Declarative Combinatorial Optimisation For Machine Learning: Challenges & Opportunities

Mohamed Siala
https://siala.github.io

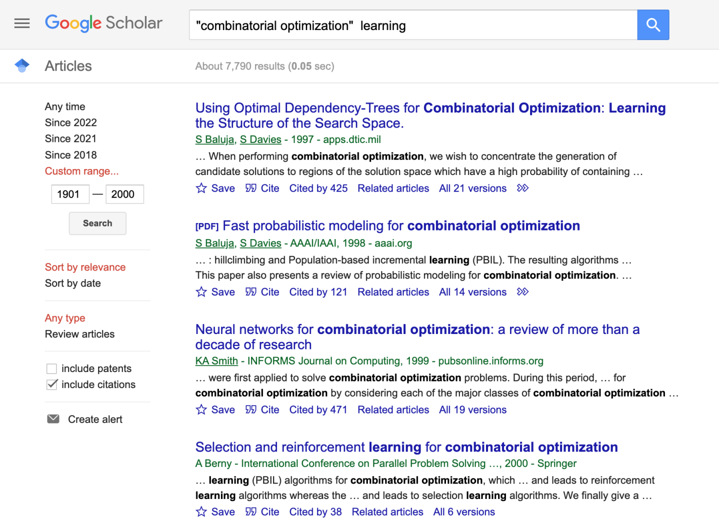
INSA-Toulouse & LAAS-CNRS

July 14, 2022

AI & Modern Decision making

- AI technology ranges from two extremes: Logic-based reasoning as opposed to learning
- Logic-based reasoning breaks down the problem to solve into a deterministic sequence of instructions. This is typically reflected with propositional logic, symbolic AI, expert systems, and yes, combinatorial optimisation
- Learning is about reasoning with pattern matching based on historical observations. This includes models based on neural networks, statistical methods ...
- Modern decision-making applications require both "orthogonal" approaches

Combinatorial Optimistion & Machine Learning


Google Scholar

"combinatorial optimization" learning

Articles

About 7,790 results (**0.05** sec)

Any time
Since 2022
Since 2021
Since 2018
Custom range...
1901 — 2000
Search

Sort by relevance
Sort by date

Any type
Review articles

include patents
include citations

Create alert

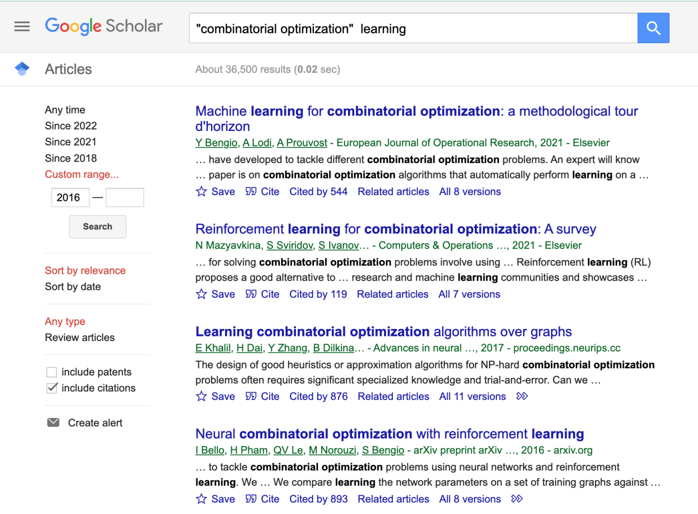
Using Optimal Dependency-Trees for **Combinatorial Optimization**: **Learning** the Structure of the Search Space.
S Baluja, S Davies - 1997 - apps.dtic.mil
… When performing **combinatorial optimization**, we wish to concentrate the generation of candidate solutions to regions of the solution space which have a high probability of containing …
Save Cite Cited by 425 Related articles All 21 versions

[PDF] Fast probabilistic modeling for **combinatorial optimization**
S Baluja, S Davies - AAAI/IAAI, 1998 - aaai.org
… : hillclimbing and Population-based incremental **learning** (PBIL). The resulting algorithms … This paper also presents a review of probabilistic modeling for **combinatorial optimization**. …
Save Cite Cited by 121 Related articles All 14 versions

Neural networks for **combinatorial optimization**: a review of more than a decade of research
KA Smith - INFORMS Journal on Computing, 1999 - pubsonline.informs.org
… were first applied to solve **combinatorial optimization** problems. During this period, … for **combinatorial optimization** by considering each of the major classes of **combinatorial optimization** …
Save Cite Cited by 471 Related articles All 19 versions

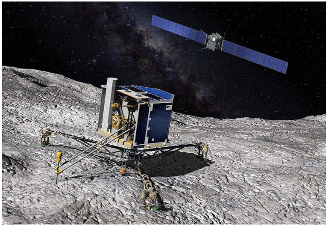
Selection and reinforcement **learning** for **combinatorial optimization**
A Berny - International Conference on Parallel Problem Solving …, 2000 - Springer
… **learning** (PBIL) algorithms for **combinatorial optimization**, which … and leads to reinforcement **learning** algorithms whereas the … and leads to selection **learning** algorithms. We finally give a …
Save Cite Cited by 38 Related articles All 6 versions

Combinatorial Optimistion & Machine Learning



Combinatorial Optimisation

Typical Applications



Solving Methodologies

1. **Adhoc methods**
   - Manually find an algorithm for the specific problem at hand
   - The algorithm can be exact (i.e., with a guarantee of optimality) or heuristic
2. **Declarative Approaches**
   - The unknown of the problems are modeled as decision variables, each associated to a domain (set of values)
   - The problem to solve is stated as a set of constraints to satisfy defined over the variables following a specific language
   - Eventually a utility function (called objective function) to optimise can be part of the problem to solve

Declarative Approaches

Why Declarative Approaches?

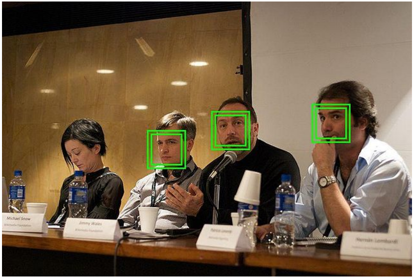
- Adhoc methods are brittle
- Declarative approaches are problem independent! The user models the problem in a specific language and the solver does the job!
- Very active community:
  - Solver competitions: SAT, MaxSAT, SMT, MIP, XCSP, minizinc competitions, ...
  - Benchmarks: CSPLib, MIPLIB, ...
  - Open and commercial Tools: Gurobi, CP Optimizer, OrTools, Chuffed, ...

Examples of Declarative Approaches

- Boolean Satisfiability (SAT)
  - Binary variables
  - The constraints are modelled using clauses (for example $a \vee \neg b \vee c \vee \neg d$)
- (Mixed) Integer Programming (MIP)
  - Integer and/or Continuous Variables
  - The constraints as well as the objective function are linear
- Constraint Programming (CP)
  - Integer, continuous, or sets variables
  - A constraint can be any mathematical relation involving a set of variables
  - The objective function can have different forms

Machine Learning

Typical Applications



Machine Learning

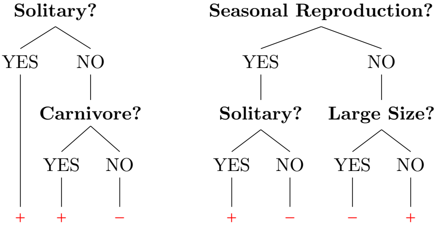
- Machine learning is a computing approach based on learning patterns from historical data
- Input $\Longleftrightarrow$ learning algorithm $\Longleftrightarrow$ ML model $\Longleftrightarrow$ decision making
- Multiple variants exist
  - **Supervised Learning (Labelled data):** Predict a function that associates inputs to outputs based on historical data
  - **Unsupervised Learning:** The task is to figure out patterns presented in the data (unlabelled data)
  - **Reinforcement learning:** Learn from a series of rewards and punishments
  - But also other variants: labelled/non labelled, semi-supervised learning, etc

Supervised ML: Problem Definition

- Given a historical data (**training set**) in the form of input-output examples: $\{(x_1, y_1), \ldots (x_n, y_n)\}$ where x_i is an input, y_i is the output of x_i
- Find a function f_h (called a hypothesis or model) that approximates the true function f
- The approximation criterion can be defined in different ways. We can consider it as a function minimizing some error.
- The hypothesis class is given: for instance decision trees, neural networks, ...

Toy Example: DTs to Predict The Likelihood of Animal Extinction

Big Size	Carnivore	Seasonal Reproduction	Solitary	Extinct
0	1	0	1	yes
1	0	0	1	yes
0	0	0	1	no
1	1	1	0	no
0	0	1	0	yes



- Tabular data
- Hypothesis space: Decision trees
- Left tree: accuracy 2/5
- Right tree: accuracy 2/5

Training Phase

- The function f_h is constructed via a **training algorithm**
- The training algorithm depends on the hypothesis space
- Examples of hypothesis space (family of functions) include polynomial functions, trigonometry functions, decision trees, decision lists, neural networks, ...

Testing Phase

- The evaluation of the constructed hypothesis (or model) is done via a set of unseen examples called **testing set**
- The testing set is usually taken randomly from the initial data. However, it shouldn't be part of the training set
- The common way is to split the initial data into a training and testing sets randomly

Learning Algorithms

- Typical machine learning algorithms are heuristic (no guarantee of optimality): gradient descent, CART, XGBoost, ...
- In many cases, it is hard to tweak the learning algorithm to meet specific requirements (such as fairness or some statistical measures, variants of the hypotheses class, ...)

Machine Learning ⟷ Combinatorial Optimisation

Machine Learning For Combinatorial Optimisation

- Solver tuning based on historical experiences (for instance Bessiere et al. ((2009)))
- Guiding the search space exploration: Reinforcement Learning as an exploration strategy (for instance Antuori et al. ((2021)))
- Handel uncertainty in several contexts such as predict-and-optimise problems and constraint-acquisition Bessiere et al. ((2020)); Mandi et al. ((2020))
- Some other references: Bengio et al. ((2021)); Hoos and Stützle ((2018))

Combinatorial Optimisation for Machine Learning [1]

Meeting specific requirements such as:

- **Robustness:** *Verifying Properties of Binarized Deep Neural Networks.* Narodytska et al., AAAI 2018
- **Fairness:** *Leveraging Integer Linear Programming to Learn Optimal Fair Rule Lists*, Aïvodj et al., CPAIOR'22
- **Privacy:** *Constrained-Based Differential Privacy for Mobility Services* Fioretto et al., **AAMAS 2018**

Combinatorial Optimisation for Machine Learning [2]

Learning with Declarative Approaches such as:

- **CP to learn Decision trees:** *Minimising Decision Tree Size as Combinatorial Optimisation.* Bessiere et al. **CP 2009**
- **CP to learn Decision trees:** *Learning optimal decision trees using constraint programming.* Verhaeghe et al., **Constraints, 2020**
- **SAT to learn NNs:** *In Search for a SAT-friendly Binarized Neural Network Architecture.* Narodytska et al., **ICLR 2020**
- **CP and MIP to learn NNs:** *Training Binarized Neural Networks Using MIP and CP.* Icarte et al., **CP 2019**

Combinatorial Optimisation for Machine Learning [3]

Post-Processing & Decision-making such as:

- **Compression:** *Lossless Compression of Deep Neural Networks.* Serra et al., CPAIOR 2020
- **Explanations:** *Using MaxSAT for Efficient Explanations of Tree Ensembles* Ignatiev et al., AAAI 2022

Combinatorial Optimisation for Fairness

Leveraging Integer Linear Programming to Learn Optimal Fair Rule Lists

Quantifying Fairness [1]

The COMPAS Example ((Angwin et al., 2016))

- Binary classification task: Recidivism within two years
- Sensitive attribute: Ethnicity (African-American/Caucasian)
- Protected Groups:
  - $\mathcal{A}$: African-American individuals;
  - $\mathcal{B}$: Caucasian individuals;

Statistical Fairness

- Principle: ensure that some measure $\mathcal{M}$ *differs by no more than* ϵ between several *protected* groups
- In the particular case of two protected groups ($\mathcal{A}$) and ($\mathcal{B}$), one need to ensure that $|\mathcal{M}(\mathcal{A}) - \mathcal{M}(\mathcal{B})| < \epsilon$

Confusion Matrix

- Consider a data set with 100 individuals: 70 positives and 30 negatives
- The confusion matrix:

	Predicted Positively	Predicted Negatively
True Positive (TP)	65	5
True Negative (TP)	10	20

- True Positive rate (TP): Positive individuals has 93% chance to be correctly predicted
- True Negative rate (TN): Negative individuals has 66% chance to be correctly predicted

Quantifying Fairness [2]

Table 1: Examples of Statistical Fairness Metrics

Metric	Statistical Measure
Statistical Parity (SP) ((Dwork et al., 2012))	Probability of Positive Prediction
Equal Opportunity (EOpp) ((Hardt et al., 2016))	True Positive Rate
Predictive Equality (PE) ((Chouldechova, 2017))	False Positive Rate
Equalized Odds (EO) ((Hardt et al., 2016))	PE and EOpp

Rule Lists

Rule Lists: Definition

Rule lists ((Rivest, 1987)) are classifiers formed by an ordered list of *if-then* rules

Example: The German Credit Dataset

- The task is to predicting whether individuals have a good or bad credit score

IF [gender:female] THEN [good score]
IF [age $\leq$ 25] THEN [bad score]
ELSE [high score]

CORELS & FairCORELS

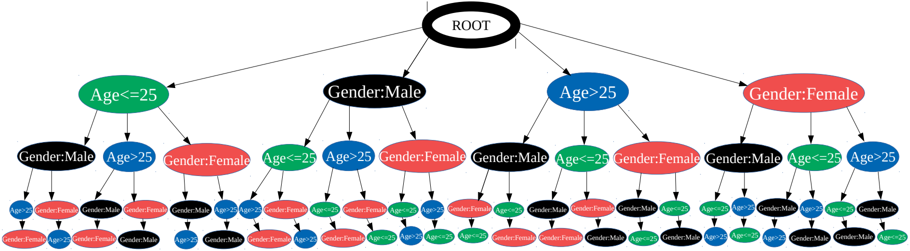


Figure 1: Breadth First Exploration

Pruning Based on an Integer Linear Program

- The idea is to bound the confusion matrix for all possible extensions at each step of the search space by reasoning about both classification and fairness requirements
- The bounding is done using using four discrete variables and two constraints
- If the ILP does not have a solution then no extension can meet both classification and fairness requirements

Example: The ILP model for Equal Opportunity

- Inputs: Prefix δ , dataset $\mathcal{E}$, accuracy lower and upper bounds L and U, unfairness tolerance ϵ
- Variables:

$$x^{TP_{\mathcal{E},p}} \in [TP^{\delta}_{\mathcal{E},p}, |\mathcal{E}^p \cap \mathcal{E}^+| - FN^{\delta}_{\mathcal{E},p}],\ x^{TP_{\mathcal{E},u}} \in [TP^{\delta}_{\mathcal{E},u}, |\mathcal{E}^u \cap \mathcal{E}^+| - FN^{\delta}_{\mathcal{E},u}],$$
$$x^{FP_{\mathcal{E},p}} \in [FP^{\delta}_{\mathcal{E},p}, |\mathcal{E}^p \cap \mathcal{E}^-| - TN^{\delta}_{\mathcal{E},p}],\ x^{FP_{\mathcal{E},u}} \in [FP^{\delta}_{\mathcal{E},u}, |\mathcal{E}^u \cap \mathcal{E}^-| - TN^{\delta}_{\mathcal{E},u}].$$

- Constraints:

Bounding the classification

$$L \leq x^{TP_{\mathcal{E},p}} + x^{TP_{\mathcal{E},u}} + |\mathcal{E}^p \cap \mathcal{E}^-| - x^{FP_{\mathcal{E},p}} + |\mathcal{E}^u \cap \mathcal{E}^-| - x^{FP_{\mathcal{E},u}} \leq U \tag{1}$$

$$-C_3 \leq |\mathcal{E}^p \cap \mathcal{E}^+| \times x^{TP_{\mathcal{E},u}} - |\mathcal{E}^u \cap \mathcal{E}^+| \times x^{TP_{\mathcal{E},p}} \leq C_3 \tag{2}$$

with $C_3 = \epsilon \times |\mathcal{E}^p \cap \mathcal{E}^+| \times |\mathcal{E}^u \cap \mathcal{E}^+|$

Bounding the fairness

Purpose of the Experimental Study

- Is the filtering effective?
- Is filtering helpful to prove optimality ?
- Does the filtering slow down the search space exploration?
- Is it a burden on the memory consumption?
- How about the quality of solutions?

Implementation and Setup I

Integrating our ILP within `FairCORELS`

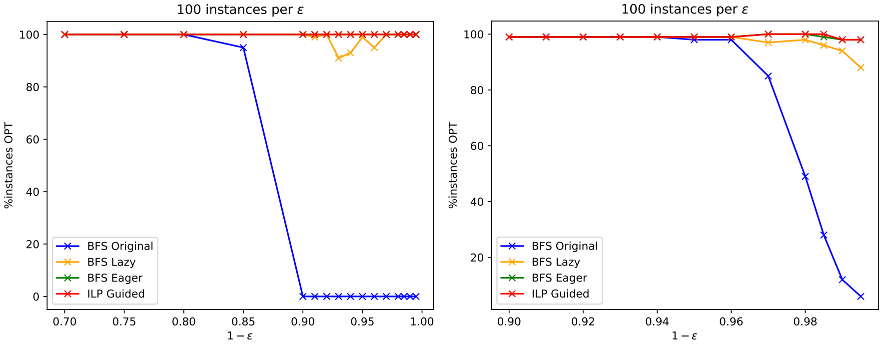
- `ILOG CPLEX 20.10` solver
- Different models
  - `BFS Original`: original `FairCORELS` with a Breadth-First Search (BFS)
  - `BFS Eager`: using a BFS policy, performs the ILP-based pruning **before** inserting a node into the priority queue
  - `BFS Lazy`: using a BFS policy, performs the ILP-based pruning **after** extracting a node from the priority queue
  - `ILP Guided`: best-first search (priority queue ordered by the ILP objectives) with an `Eager` pruning

Implementation and Setup II

We compare the four approaches with the four statistical measures mentioned before using many values for ϵ on 100 randomized runs

- Two datasets:
    - COMPAS ((Angwin et al., 2016))
        - Number of examples: 6150
        - Binary classification task: Recidivism within two years
        - Sensitive attribute: Ethnicity (African-American/Caucasian)
        - Number of binary rules: 18
    - German Credit ((Dua and Graff, 2017))
        - Number of examples: 1000
        - Binary classification task: Good or bad credit score
        - Sensitive attribute: Age (Low/High)
        - Number of binary rules: 49
- Maximum memory use: 4 Gb
- COMPAS: 20 minutes, German Credit 40 minutes
- For each dataset: 100 random different train/test splits

Certifying Optimality I

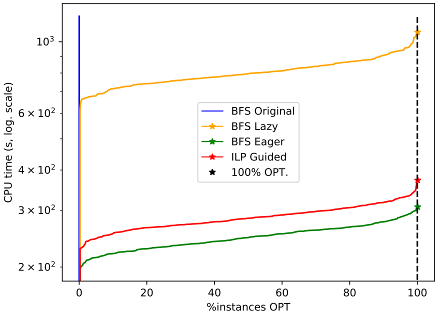


(a) COMPAS dataset

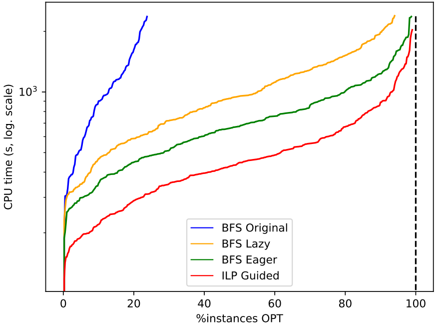
(b) German Credit dataset

Figure 2: Proportion of instances solved to optimality as a function of $1 - \epsilon$.

Certifying Optimality II



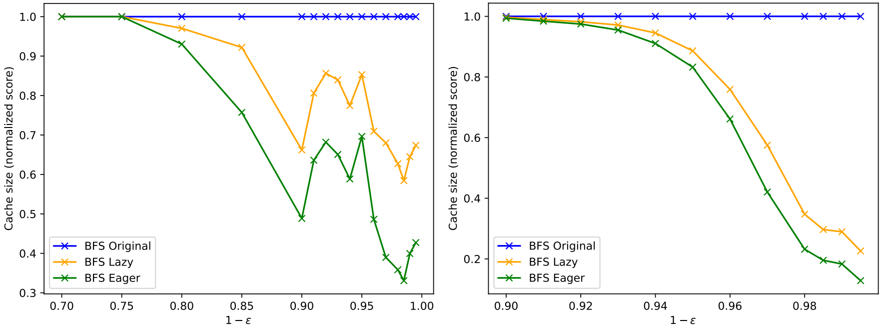
(a) COMPAS dataset



(b) German Credit dataset

Figure 3: CPU time as a function of the proportion of instances solved to optimality, for high fairness requirements (unfairness tolerances ranging between 0.005 and 0.02).

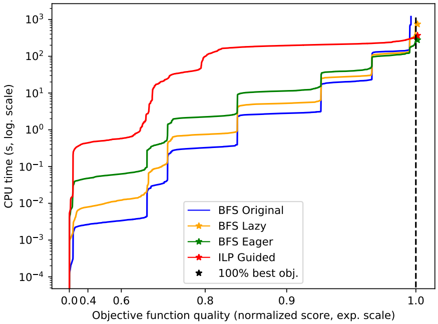
Reducing Priority Queue (Cache) Size



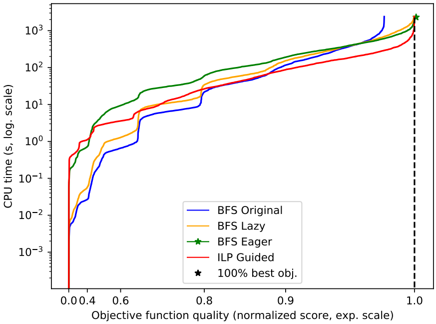
(a) COMPAS dataset (b) German Credit dataset

Figure 4: Relative cache size (#nodes) as a function of $1-\epsilon$ (experiments for the Equal Opportunity fairness metric).

Speeding Up Convergence



(a) COMPAS dataset



(b) German Credit dataset

Figure 5: Solving time as a function of the objective function quality normalized score, for high fairness requirements (unfairness tolerances ranging between 0.005 and 0.02).

Conclusions

- The main idea is to combine accuracy and fairness jointly to prune the search space
- The confusion matrix is bounded effectively thanks to an ILP
- The search space is efficiently boosted on three levels:
  - Finding better solutions quicker (after few seconds)
  - Proofs of optimality
  - Less memory usage

Related Team Work

- Aïvodji et al. ((2022)) *Improving Fairness Generalization Through a Sample-Robust Optimization Method.* Aïvodj et al., **Machine Learning journal, 2022**
- Ferry et al. ((2022)) *Leveraging Integer Linear Programming to Learn Optimal Fair Rule Lists.* Aïvodj et al., **CPAIOR 2022**
- Aïvodji et al. ((2021)) *FairCORELS, an Open-Source Library for Learning Fair Rule Lists.* Aïvodj et al., **CIKM'21**
- Ignatiev et al. ((2020)) *Towards Formal Fairness in Machine Learning.* Ignatiev et al., **CP'20**
- Aïvodji et al. ((2019)) *Learning Fair Rule Lists.* Aïvodj et al., **CoRR abs/1909.03977 (2019)**

Learning via Combinatorial Optimisation

Learning Binary Decision Diagrams (BDD) via MaxSAT

Why BDDs

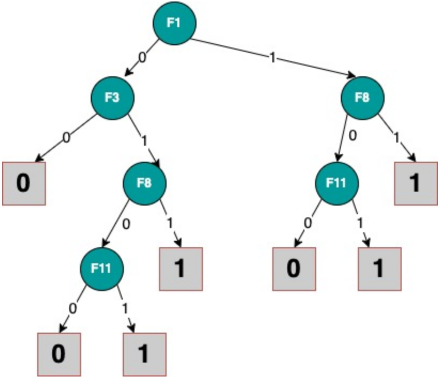


Figure 6: An Example of Decision Tree

Learning Algorithms: The Binary Decision Diagram Example

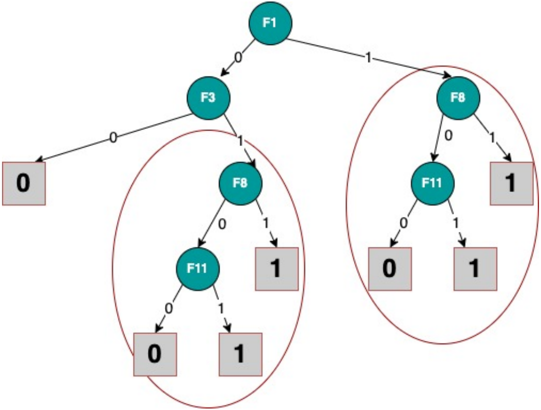


Figure 7: Fragmentation and Redundancy with Decision Tree

An equivalent BDD

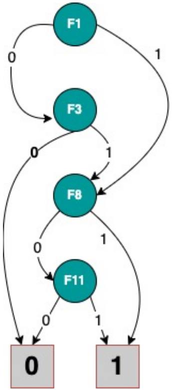


Figure 8: Equivalent Binary Decision Diagram (BDD)

Binary Decision Diagram

- Let $[x_1, \ldots x_n]$ be a sequence of n Boolean variables
- A BDD is a rooted, directed, acyclic graph
- Two types of nodes: terminal and non terminal
- Exactly two terminal nodes labelled with two different values (0 and 1)
- Each non-terminal node is associated to a distinct Boolean Variable x_i
- Each non-terminal node has exactly two children
- **Ordered property:** The variables ordering from any path from the root to a sink node is compatible the order in the sequence $[x_1, \ldots x_n]$
- **Reduced Property:** No isomorphic sub-graphs

Learning BDD Kohavi and Li ((1995))

- Heuristic approach
- Top-Down approach
- The idea of is to build an Oblivious Decision Tree, then merge isomorphic sub-trees
- Hardly flexible to handle additional requirements and properties

Boolean Functions as Strings

- Given $S = [x_1, \dots x_n]$ a sequence of Boolean variables, a Boolean function over S can be represented by a binary string of size 2^n that corresponds to the output of the truth table.
- For instance, with three variables, the string 01100110 represents the following Boolean function:

x_1	x_2	x_3	output
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Beads and BDDs

- A *bead* is a binary string of size 2^n such that the first half is different from the second half
- For instance:
  - $a = 01111101$ is a bead $0111 \neq 1101$
  - $b = 01110111$ is not a bead $0111 = 0111$
- Proposition From Knuth ((2009)) : All vertices in a BDD, are in one-to-one correspondence with the beads of the Boolean function it represents

Maximum Satisfiability (MaxSAT)

- A clause is a disjunction of Boolean variables or their negations. For instance $a \vee \neg b \vee \neg c \vee d$
- A MaxSAT problem is defined by
  - A set of Boolean variables $[x_1, \dots x_n]$
  - A set of Hard clauses to satisfy
  - A set of Soft clauses that can be violated
  - The purpose is to find an assignment of the variables that satisfies all the hard clauses and maximizes the number of satisfied soft clauses

MaxSAT for Learning an Optimal BDD

- Consider a binary dataset with M examples and K features
- The purpose is to learn a BDD of depth H with the maximum accuracy
- The idea is to figure out a sequencing of H features that are used in the desired BDD
- The sequencing of the features with the output string are used to find the beads of the Boolean function
- Once the sequencing and the beads are identified, the BDD is constructed as a post processing step

MaxSAT Model: Variables

Three Sets of Variables

- a_r^i where $r \in [1..K]$ and $i \in [1..H]$ is true iff the feature r is in the position i of the sequence of features
- c_j where $j \in [1..2^H]$ is true iff the j^{th} value of the output string is 1
- d_i^q where $i \in [1..H]$ and $q \in [1..M]$ is true iff for example e_q, the value of the i^{th} feature in the feature ordering is 1

MaxSAT Model: Constraints (1)

1. For each feature r, $\sum_{i=1}^{H} a_r^i \le 1$
2. For each level i, $\sum_{r=1}^{K} a_r^i = 1$
3. The truth table is a bead: $\bigvee_{j=1}^{2^{H-1}} (c_j \oplus c_{j+2^{H-1}})$
4. Consistency w.r.t. examples:
   $\forall q \in [1, M], \forall i \in [1, \dots H], \forall r \in [1, K]$:
   - If the value of f_r is 1 in example e_q then: $a_r^i \rightarrow d_i^q$
   - If the value of f_r is 0 in example e_q then : $a_r^i \rightarrow \neg d_i^q$
5. For each positive example e_q, we have 2^H constraints for classifying examples correctly:

$$\begin{aligned} \neg d_1^q \wedge \neg d_2^q \wedge \cdots \wedge \neg d_{H-1}^q \wedge \neg d_H^q \rightarrow c_1 \\ \neg d_1^q \wedge \neg d_2^q \wedge \cdots \wedge \neg d_{H-1}^q \wedge d_H^q \rightarrow c_2 \\ \dots \\ d_1^q \wedge d_2^q \wedge \cdots \wedge d_{H-1}^q \wedge d_H^q \rightarrow c_{2^H} \end{aligned} \tag{3}$$

6. The same idea is applied for negative examples (with $\neg c_j$)

MaxSAT Model: Constraints (2)

1. HARD: For each feature r, $\sum_{i=1}^{H} a_r^i \leq 1$
2. HARD: For each level i, $\sum_{r=1}^{K} a_r^i = 1$
3. HARD: The truth table is a bead: $\bigvee_{j=1}^{2^{H-1}} (c_j \oplus c_{j+2^{H-1}})$
4. HARD: For each example e_q, $\forall i \in [1, \dots H], \forall r \in [1, K]$:
   - If the value of f_r is 1 in example e_q then: $a_r^i \rightarrow d_i^q$
   - If the value of f_r is 0 in example e_q then : $a_r^i \rightarrow \neg d_i^q$
5. SOFT: For each positive example e_q, we have 2^H constraints for classifying examples correctly:

$$\begin{gathered}
\neg d_1^q \wedge \neg d_2^q \wedge \cdots \wedge \neg d_{H-1}^q \wedge \neg d_H^q \rightarrow c_1 \\
\neg d_1^q \wedge \neg d_2^q \wedge \cdots \wedge \neg d_{H-1}^q \wedge d_H^q \rightarrow c_2 \\
\dots \\
d_1^q \wedge d_2^q \wedge \cdots \wedge d_{H-1}^q \wedge d_H^q \rightarrow c_{2^H}
\end{gathered} \tag{4}$$

6. SOFT: The same idea applies for negative examples (with $\neg c_j$)

Experimental Study

- 15 datasets with different sizes and distributions from CP4IM
  `https://dtai.cs.kuleuven.be/CP4IM/datasets/`
- 15 minutes time limit for the loadra solver
  `https://github.com/jezberg/loandra`
- How does the MaxSAT model compares to OODG ?
- How does the MaxSAT model compares to decision tree models ?
- How to tackle scalability?

MaxSAT Models vs. The Heuristic Approach OODG in Training

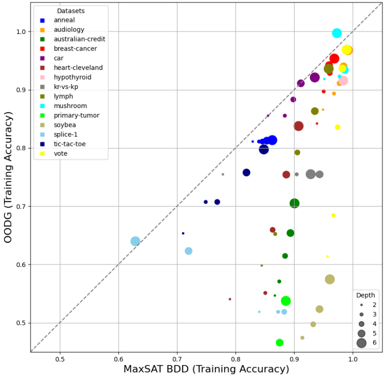


Figure 9: MaxSAT Model vs. OODG : **Better Training Accuracy**

MaxSAT Models vs. The Heuristic Approach OODG in Testing

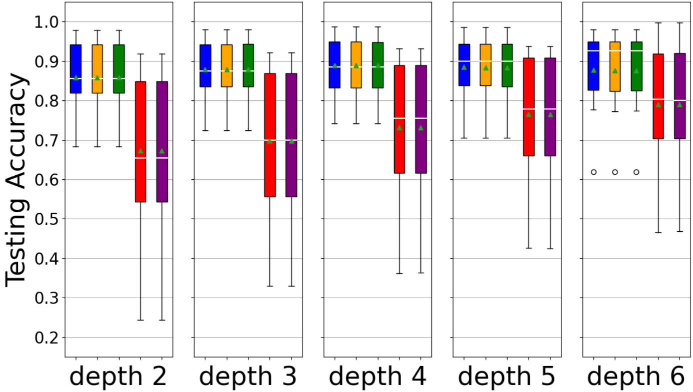


Figure 10: MaxSAT Models vs. OODG in Testing: **Better Generalisation**

MaxSAT BDD vs. MaxSAT Decision Tree Models

Datasets	H	MaxSAT BDD-C				MaxSAT-DT				
		Train	Test	Size	E_Size	Train	Test	Size	E_Size	F_d
anneal	2	82.92	82.19	[illegible]	[illegible]	83.18	82.14	[illegible]	[illegible]	2.88
	3	84	83.55	[illegible]	[illegible]	85.07	84.66	[illegible]	[illegible]	5.76
	4	84.58	83.84	[illegible]	[illegible]	86.05	84.78	[illegible]	[illegible]	8.64
	5	85.33	83.92	[illegible]	71.08	86.44	84.88	[illegible]	[illegible]	11.08
	6	86.26	83.70	[illegible]	99.47	87.6	85.76	[illegible]	[illegible]	17.32
audiology	2	94.91	94.92	4	10.59	95.49	94.92	7	31.35	3
	3	96.78	95.84	[illegible]	16.41	97.82	95.56	[illegible]	88.75	5.28
	4	97.73	95.56	[illegible]	22.56	99.51	94.54	[illegible]	272.15	8.68
	5	98.40	94.44	[illegible]	29.82	99.95	93.98	27	915.29	11.72
	6	99.17	95.84	[illegible]	39.59	99.86	94.08	[illegible]	3323.6	10.88
australian	2	86.70	85.94	4.72	26.79	86.93	85.33	6.68	59.65	2.84
	3	87.45	84.81	5.32	41.15	88.09	84.87	13.08	146.15	5.68
	4	88.45	86.03	7.4	56.85	88.74	85.18	17.48	377.62	7.92
	5	89.36	85.91	10.44	75.9	89.28	84.75	22.52	1076.35	10.08
	6	90.05	85.7	17.32	102.49	89.49	84.84	27.08	3433.64	12.20
cancer	2	93.88	93.59	4	20.29	94.91	94.2	7	45.56	3
	3	95.02	93.91	5.84	31.37	96.6	94.73	15	110.85	6.96
	4	96.06	95.49	7.96	43.89	97.34	94.17	21	283.77	9.44
	5	95.94	93.74	10.68	59.91	97.99	94.35	29.32	800.89	13.20
	6	96.84	94.35	14.8	83.83	98.87	93.41	45.72	2536.91	19.88
car	2	85.53	85.53	4	13.32	85.53	85.53	6.84	32.01	2.92
	3	88.40	87.41	5.08	21.95	89.25	87.45	12.68	71.83	5.64
	4	89.84	88.54	6.84	34.44	91.62	89.68	20.36	162.46	7.68
	5	91.13	89.91	9.6	55.79	93.78	92.77	29.56	389.68	10.24
	6	93.51	92.99	[illegible]	97.06	95.8	95.06	[illegible]	[illegible]	10.88
cleveland	2	79.04	72.57	4	9.48	80.76	72.84	7	25.57	3
	3	85.07	83.37	6	14.73	85.68	76.55	[illegible]	68.93	5.72
	4	86.32	79.46	[illegible]	20.55	86.77	76.75	[illegible]	[illegible]	8.04
	5	88.65	78.72	[illegible]	27.89	87.26	74.45	[illegible]	[illegible]	10.84
	6	90.74	77.29	[illegible]	38.66	88.58	75.81	[illegible]	[illegible]	13.08
hypothyroid	2	97.84	97.84	[illegible]	[illegible]	97.84	97.84	[illegible]	[illegible]	2.48
	3	98.09	98.04	[illegible]	[illegible]	98.14	97.82	[illegible]	[illegible]	4.32
	4	98.27	98.13	[illegible]	[illegible]	98.38	98.01	[illegible]	[illegible]	7.12
	5	98.30	98.05	[illegible]	[illegible]	98.45	98	[illegible]	[illegible]	8.92
	6	98.37	97.95	[illegible]	[illegible]	98.46	97.91	[illegible]	[illegible]	14.04

Figure 11: Lighter Encoding Size

Scalability

- A simple way to tackle scalability is to model the problem on a subset of features
- Pre-Processing using CART to select a subset of (important) features $\mathcal{F}$
- Solve the problem using only $\mathcal{F}$
- Eventually a sampling of the examples can be used to improve scalability

CART, MaxSAT BDD, and Heuristic MaxSAT(1)

Datasets	d	CART				Heuristic MaxSAT-BDD						MaxSAT-BDD					
		Train	Test	Size	F_d	Opt	Train	Test	Size	E_size	Time	Opt	Train	Test	Size	E_size	Time
anneal	2	81.53	81.2	6.12	2.56	100	81.53	81.1	3.56	1.45	0.13	100	82.92	82.1	5	24.09	92.93
	3	81.72	81.3	11.08	4.92	100	81.71	81.3	5.24	4.03	1.64	0	84	83.5	7	37.21	TO
	4	82.60	81.3	18.04	8.40	100	82.57	81.0	7.08	9.64	109.65	0	84.58	83.84	9.40	52.06	TO
	5	84.69	82.2	27.88	12.32	12	84.86	83.37	11.12	20.62	780.08	0	85.33	83.92	11.72	71.08	TO
	6	86.32	84.0	39.80	17	0	86.01	83.74	13.56	42.6	845.72	0	86.26	83.70	14.68	99.47	TO
audiology	2	94.91	94.92	5	2	100	94.91	94.92	4	0.35	0.01	100	94.91	94.92	4	10.59	0.46
	3	97.36	94.82	9	4	100	96.78	95.38	5.04	1	0.02	100	96.78	95.84	5.04	16.41	6.63
	4	98.73	95.37	13.08	6	100	97.73	95.56	7.04	2.27	0.08	100	97.73	95.56	6.96	22.56	56.31
	5	99.42	95.28	17.08	8	100	98.31	95.28	9.76	4.8	0.49	72	98.40	94.44	9.88	29.82	578.99
	6	99.88	95.47	19.08	9	100	98.87	95.84	13	9.78	2.13	48	99.17	95.84	14.28	39.59	613.06
australian	2	86.68	86.62	7	3	100	86.68	86.62	4.92	1.26	0.09	100	86.7	85.94	4.72	26.79	167.99
	3	86.91	84.26	13.08	6	100	86.83	85.09	5.48	3.59	2.41	0	87.45	84.81	5.32	41.15	TO
	4	89.23	85.79	24.92	11.84	84	88.24	85.30	6.8	9.22	536.16	0	88.45	86.03	7.40	56.85	TO
	5	90.9	84.53	41.64	19.24	0	89.27	85.67	10.64	20.28	845.28	0	89.36	85.91	10.44	75.90	TO
	6	92.86	83.24	64.28	28.84	0	89.95	84.47	16.12	41.85	TO	0	90.05	85.7	17.32	102.49	TO
cancer	2	94.5	93.91	7	3	100	93.81	93.59	4	1.32	0.06	100	93.88	93.59	4	20.29	5.89
	3	95.7	94.41	13.24	6.08	100	94.89	94.14	5.64	3.78	0.52	100	95.02	93.91	5.84	31.37	525.59
	4	96.91	94.26	21.08	9.88	100	95.71	94.50	7.8	8.77	20.44	0	96.06	95.49	7.96	43.89	TO
	5	97.83	94.20	30.36	14.04	60	96.35	94.35	10.92	18.32	637.98	0	95.94	93.74	10.68	59.91	TO
	6	98.54	94.38	38.84	17.68	0	96.98	94.67	15.20	36.33	864.36	0	96.84	94.35	14.8	83.83	TO
car	2	85.53	85.53	5	2	100	85.53	85.53	4	2.77	0.14	100	85.53	85.53	4	13.32	24.82
	3	88.5	87.53	7	3	100	88.5	87.53	5	6.94	0.74	8	88.40	87.41	5.08	21.95	TO
	4	89.46	87.86	11	5	100	89.45	87.93	6.4	16.64	14.83	0	89.84	88.54	6.84	34.44	TO
	5	93.88	93.47	18.20	7.80	24	91.34	90.08	9.24	37.43	843.14	0	91.13	89.91	9.60	55.79	TO
	6	94.9	93.37	28.68	10.32	0	93.30	92.65	11.76	79.23	TO	0	93.51	92.99	13.36	97.06	TO
cleveland	2	78.13	72.97	7	2.72	100	77.99	72.43	3.76	0.55	0.04	100	79.04	72.57	4	9.48	83.84
	3	85.68	80.41	15	6.24	100	85.07	84.2	6	1.68	2.28	0	85.07	83.37	6	14.73	TO
	4	88.31	77.09	29.96	13	24	86.15	81.49	7.6	4.46	811.89	0	86.32	79.46	7.84	20.55	TO
	5	92.9	76.82	49.88	21.36	0	88.21	78.84	13.24	9.82	862.94	0	88.65	78.72	13.08	27.89	TO
	6	96.3	74.79	67.80	28.92	0	90.64	78.64	20.2	19.2	TO	0	90.74	77.29	21.04	38.66	TO
hypothyroid	2	97.84	97.84	6.92	2.96	100	97.84	97.84	4	6.2	0.43	100	97.84	97.84	4	92.65	77.76
	3	98.13	97.86	12.84	5.52	100	98.09	97.99	5.16	16.95	4.62	0	98.09	98.04	5.12	142.78	TO
	4	98.39	98.15	22.04	9.80	100	98.28	98.2	6.56	41.23	262.45	0	98.27	98.13	6.72	200.09	TO
	5	98.48	98.04	31.72	14.24	0	98.32	98.07	8.84	87.02	TO	0	98.30	98.05	9.28	274.03	TO
	6	98.6	97.99	43.56	18.92	0	98.37	97.99	13.32	175.62	TO	0	98.37	97.95	13.68	385.40	TO

Figure 12: **Generalisation**

CART, MaxSAT BDD, and Heuristic MaxSAT(2)

Datasets	d	CART				Heuristic MaxSAT-BDD						MaxSAT-BDD					
		Train	Test	Size	F_d	Opt	Train	Test	Size	E_size	Time	Opt	Train	Test	Size	E_size	Time
anneal	2	81.53	81.21	6.12	2.56	100	81.53	81.13	3.56	1.45	0.13	100	82.92	82.19	5	24.09	92.93
	3	81.72	81.38	11.08	4.92	100	81.71	81.33	5.24	4.03	1.64	0	84	83.55	7	37.21	TO
	4	82.60	81.33	18.04	8.40	100	82.57	81.08	7.08	9.64	109.65	0	84.58	83.84	9.40	52.06	TO
	5	84.69	82.29	27.88	12.32	12	84.86	83.37	11.12	20.62	780.08	0	85.33	83.92	11.72	71.08	TO
	6	86.32	84.04	39.80	17	0	86.01	83.74	13.56	42.6	845.72	0	86.26	83.70	14.68	99.47	TO
audiology	2	94.91	94.92	5	2	100	94.91	94.92	4	0.35	0.01	100	94.91	94.92	4	10.59	0.46
	3	97.36	94.82	9	4	100	96.78	95.38	5.04	1	0.02	100	96.78	95.84	5.04	16.41	6.63
	4	98.73	95.37	13.08	6	100	97.73	95.56	7.04	2.27	0.08	100	97.73	95.56	6.96	22.56	56.31
	5	99.42	95.28	17.08	8	100	98.31	95.28	9.76	4.8	0.49	72	98.40	94.44	9.88	29.82	578.99
	6	99.88	95.47	19.08	9	100	98.87	95.84	13	9.78	2.13	48	99.17	95.84	14.28	39.59	613.06
australian	2	86.68	86.62	7	3	100	86.68	86.62	4.92	1.26	0.09	100	86.7	85.94	4.72	26.79	167.99
	3	86.91	84.26	13.08	6	100	86.83	85.09	5.48	3.59	2.41	0	87.45	84.81	5.32	41.15	TO
	4	89.23	85.79	24.92	11.84	84	88.24	85.30	6.8	9.22	536.16	0	88.45	86.03	7.40	56.85	TO
	5	90.9	84.53	41.64	19.24	0	89.27	85.67	10.64	20.28	845.28	0	89.36	85.91	10.44	75.90	TO
	6	92.86	83.24	64.28	28.84	0	89.95	84.47	16.12	41.85	TO	0	90.05	85.7	17.32	102.49	TO
cancer	2	94.5	93.91	7	3	100	93.81	93.59	4	1.32	0.06	100	93.88	93.59	4	20.29	5.89
	3	95.7	94.41	13.24	6.08	100	94.89	94.14	5.64	3.78	0.52	100	95.02	93.91	5.84	31.37	525.59
	4	96.91	94.26	21.08	9.88	100	95.71	94.50	7.8	8.77	20.44	0	96.06	95.49	7.96	43.89	TO
	5	97.83	94.20	30.36	14.04	60	96.35	94.35	10.92	18.32	637.98	0	95.94	93.74	10.68	59.91	TO
	6	98.54	94.38	38.84	17.68	0	96.98	94.67	15.20	36.33	864.36	0	96.84	94.35	14.8	83.83	TO
car	2	85.53	85.53	5	2	100	85.53	85.53	4	2.77	0.14	100	85.53	85.53	4	13.32	24.82
	3	88.5	87.53	7	3	100	88.5	87.53	5	6.94	0.74	8	88.40	87.41	5.08	21.95	TO
	4	89.46	87.86	11	5	100	89.45	87.93	6.4	16.64	14.83	0	89.84	88.54	6.84	34.44	TO
	5	93.88	93.47	18.20	7.80	24	91.34	90.08	9.24	37.43	843.14	0	91.13	89.91	9.60	55.79	TO
	6	94.9	93.37	28.68	10.32	0	93.30	92.65	11.76	79.23	TO	0	93.51	92.99	13.36	97.06	TO
cleveland	2	78.13	72.97	7	2.72	100	77.99	72.43	3.76	0.55	0.04	100	79.04	72.57	4	9.48	83.84
	3	85.68	80.41	15	6.24	100	85.07	84.2	6	1.68	2.28	0	85.07	83.37	6	14.73	TO
	4	88.31	77.09	29.96	13	24	86.15	81.49	7.6	4.46	811.89	0	86.32	79.46	7.84	20.55	TO
	5	92.9	76.82	49.88	21.36	0	88.21	78.84	13.24	9.82	862.94	0	88.65	78.72	13.08	27.89	TO
	6	96.3	74.79	67.80	28.92	0	90.64	78.64	20.2	19.2	TO	0	90.74	77.29	21.04	38.66	TO
hypothyroid	2	97.84	97.84	6.92	2.96	100	97.84	97.84	4	6.2	0.43	100	97.84	97.84	4	92.65	77.76
	3	98.13	97.86	12.84	5.52	100	98.09	97.99	5.16	16.95	4.62	0	98.09	98.04	5.12	142.78	TO
	4	98.39	98.15	22.04	9.80	100	98.28	98.2	6.56	41.23	262.45	0	98.27	98.13	6.72	200.09	TO
	5	98.48	98.04	31.72	14.24	0	98.32	98.07	8.84	87.02	TO	0	98.30	98.05	9.28	274.03	TO
	6	98.6	97.99	43.56	18.92	0	98.37	97.99	13.32	175.62	TO	0	98.37	97.95	13.68	385.40	TO

Figure 13: **Optimality**

CART, MaxSAT BDD, and Heuristic MaxSAT(3)

Datasets	d	CART				Heuristic MaxSAT-BDD						MaxSAT-BDD					
		Train	Test	Size	F_d	Opt	Train	Test	Size	E_size	Time	Opt	Train	Test	Size	E_size	Time
anneal	2	81.53	81.21	6.12	2.56	100	81.53	81.13	3.56	1.4[illegible]	0.[illegible]3	100	82.92	82.19	5	24.[illegible]9	92[illegible]93
	3	81.72	81.38	11.08	4.92	100	81.71	81.33	5.24	4.[illegible]3	1.[illegible]4	0	84	83.55	7	37.[illegible]21	TO
	4	82.60	81.33	18.04	8.40	100	82.57	81.08	7.08	9.[illegible]4	109.[illegible]5	0	84.58	83.84	9.40	52[illegible]06	TO
	5	84.69	82.29	27.88	12.32	12	84.86	83.37	11.12	20[illegible]62	780.[illegible]8	0	85.33	83.92	11.72	71[illegible]08	TO
	6	86.32	84.04	39.80	17	0	86.01	83.74	13.56	4[illegible].6	845.[illegible]2	0	86.26	83.70	14.68	99[illegible].47	TO
audiology	2	94.91	94.92	5	2	100	94.91	94.92	4	0[illegible]35	0.01	100	94.91	94.92	4	1[illegible].59	0.46
	3	97.36	94.82	9	4	100	96.78	95.38	5.04	[illegible]1	0.02	100	96.78	95.84	5.04	1[illegible].41	6.63
	4	98.73	95.37	13.08	6	100	97.73	95.56	7.04	[illegible].27	0.08	100	97.73	95.56	6.96	[illegible]2.56	56.31
	5	99.42	95.28	17.08	8	100	98.31	95.28	9.76	[illegible]4.8	0.49	72	98.40	94.44	9.88	[illegible]9.82	578.9[illegible]
	6	99.88	95.47	19.08	9	100	98.87	95.84	13	[illegible].78	2.13	48	99.17	95.84	14.28	[illegible]9.59	613.06
australian	2	86.68	86.62	7	3	100	86.68	86.62	4.92	[illegible].26	0.09	100	86.7	85.94	4.72	[illegible]6.79	167.99
	3	86.91	84.26	13.08	6	100	86.83	85.09	5.48	[illegible].59	2.41	0	87.45	84.81	5.32	[illegible]1.15	TO
	4	89.23	85.79	24.92	11.84	84	88.24	85.30	6.8	[illegible].22	536.16	0	88.45	86.03	7.40	[illegible]6.85	TO
	5	90.9	84.53	41.64	19.24	0	89.27	85.67	10.64	20.28	845.28	0	89.36	85.91	10.44	75.90	TO
	6	92.86	83.24	64.28	28.84	0	89.95	84.47	16.12	[illegible]1.85	TO	0	90.05	85.7	17.32	[illegible]02.49	TO
cancer	2	94.5	93.91	7	3	100	93.81	93.59	4	1.32	0.06	100	93.88	93.59	4	20.29	5.89
	3	95.7	94.41	13.24	6.08	100	94.89	94.14	5.64	3.78	0.52	100	95.02	93.91	5.84	31.37	525.59
	4	96.91	94.26	21.08	9.88	100	95.71	94.50	7.8	8.77	20.44	0	96.06	95.49	7.96	43.89	TO
	5	97.83	94.20	30.36	14.04	60	96.35	94.35	10.92	18.32	637.98	0	95.94	93.74	10.68	59.91	TO
	6	98.54	94.38	38.84	17.68	0	96.98	94.67	15.20	36.33	864.36	0	96.84	94.35	14.8	83.83	TO
car	2	85.53	85.53	5	2	100	85.53	85.53	4	2.77	0.14	100	85.53	85.53	4	13.32	24.82
	3	88.5	87.53	7	3	100	88.5	87.53	5	6.94	0.74	8	88.40	87.41	5.08	21.95	TO
	4	89.46	87.86	11	5	100	89.45	87.93	6.4	16.64	14.83	0	89.84	88.54	6.84	34.44	TO
	5	93.88	93.47	18.20	7.80	24	91.34	90.08	9.24	37.43	843.14	0	91.13	89.91	9.60	55.79	TO
	6	94.9	93.37	28.68	10.32	0	93.30	92.65	11.76	79.23	TO	0	93.51	92.99	13.36	97.06	TO
cleveland	2	78.13	72.97	7	2.72	100	77.99	72.43	3.76	0.55	0.04	100	79.04	72.57	4	9.48	83.84
	3	85.68	80.41	15	6.24	100	85.07	84.2	6	1.68	2.28	0	85.07	83.37	6	14.73	TO
	4	88.31	77.09	29.96	13	24	86.15	81.49	7.6	4.46	811.89	0	86.32	79.46	7.84	20.55	TO
	5	92.9	76.82	49.88	21.36	0	88.21	78.84	13.24	9.82	862.94	0	88.65	78.72	13.08	27.89	TO
	6	96.3	74.79	67.80	28.92	0	90.64	78.64	20.2	19.2	TO	0	90.74	77.29	21.04	38.66	TO
hypothyroid	2	97.84	97.84	6.92	2.96	100	97.84	97.84	4	6.2	0.43	100	97.84	97.84	4	92.65	77.76
	3	98.13	97.86	12.84	5.52	100	98.09	97.99	5.16	16.95	4.62	0	98.09	98.04	5.12	142.78	TO
	4	98.39	98.15	22.04	9.80	100	98.28	98.2	6.56	41.23	262.45	0	98.27	98.13	6.72	200.09	TO
	5	98.48	98.04	31.72	14.24	0	98.32	98.07	8.84	87.02	TO	0	98.30	98.05	9.28	274.03	TO
	6	98.6	97.99	43.56	18.92	0	98.37	97.99	13.32	[illegible]75.62	TO	0	98.37	97.95	13.68	385.40	TO
	2	77.30	76.35	5	2	100	77.30	76.35	4	5.12	0.55	0	77.83	77.01	4	77.88	TO

Figure 14: **Scalability**

Conclusions

- We proposed exact and heuristic models for learning BDDs thanks to the notion of beads and the flexibility of MaxSAT
- The proposed approach outperforms the existing heuristic approach on many levels (generalisation and proofs of optimality)
- The models that we propose are orders of magnitude lighter than similar models for decision trees
- Our propositions are competitive to state-of-the art decision tree models in terms in generalisation however, they avoid fragmentation and redundancy
- The proposed models are highly flexible to handle different constraints such as the height, specific features restrictions, as well as counting constraints (that might be useful to meet specific requirements such as fairness and balanced predictions, . . .)

Related Team Work

- Hu et al. ((2022)) *Optimizing Binary Decision Diagrams with MaxSAT for classification*. Hu et al., **AAAI'22**
- Hu et al. ((2020)) *Learning Optimal Decision Trees with MaxSAT and its Integration in AdaBoost*. Hu et al., **IJCAI'20**

Take Away Messages

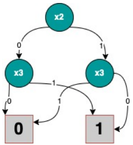
- We have several tools in declarative combinatorial solving that can be used to address many aspects of machine learning
- We live in an exciting research area where two ‘orthogonal’ computing approaches are helping each other
- Modern decision making problems require both combinatorial and learning reasoning
- The challenges are mainly related to formulation and scalability

Thank you!

- The references mentioned in the slides are not exhaustive. They are given as examples
- The work presented here would not be possible without the following amazing researchers (given in a lexicographical order)
    - Ulrich Aïvodji
    - Martin C. Cooper
    - Julien Ferry,
    - Sébastien Gambs
    - Emmanuel Hebrard
    - Hao Hu
    - Marie-José Huguet
    - Alexey Ignatiev
    - João Marques-Silva

Thank you!

Appendix: Beads and BDDs: An Example



- Consider the sequence $[x_1, x_2, x_3]$ with 01100110
- 01100110 is not a bead, so x_1 is discarded and 0110 is considered on the sequence $[x_2, x_3]$
- 0110 is a bead, therefore x_2 is used as a root node
- 01 and 10 are treated separately on the sequence $[x_3]$
- 01 is a bead, therefore a node x_3 is created and an edge (labelled with 0) from x_2 to this node is created
- 10 is a bead, therefore a node x_3 is created and an edge (labelled with 1) from x_2 to this node is created
- Finally, two beads are left (1 and 0) and their correspondent nodes are created similarly

References I

U. Aïvodji, J. Ferry, S. Gambs, M. Huguet, and M. Siala. Learning fair rule lists. *CoRR*, abs/1909.03977, 2019. URL `http://arxiv.org/abs/1909.03977`.

U. Aïvodji, J. Ferry, S. Gambs, M.-J. Huguet, and M. Siala. Faircorels, an open-source library for learning fair rule lists. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 4665–4669, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384469. doi: 10.1145/3459637.3481965.

U. Aïvodji, J. Ferry, S. Gambs, M. Huguet, and M. Siala. Improving fairness generalization through a sample-robust optimization method. *Machine Learning*, 2022.

J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias. *ProPublica, May*, 23, 2016.

References II

V. Antuori, E. Hebrard, M. Huguet, S. Essodaigui, and A. Nguyen. Combining monte carlo tree search and depth first search methods for a car manufacturing workshop scheduling problem. In L. D. Michel, editor, *27th International Conference on Principles and Practice of Constraint Programming, CP 2021, Montpellier, France (Virtual Conference), October 25-29, 2021*, volume 210 of *LIPIcs*, pages 14:1–14:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi: 10.4230/LIPIcs.CP.2021.14. URL `https://doi.org/10.4230/LIPIcs.CP.2021.14`.

Y. Bengio, A. Lodi, and A. Prouvost. Machine learning for combinatorial optimization: A methodological tour d'horizon. *Eur. J. Oper. Res.*, 290(2):405–421, 2021. doi: 10.1016/j.ejor.2020.07.063. URL `https://doi.org/10.1016/j.ejor.2020.07.063`.

References III

C. Bessiere, E. Hebrard, and B. O'Sullivan. Minimising decision tree size as combinatorial optimisation. In I. P. Gent, editor, *Principles and Practice of Constraint Programming - CP 2009, 15th International Conference, CP 2009, Lisbon, Portugal, September 20-24, 2009, Proceedings*, volume 5732 of *Lecture Notes in Computer Science*, pages 173–187. Springer, 2009. doi: 10.1007/978-3-642-04244-7_16. URL `https://doi.org/10.1007/978-3-642-04244-7_16`.

C. Bessiere, C. Carbonnel, A. Dries, E. Hebrard, G. Katsirelos, N. Lazaar, N. Narodytska, C. Quimper, K. Stergiou, D. C. Tsouros, and T. Walsh. Partial queries for constraint acquisition. *CoRR*, abs/2003.06649, 2020. URL `https://arxiv.org/abs/2003.06649`.

References IV

A. Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017.

D. Dua and C. Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226. ACM, 2012.

J. Ferry, U. Aïvodji, S. Gambs, M.-J. Huguet, and M. Siala. Leveraging integer linear programming to learn optimal fair rule lists. In *19th International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, 2022.

References V

M. Hardt, E. Price, N. Srebro, et al. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323, 2016.

H. H. Hoos and T. Stützle. Empirical analysis of randomised algorithms. In T. F. Gonzalez, editor, *Handbook of Approximation Algorithms and Metaheuristics, Second Edition, Volume 1: Methologies and Traditional Applications*, pages 225–242. Chapman and Hall/CRC, 2018. doi: 10.1201/9781351236423-13. URL `https://doi.org/10.1201/9781351236423-13`.

H. Hu, M. Siala, E. Hebrard, and M. Huguet. Learning optimal decision trees with maxsat and its integration in adaboost. In C. Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 1170–1176. ijcai.org, 2020. doi: 10.24963/ijcai.2020/163. URL `https://doi.org/10.24963/ijcai.2020/163`.

References VI

H. Hu, M. Huguet, and M. Siala. Optimizing binary decision diagrams with maxsat for classification. *AAAI*, 2022.

A. Ignatiev, M. C. Cooper, M. Siala, E. Hebrard, and J. Marques-Silva. Towards formal fairness in machine learning. In H. Simonis, editor, *Principles and Practice of Constraint Programming - 26th International Conference, CP 2020, Louvain-la-Neuve, Belgium, September 7-11, 2020, Proceedings*, volume 12333 of *Lecture Notes in Computer Science*, pages 846–867. Springer, 2020. doi: 10.1007/978-3-030-58475-7_49. URL `https://doi.org/10.1007/978-3-030-58475-7_49`.

D. E. Knuth. *The Art of Computer Programming, Volume 4, Fascicle 1: Bitwise Tricks; Techniques; Binary Decision Diagrams.* Addison-Wesley Professional, 12th edition, 2009. ISBN 0321580508.

References VII

R. Kohavi and C. Li. Oblivious decision trees, graphs, and top-down pruning. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Montréal Québec, Canada, August 20-25 1995, 2 Volumes*, pages 1071–1079. Morgan Kaufmann, 1995.

J. Mandi, E. Demirovic, P. J. Stuckey, and T. Guns. Smart predict-and-optimize for hard combinatorial optimization problems. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 1603–1610. AAAI Press, 2020. URL `https://ojs.aaai.org/index.php/AAAI/article/view/5521`.

References VIII

R. L. Rivest. Learning decision lists. *Machine learning*, 2(3):229–246, 1987.