampere University

Experimental Setup and Goals

- We study **three data sets** from the UCI machine learning repository.
- For each data set, we find a *very short* Boolean formula that
  - has relatively *low explanation error* but
  - *avoids overfitting*, since any longer formulas would overfit.
- The accuracy of explaining formulas is contrasted with others:
  1. Y. Yang and G. Webb: *Proportional k-interval discretization for naive-Bayes classifiers*. In ECML, 564–575, 2001.
  2. J. Griffith, P. O'Dea, and C. O'Riordan: *A neural net approach to data mining: Classification of users to aid information management*. In Intelligent Exploration of the Web, 389–401, 2003.
  3. B. Ster and A. Dobnikar: *Neural networks in medical diagnosis: Comparison with other methods*. In EANN, 427–430, 1996.
  4. V. Sigillito et al.: *Classification of radar returns from the ionosphere using neural networks*. Johns Hopkins APL Tech. Digest, 10, 262–266, 1989.

Main Concepts and Definitions

Boolean Data Model:

- a set W of data points,
- sets $\{p_1, \ldots, p_k\}$ of attributes, i.e., unary relations $p_i \subseteq W$,
- a target predicate $q \subseteq W$ to be classified and explained such that
- $q \notin \{p_1, \ldots, p_k\}$.

Explanation

- A short Boolean formula φ over $\{p_1, \ldots, p_k\}$ with small error.

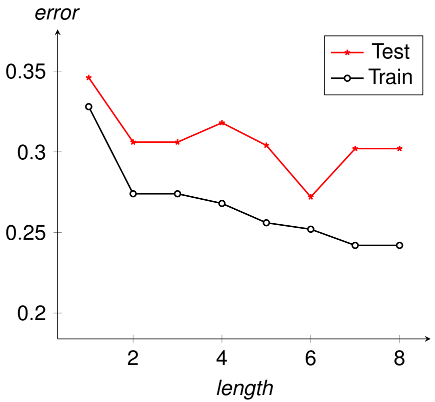
Explanation Error

- The percentage of points that disagree on φ and q over W, i.e.,

$$\frac{|\{w \in W \;:\; w \text{ satisfies } \neg(\varphi \leftrightarrow q)\}|}{|W|}$$

German credit data: Who gets a loan?

- There are 1000 data points with 68 Boolean attributes.



German credit data: Who gets a loan?

We obtained the following *explanation* for positive decisions:

$$\neg\big(\text{negative\_balance} \bigwedge \text{above\_median\_loan\_duration}\big) \bigvee \text{employment\_on\_managerial\_level}.$$

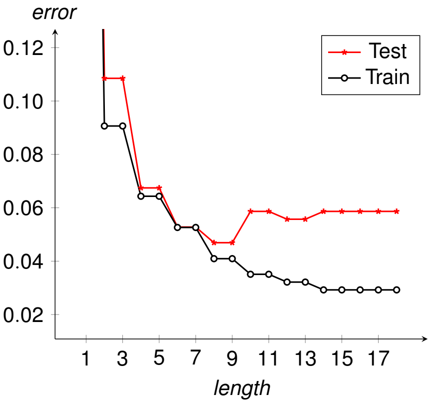
Length: 6
Error: 0.27

In comparison to:

- Naive-Bayes classifiers give error 0.25 (Y. Yang and G. Webb, 2001).
- Neural networks give error 0.24 (J. Griffith et al., 2003).

Breast cancer data: Is a tumor benign?

- There are 683 data points with 9 Boolean attributes.
- The data is based on a **rough** Booleanization.



Breast cancer data: Is a tumor benign?

The explaining formula is

$$\neg(((p \wedge q) \vee r) \wedge s)$$

where p = "thick clump", q = "bare nuclei",
r = "single epithelial cell size", s = "**non-**uniform cell shape".

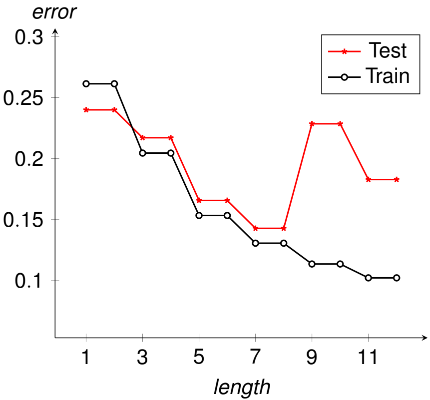
Length: 8
Error: 0.047

In comparison to:

- Naive-Bayes classifiers give error 0.026 (Y. Yang and G. Webb, 2001).
- The best error for several methods—including neural networks—is 0.032 (B. Ster and A. Dobnikar, 1996).

Ionosphere data: Is signal “good” ?

- There are 351 data points with 34 attributes.
- The data results from a **very rough** Booleanization of the original, non-Boolean data set—decreasing expectations on explanations.



Ionosphere data: Is signal “good” ?

The radar data is explained by

$$((p \wedge q) \vee r) \wedge s$$

where the attributes describe numerical signal information.

Length: 7
Error: 0.14

In comparison to:

- Naive-Bayes classifiers give error 0.1 (Y. Yang and G. Webb, 2001).
- Neural networks give error 0.04 (V. Sigillito et al., 1989).

Ideal Classifiers

- Data is sampled from a background probability distribution

$$\mu : \textsc{Propositional assignments} \rightarrow [0, 1].$$

- The distribution μ gives an **ideal theoretical classifier** φ_μ such that
  - φ_μ is the best possible explanation over data sets W sampled from μ, i.e.,
  - φ_μ yields the least possible **expected explanation error** over the class of all datasets W sampled from μ.

- Each data set W gives rise to an **ideal empirical classifier** φ_W.

Bias Gap of Ideal Classifiers

- The ideal empirical classifier φ_W gives the least possible explanation error over W specifically.
- Let $\mathbb{E}_n(\text{ERROR}(\varphi_W))$ denote the *expected value* of the explanation error for formulas φ_W over data sets W of size n sampled from μ.
- In (Jaakkola et al., JELIA, 2023), we show that the **bias gap**

$$|\,\text{ERROR}(\varphi_\mu) - \mathbb{E}_n(\text{ERROR}(\varphi_W))\,| \;\leq\; \frac{1}{2}\sqrt{\frac{2^k}{n}}$$

where k is the size of the vocabulary $\{p_1, \ldots, p_k\}$ used by classifiers.

Estimating Bias Gaps in Practice

- The **upper bound** $\frac{1}{2}\sqrt{2^k/n}$ on bias gap can be used to estimate sufficient sample size $\boldsymbol{n}$ based on the vocabulary size $\boldsymbol{k}$, or vice versa.
- Such considerations may help us to detect overfitting.

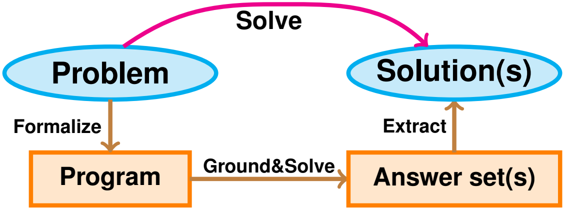
Examples

- For classifiers based on $\boldsymbol{k = 3}$ Boolean variables, a sample size of $\boldsymbol{n = 1000}$ is sufficient to ensure a bias gap of at most **0.045**.
- This is applicable to the Credit data set with **1000** data points.
- For **6400** data points, **6** attributes give a bias gap of at most **0.05**.

Implementation in ASP/ASO

- We implemented the computation of **minimum-size explanations** using an answer-set program that evaluates hypotheses over data.

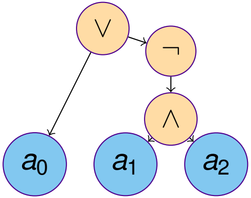
- An **input instance** is a *data matrix* encoded by predicate `val(D,A,V)` where
  - `D` is a data point (*row* in data),
  - `A` is an attribute (*column* in data), and
  - `V` is its binary *value* – either `0` or `1`.




```
1000001001
1111111101
0000011001
1110111101
0001001001
1111111100
0000011001
0010001001
0000000011
0100001001
0000001001
0000001001
1111011100
0000011001
1111111110
1111101100
: : : : : : : : : :
```

Representing the Hypothesis Space

- The hypothesis space is represented in **reverse Polish notation**.

$a_0 \vee \neg(a_1 \wedge a_2) \longmapsto$  $\longmapsto a_2, a_1, \wedge, \neg, a_0, \vee$

- Some **operations** of interest:
  1. Parsing a formula
  2. Reversing the syntax tree (post-order)
  3. Reconstructing the formula (stack)
  4. Evaluating the formula (stack)

Generating Hypothesis Spaces

```
% Domains
#const l=10.
node(1..l).  root(l).  op(neg;and;or).
data(D) :- val(D,A,B).
attr(A) :- val(D,A,B).

% Choose the actual length
{used(N)} :- node(N).
used(N+1) :- used(N), node(N+1).
used(N) :- root(N).

%  Choose leaf nodes and inner nodes, and label them
{leaf(N)} :- used(N).
inner(N) :- used(N), not leaf(N).
{ op(N,O): op(O) } = 1 :- inner(N).
{ lat(N,A): attr(A) } = 1 :- leaf(N).
```

...	lat(5,2)	lat(6,1)	op(7,and)	op(8,neg)	lat(9,0)	op(10,or)

Evaluating Hypotheses over Data

- The order of evaluation is governed by the chosen *sequence* of atoms and connectives and Boolean values present in a *stack*.
- Attributes are false unless marked true in the data set.
- The truth values of other subformulas are chosen but constrained in the standard way (the rules for "`or`" are given below).

```
% Truth values for atoms and subformulas
true(D,N) :- data(D), leaf(N), lat(N,A), val(D,A,1).
{true(D,N)} :- data(D), used(N), inner(N).

% Constraints for disjunctions (as an example)
:- data(D), op(N,or), true(D,N), not true(D,N-1),
   count(N,I), stack(N,I-1,N3), not true(D,N3).
:- data(D), op(N,or), not true(D,N), true(D,N-1).
:- data(D), op(N,or), not true(D,N),
   count(N,I), stack(N,I-1,N2), true(D,N2).
```

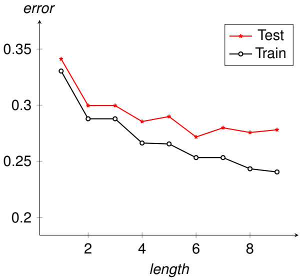
Two-Way Objective Function

- In general, given a data set W, we have no preliminary idea of error rates nor the required formula size.
- We resort to *answer-set optimization* (ASO, Soininen et al., 2002) and deploy an *objective function* that **minimizes**
  - the number of incorrect classifications (with *1st priority*) and
  - the formula size (with *2nd priority*).
- The user is supposed to give an upper bound only: -cl=<number>

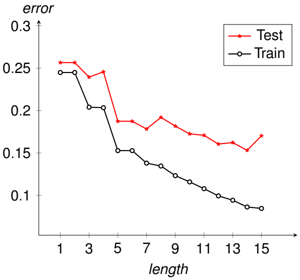
```
error(D) :- data(D), val(D,A,0), expl(A), true(D,N), root(N).
error(D) :- data(D), val(D,A,1), expl(A), not true(D,N), root(N).

#minimize { 1@1,D: error(D);  1@0,N: used(N), node(N) }.
```

Results from UCI ML Data Sets

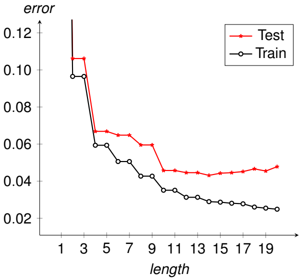
German Credit



Ionosphere



Breast Cancer



Data Set	Rows	Cols	kbits	Error	*l*
German Credit	1000	69	67	0.270	6
Ionosphere	351	34	11	0.140	7
Breast Cancer	683	10	6	0.047	8

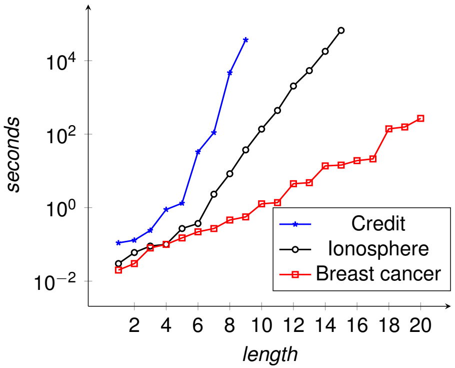
Besides the actual optima, also intermediate formulas may turn out useful!

Performance of CLINGO (v. 5.4.0)

- Hypothesis spaces can be reduced by deleting **symmetric** and **suboptimal** formulas.
- Scales surprisingly well in light of *current-best-hypothesis search* [5].
- We used a model counter (D4) to compute sizes of certain spaces (*without data*):

k	l	$\|HS\|$ ($\times 10^9$)
69	9	17
34	15	634 403
10	20	226 860

Runtime (optimization)



- Contrast with T. Mitchell, *Generalization as search*. AIJ 18(2), 203–226, 1982.

Related Logic-Based Works

- G. Audemard et al.: *On the Computational Intelligibility of Boolean Classifiers*. In KR, 74–86, 2021.
- D. Buchfuhrer and C. Umans: *The complexity of Boolean formula minimization*. JCSS 77(1), 142–153, 2011.
- J. Feldman: *Minimization of Boolean Complexity in Human Learning*. Nature 407(6804), 630–633, 2022.
- J. Quinlan: *Induction of Decision Trees*. Machine Learning 1(1), 81–106, 1986.
- A. Shih, A. Choi, and A. Darwiche: *A Symbolic Approach to Explaining Bayesian Network Classifiers*. In IJCAI, 5103–5111, 2018.

$\Longrightarrow$ In these results, various **normal forms** play a central role.

Conclusion

- Short Boolean formulas can be used as easily *interpretable* explanations and classifiers.
- *Shorter length*
  - increases (intuitive) interpretability,
  - can be necessary to avoid overfitting.
- We provide *new mathematical bounds* for Boolean overfitting.
- Our preliminary experimental results indicate that
  - formulas can provide *natural explanations* in practice;
  - state-of-the art ASP solvers seem to *scale well* for reasonably sized data sets (with matrices up to hundreds of kbits); and
  - conflict-driven learning solvers are able to *prune hypothesis spaces* effectively based on their logical descriptions.
- The results are based on *straightforward Booleanization* of data sets.

Conclusion

Questions / Comments ?