MLIC: A MaxSAT-Based framework for learning interpretable classification rules

Dmitry Malioutov[1] Kuldeep S. Meel[2]

[1]IBM Research, USA
[2]School of Computing, National University of Singapore

CP 2018

The Rise of Artificial Intelligence

- "In Phoenix, cars are self-navigating the streets. In many homes, people are barking commands at tiny machines, with the machines responding. On our smartphones, apps can now recognize faces in photos and translate from one language to another." (New York Times, 2018)
- "AI is the new electricity" (Andrew Ng, 2017)

- Core public agencies, such as those responsible for criminal justice, healthcare, welfare, and education (e.g., "high stakes" domains) should no longer use "black box" AI and algorithmic systems (AI Now Institute, 2018)

The Need for Interpretable Models

- Core public agencies, such as those responsible for criminal justice, healthcare, welfare, and education (e.g., “high stakes” domains) should no longer use “black box” AI and algorithmic systems (AI Now Institute, 2018)
- The practitioners adopt techniques that can be interpreted and validated by them

The Need for Interpretable Models

- Core public agencies, such as those responsible for criminal justice, healthcare, welfare, and education (e.g., "high stakes" domains) should no longer use "black box" AI and algorithmic systems (AI Now Institute, 2018)
- The practitioners adopt techniques that can be interpreted and validated by them
- Medical and education domains see usage of techniques such as classification rules, decision rules, and decision lists.

Prior Work

- Long history of interpretable classification models from data such as decision trees, decision lists, checklists etc with tools such as C4.5, CN2, RIPPER, SLIPPER

Prior Work

- Long history of interpretable classification models from data such as decision trees, decision lists, checklists etc with tools such as C4.5, CN2, RIPPER, SLIPPER
- The problem of learning optimal interpretable models is computationally intractable

Prior Work

- Long history of interpretable classification models from data such as decision trees, decision lists, checklists etc with tools such as C4.5, CN2, RIPPER, SLIPPER
- The problem of learning optimal interpretable models is computationally intractable
- Prior work, which was mostly rooted in late 1980s and 1990s, focused on greedy approaches

Our Approach

Objective Learn rules that are accurate and interpretable. The learning procedure is offline, so learning does not need to happen in real time.

Our Approach

Objective: Learn rules that are accurate and interpretable. The learning procedure is offline, so learning does not need to happen in real time.

Approach:
- The problem of rule learning is inherently an optimization problem
- The past few years have seen *SAT revolution* and development of tools that employ SAT as core engine

Our Approach

Objective Learn rules that are accurate and interpretable. The learning procedure is offline, so learning does not need to happen in real time.

Approach
- The problem of rule learning is inherently an optimization problem
- The past few years have seen *SAT revolution* and development of tools that employ SAT as core engine
- Can we take advantage of *SAT revolution*, in particular progress on MaxSAT solvers?

Key Contributions

- A MaxSAT-based framework, MLIC, that **provably** trades off accuracy vs interpretability of rules
- The prototype implementation is capable of finding optimal (or high quality near-optimal) classification rules from large data sets

Part I

From Rule Learning to MaxSAT

Binary Classification

- Features: $\mathbf{x} = \{x^1, x^2, \cdots x^m\}$
- Input: Set of training samples$\{\mathbf{X}_i, y_i\}$
  - each vector $\mathbf{X}_i \in \mathcal{X}$ contains valuation of the features for sample i,
  - $y_i \in \{0, 1\}$ is the binary label for sample i
- Output: Classifier $\mathcal{R}$, i.e. $y = \mathcal{R}(\mathbf{x})$
- Our focus: classifiers that can be represented as CNF Formulas $\mathcal{R} := C_1 \wedge C_2 \wedge \cdots \wedge C_k$.
- Size of classifiers: $|\mathcal{R}| = \Sigma_i |C_i|$

Input Set of training samples $\{\mathbf{X}_i, y_i\}$

Output Classifier $\mathcal{R}$

- Constraint Learning:

$$\min_{\mathcal{R}} |\mathcal{R}| \quad \text{such that } \mathcal{R}(\mathbf{X}_i) = y_i, \quad \forall i$$

Constraint Learning vs Machine Learning

Input Set of training samples$\{\mathbf{X}_i, y_i\}$

Output Classifier $\mathcal{R}$

- Constraint Learning:

$$\min_{\mathcal{R}} |\mathcal{R}| \quad \text{such that } \mathcal{R}(\mathbf{X}_i) = y_i, \quad \forall i$$

- Machine Learning:

$$\min_{\mathcal{R}} |\mathcal{R}| + \lambda |\mathcal{E}_{\mathcal{R}}| \quad \text{such that } \mathcal{R}(\mathbf{X}_i) = y_i, \quad \forall i \notin \mathcal{E}_{\mathcal{R}}$$

MLIC

- Step 1 Discretization of Features
- Step 2 Transformation to MaxSAT Query
- Step 3 Invoke a MaxSAT Solver and extract $\mathcal{R}$ from MaxSAT solution

Encoding to MaxSAT

Input Features: $\mathbf{x} = \{x^1, x^2, \cdots x^m\}$; Training Data: $\{\mathbf{X}_i, y_i\}$ over m featues

Output $\mathcal{R}$ of k clauses

Key Ideas

- $k \times m$ binary coefficients, denoted by $\{b_1^1, b_1^2, \cdots b_1^m \cdots b_k^m\}$, such that $\mathcal{R}_i = (b_i^1 x^1 \vee b_i^2 x^2 \ldots \vee b_i^m x^m)$
- For every sample i, we have noise variable η_i to encode sample i should be considered as noise or not.

Encoding to MaxSAT

Key Ideas

- $k \times m$ binary coefficients, denoted by $\{b_1^1, b_1^2, \cdots b_1^m \cdots b_k^m\}$, such that $\mathcal{R}_i = (b_i^1 x^1 \vee b_i^2 x^2 \ldots \vee b_i^m x^m)$
- For every sample i, we have noise variable η_i to encode whether sample i should be considered as noise or not.

1. $R = \bigwedge_{l=1}^{k} R_l(\mathbf{x} \mapsto X_i)$: Output of substituting valuation of feature vectors of ith sample

Encoding to MaxSAT

Key Ideas

- $k \times m$ binary coefficients, denoted by $\{b_1^1, b_1^2, \cdots b_1^m \cdots b_k^m\}$, such that $\mathcal{R}_i = (b_i^1 x^1 \vee b_i^2 x^2 \ldots \vee b_i^m x^m)$
- For every sample i, we have noise variable η_i to encode whether sample i should be considered as noise or not.

1. $R = \bigwedge_{l=1}^{k} R_l(\mathbf{x} \mapsto X_i)$: Output of substituting valuation of feature vectors of ith sample
2. $D_i := (\neg\eta_i \rightarrow (y_i \leftrightarrow R(\mathbf{x} \mapsto X_i)))$; $W(D_i) = \top$
   If η_i is False, y_i is equivalent to prediction of the Rule

Encoding to MaxSAT

Key Ideas

- $k \times m$ binary coefficients, denoted by $\{b_1^1, b_1^2, \cdots b_1^m \cdots b_k^m\}$, such that $\mathcal{R}_i = (b_i^1 x^1 \vee b_i^2 x^2 \ldots \vee b_i^m x^m)$
- For every sample i, we have noise variable η_i to encode whether sample i should be considered as noise or not.

1. $R = \bigwedge_{l=1}^{k} R_l(\mathbf{x} \mapsto X_i)$: Output of substituting valuation of feature vectors of ith sample
2. $D_i := (\neg\eta_i \rightarrow (y_i \leftrightarrow R(\mathbf{x} \mapsto X_i))); W(D_i) = \top$
   If η_i is False, y_i is equivalent to prediction of the Rule
3. $V_i^j := (b_i^j); \qquad W\left(V_i^j\right) = 1$
   We want as few b_i^j to be true as possible

Encoding to MaxSAT

Key Ideas

- $k \times m$ binary coefficients, denoted by $\{b_1^1, b_1^2, \cdots b_1^m \cdots b_k^m\}$, such that $\mathcal{R}_i = (b_i^1 x^1 \vee b_i^2 x^2 \ldots \vee b_i^m x^m)$
- For every sample i, we have noise variable η_i to encode whether sample i should be considered as noise or not.

1. $R = \bigwedge_{l=1}^{k} R_l(\mathbf{x} \mapsto X_i)$: Output of substituting valuation of feature vectors of ith sample
2. $D_i := (\neg\eta_i \rightarrow (y_i \leftrightarrow R(\mathbf{x} \mapsto X_i)))$; $W(D_i) = \top$
   If η_i is False, y_i is equivalent to prediction of the Rule
3. $V_i^j := (b_i^j)$; $\quad W\left(V_i^j\right) = 1$
   We want as few b_i^j to be true as possible
4. $N_i := (\eta_i)$; $\quad W(N_i) = \lambda$
   We want as few η_i to be true as possible

Encoding to MaxSAT

1. $R = \bigwedge_{l=1}^{k} R_l(\mathbf{x} \mapsto X_i)$: Output of substituting valuation of feature vectors of ith sample
2. $D_i := (\neg\eta_i \rightarrow (y_i \leftrightarrow R(\mathbf{x} \mapsto X_i))); W(D_i) = \top$
3. $V_i^j := (b_i^j); \qquad W\left(V_i^j\right) = 1$

   We want as few b_i^j to be true as possible
4. $N_i := (\eta_i); \qquad W(N_i) = \lambda$

   We want as few η_i to be true as possible

Construction

Let $Q^k = \bigwedge_i D_i \wedge \bigwedge_i N_i \wedge \bigwedge_{i,j} V_i^j$

$\sigma^* = \text{MaxSAT}(Q^k, W)$, *then* $x^j \in \mathcal{R}_i$ *iff* $\sigma^*(b_i^j) = 1$.

Remember, $\mathcal{R}_i = (b_i^1 x^1 \vee b_i^2 x^2 \ldots \vee b_i^m x^m)$

Theorem (Provable trade off of accuracy vs interpretability of rules)

Let $\mathcal{R}_1 \leftarrow MLIC(\mathbf{X}, \mathbf{y}, k, \lambda_1)$ *and* $\mathcal{R}_2 \leftarrow MLIC(\mathbf{X}, \mathbf{y}, k, \lambda_2)$*, if* $\lambda_2 > \lambda_1$ *then* $|\mathcal{R}_1| \leq |\mathcal{R}_2|$ *and* $|\mathcal{E}_{\mathcal{R}_1}| \geq |\mathcal{E}_{\mathcal{R}_2}|$*.*

Learning DNF Rules

- $(\mathbf{y} = S(\mathbf{x})) \leftrightarrow \neg(y = \neg S(\mathbf{x}))$.
- And if S is a DNF formula, then $\neg S$ is a CNF formula.
- To learn rule S, we simply call MLIC with $\neg\mathbf{y}$ as input and negate the learned rule.

Part II

Experimental Results

Illustrative Example

- Iris Classification:
- Features: sepal length, sepal width, petal length, and petal width
- MLIC learned $\mathcal{R}$:=
  1. (sepal length $>$ 6.3 $\vee$ sepal width $>$ 3.0 $\vee$ petal width $<=$ 1.5) $\wedge$
  2. (sepal width $<=$ 2.7 $\vee$ petal length $>$ 4.0 $\vee$ petal width $>$ 1.2) $\wedge$
  3. (petal length $<=$ 5.0)

Accuracy

Dataset	Size	# Features	RIPPER	Log Reg	NN	RF	SVM	MLIC
TomsHardware	28170	830	0.968 (92.8)	0.976 (0.2)	0.977 (3.4)	0.976 (64.9)	Timeout	0.969 (2000)
Twitter	49990	1050	0.938 (187.3)	0.963 (0.2)	0.965 (6.8)	0.962 (250.9)	0.962 (1010.0)	0.958 (2000)
adult-data	32560	262	0.852 (0.5)	0.801 (0.3)	0.866 (3.0)	0.844 (41.8)	Timeout	0.755 (2000)
credit-card	30000	334	0.811 (0.7)	0.781 (0.1)	0.822 (3.9)	0.82 (25.5)	Timeout	0.82 (2000)
ionosphere	350	564	0.886 (0.1)	0.909 (0.1)	0.926 (1.2)	0.909 (1.3)	0.886 (0.1)	0.889 (15.04)
PIMA	760	134	0.774 (0.1)	0.749 (0.1)	0.764 (1.3)	0.761 (1.3)	0.77 (21.4)	0.736 (2000)
parkinsons	190	392	0.868 (0.1)	0.884 (0.1)	0.921 (1.2)	0.895 (1.1)	0.879 (1.6)	0.895 (245)
Trans	740	64	0.78 (0.0)	0.759 (0.0)	0.788 (1.2)	0.788 (1.2)	0.765 (372.3)	0.797 (1177)
WDBC	560	540	0.961 (0.1)	0.936 (0.0)	0.961 (1.3)	0.943 (1.4)	0.955 (3.0)	0.946 (911)

Intepretability

Dataset	Size	# Features	RIPPER	MLIC
TomsHardware	28170	830	57.5	4
Twitter	49990	1050	78.5	15
adult-data	32560	262	74.5	51.5
credit-card	30000	334	7.5	4
ionosphere	350	564	3	5.5
PIMA	760	134	5	9
parkinsons	190	392	6.5	6
Trans	740	64	6	4

Learning Rate

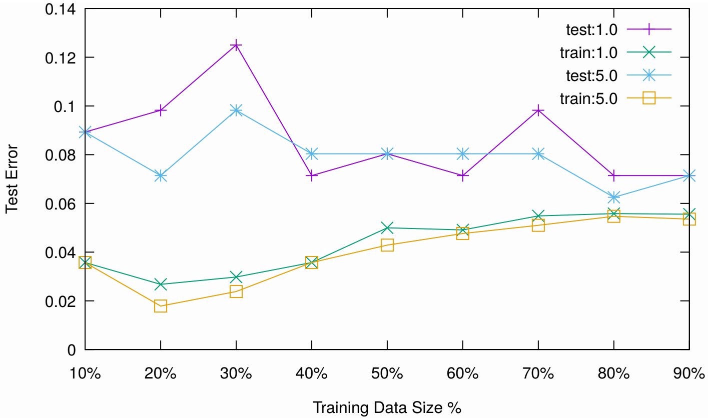

Figure: Plot demonstrating behavior of training and test accuracy vs Size of Training data for WDBC.

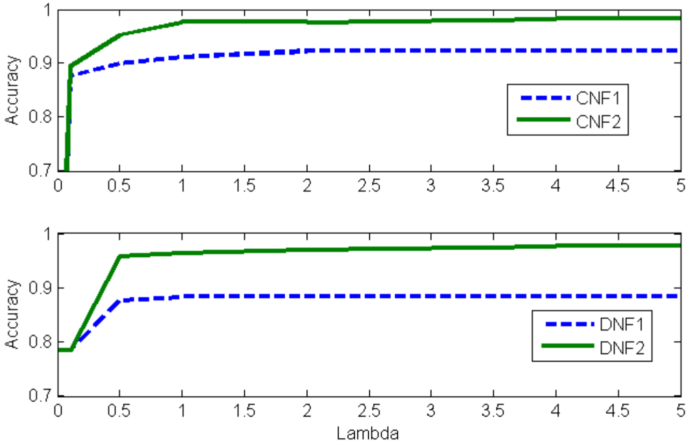

Figure: Plot demonstrating monotone behavior of training accuracy vs λ for CNF and DNF rules with $k = 1$ and 2.

Part III

Conclusion

Summary

- Need for interpretable machine learning systems for usage of AI in core public functions
- The learning task is offline, so allows usage of formal reasoning tools that can provide certificate of correctness
- Long history of prior work: Heuristics to work around combinatorial hardness of optimization problems
- The success of MaxSAT solvers offers opportunity to design techniques with rigorous formal guarantees
- MLIC introduces an approach to use MaxSAT solvers to compute small CNF/DNF rules

Call to the MaxSAT community

Incremental Solving

- The performance of MaxSAT solvers degrade as the problem size increases.
- For training data of size $|D|$ MLIC constructs a query of size $|D| \times k$ to learn k-clause rules

Call to the MaxSAT community

Incremental Solving

- The performance of MaxSAT solvers degrade as the problem size increases.
- For training data of size $|D|$ MLIC constructs a query of size $|D| \times k$ to learn k-clause rules
- State of the art ML techniques learn continuously. Incremental MaxSAT solving? Streaming MaxSAT?

Call to the MaxSAT community

Incremental Solving

- The performance of MaxSAT solvers degrade as the problem size increases.
- For training data of size $|D|$ MLIC constructs a query of size $|D| \times k$ to learn k-clause rules
- State of the art ML techniques learn continuously. Incremental MaxSAT solving? Streaming MaxSAT?

Encodings

- Boolean formulas can express any function.
- That should allow us to learn other popular structures such as decision trees, decision lists etc.

Call to the MaxSAT community

Incremental Solving

- The performance of MaxSAT solvers degrade as the problem size increases.
- For training data of size $|D|$ MLIC constructs a query of size $|D| \times k$ to learn k-clause rules
- State of the art ML techniques learn continuously. Incremental MaxSAT solving? Streaming MaxSAT?

Encodings

- Boolean formulas can express any function.
- That should allow us to learn other popular structures such as decision trees, decision lists etc.
- We need to know about the effect of encodings on MaxSAT problems

Call to the MaxSAT community

Incremental Solving

- The performance of MaxSAT solvers degrade as the problem size increases.
- For training data of size $|D|$ MLIC constructs a query of size $|D| \times k$ to learn k-clause rules
- State of the art ML techniques learn continuously. Incremental MaxSAT solving? Streaming MaxSAT?

Encodings

- Boolean formulas can express any function.
- That should allow us to learn other popular structures such as decision trees, decision lists etc.
- We need to know about the effect of encodings on MaxSAT problems

The area of interpretable machine learning systems will be crucial in the next decade and MaxSAT community can play a central role.

Call to the MaxSAT community

Incremental Solving

- The performance of MaxSAT solvers degrade as the problem size increases.
- For training data of size $|D|$ MLIC constructs a query of size $|D| \times k$ to learn k-clause rules
- State of the art ML techniques learn continuously. Incremental MaxSAT solving? Streaming MaxSAT?

Encodings

- Boolean formulas can express any function.
- That should allow us to learn other popular structures such as decision trees, decision lists etc.
- We need to know about the effect of encodings on MaxSAT problems

The area of interpretable machine learning systems will be crucial in the next decade and MaxSAT community can play a central role.

Multiple postdoc positions and Ph.D. positions available at the National University of Singapore. Remember, Singapore has been rated as the best city in the world to live in. And of course, you get to see sun everyday!