Faithful and Customizable Explanations of Black Box Models

Himabindu Lakkaraju[1], Ece Kamar[2], Rich Caruana[2], Jure Leskovec[3]
[1]Harvard University [2]Microsoft Research [3]Stanford University

AIES 2019

Presenter : Derrick Blakely
https://qdata.github.io/deep2Read

Outline

Outline

ML for High Stakes Decisions

- Medical diagnoses
- Recidivism prediction
- Air quality/pollution models
- Inspiring Big Data stock photos
- Finance
- Etc etc etc

ML for High Stakes Decisions

- Medical diagnoses
- Recidivism prediction
- Air quality/pollution models
- Inspiring Big Data stock photos
- Finance
- Etc etc etc

ML for High Stakes Decisions

- Medical diagnoses
- Recidivism prediction
- Air quality/pollution models
- Inspiring Big Data stock photos
- Finance
- Etc etc etc

ML for High Stakes Decisions

- Medical diagnoses
- Recidivism prediction
- Air quality/pollution models
- Inspiring Big Data stock photos
- Finance
- Etc etc etc

ML for High Stakes Decisions

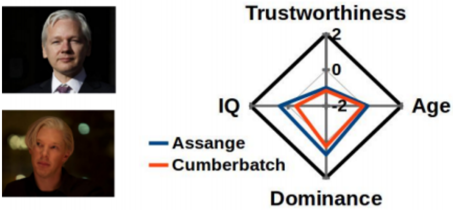
- Medical diagnoses
- Recidivism prediction
- Air quality/pollution models
- Inspiring Big Data stock photos
- Finance
- Etc etc etc

ML for High Stakes Decisions

- Medical diagnoses
- Recidivism prediction
- Air quality/pollution models
- Inspiring Big Data stock photos
- Finance
- Etc etc etc

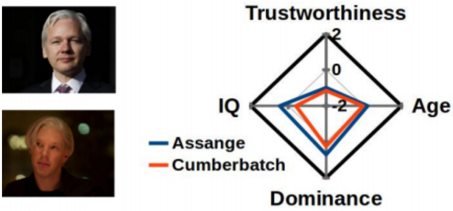
Issues

- Proprietary black box models
- Uninterpretable models (often deep learning)
- Often want explanations of model decisions
- Models should be trustworthy
- Want easy understanding and validation



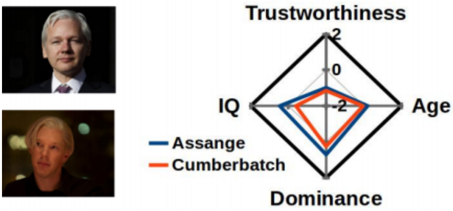
Issues

- Proprietary black box models
- Uninterpretable models (often deep learning)
- Often want explanations of model decisions
- Models should be trustworthy
- Want easy understanding and validation



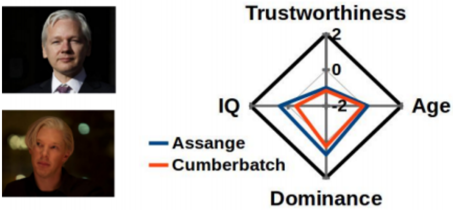
Issues

- Proprietary black box models
- Uninterpretable models (often deep learning)
- Often want explanations of model decisions
- Models should be trustworthy
- Want easy understanding and validation



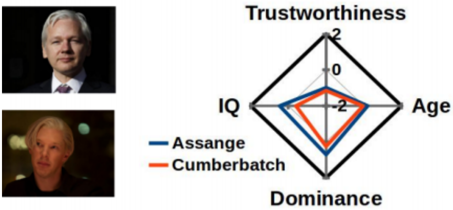
Issues

- Proprietary black box models
- Uninterpretable models (often deep learning)
- Often want explanations of model decisions
- Models should be trustworthy
- Want easy understanding and validation



Issues

- Proprietary black box models
- Uninterpretable models (often deep learning)
- Often want explanations of model decisions
- Models should be trustworthy
- Want easy understanding and validation



The Atlantic   Popular   Latest   Sections   Magazine   More   Subscribe

TECHNOLOGY

A Popular Algorithm Is No Better at Predicting Crimes Than Random People

The COMPAS tool is widely used to assess a defendant's risk of committing more crimes, but a new study puts its usefulness into perspective.

ED YONG JAN 17, 2018

MORE IN THIS SERIES

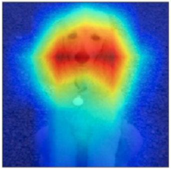
Beyond the age of mass incarceration

The Criminal-Justice Bill Had Broad Bipartisan Support and Still Almost Died

ANDREW KRAGIE

Problems in Explainable AI

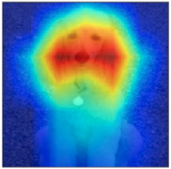
- Fidelity
- Unambiguity
- Interpretability



Cat

Problems in Explainable AI

- Fidelity
- Unambiguity
- Interpretability



Cat

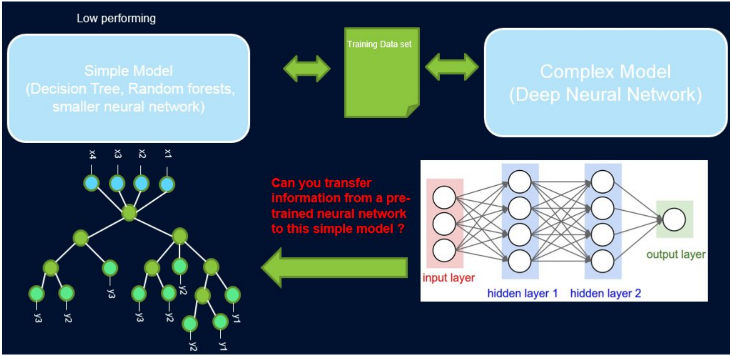
Problems in Explainable AI

- Fidelity
- Unambiguity
- Interpretability

Cat

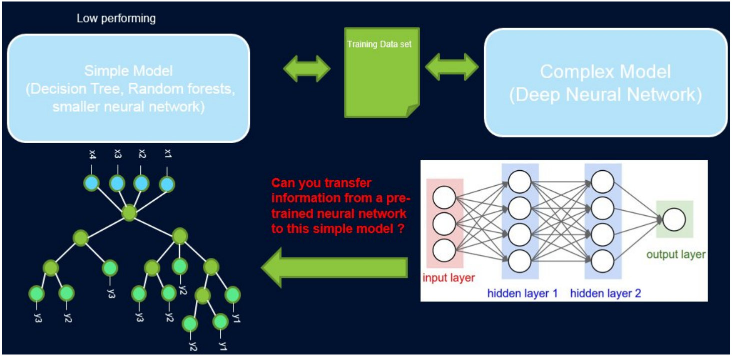
Outline

Related Work



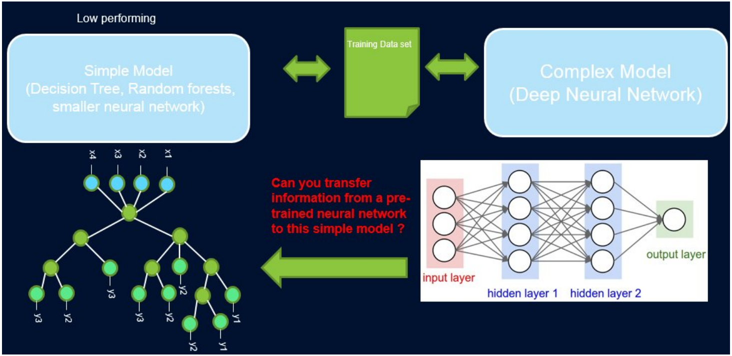
- Logic-based approximations of black box
- Decision trees
- Decision lists
- Decision sets

Related Work



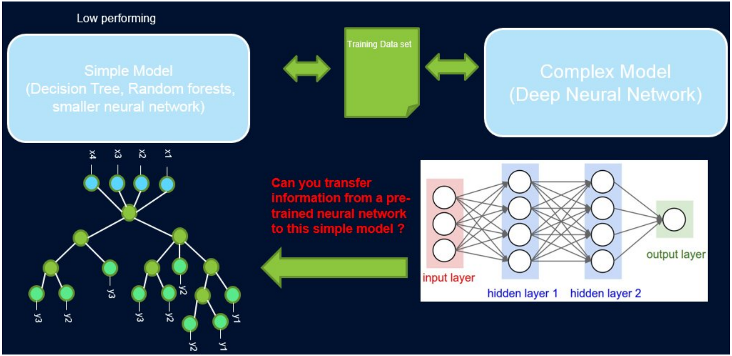
- Logic-based approximations of black box
- Decision trees
- Decision lists
- Decision sets

Related Work



- Logic-based approximations of black box
- Decision trees
- Decision lists
- Decision sets

Related Work



- Logic-based approximations of black box
- Decision trees
- Decision lists
- Decision sets

Related Work Shortcomings

- Explanations are often ambiguous
- Explanations too complicated
- Not customizable
- Don't target an important use-case:
  "How does the model's decision vary based on patient age?"
- Feature *subspaces* matter

Related Work Shortcomings

- Explanations are often ambiguous
- Explanations too complicated
- Not customizable
- Don't target an important use-case:
  "How does the model's decision vary based on patient age?"
- Feature *subspaces* matter

Related Work Shortcomings

- Explanations are often ambiguous
- Explanations too complicated
- Not customizable
- Don't target an important use-case:
  "How does the model's decision vary based on patient age?"
- Feature *subspaces* matter

Related Work Shortcomings

- Explanations are often ambiguous
- Explanations too complicated
- Not customizable
- Don't target an important use-case:
  "How does the model's decision vary based on patient age?"
- Feature *subspaces* matter

Related Work Shortcomings

- Explanations are often ambiguous
- Explanations too complicated
- Not customizable
- Don't target an important use-case:
  "How does the model's decision vary based on patient age?"
- Feature *subspaces* matter

Related Work Shortcomings

- Explanations are often ambiguous
- Explanations too complicated
- Not customizable
- Don't target an important use-case:
  "How does the model's decision vary based on patient age?"
- Feature *subspaces* matter

A bigger shortcoming

Please Stop Explaining Black Box Models for High-Stakes Decisions

Cynthia Rudin
Duke University
cynthia@cs.duke.edu

Abstract

Black box machine learning models are currently being used for high stakes decision-making throughout society, causing problems throughout healthcare, criminal justice, and in other domains. People have hoped that creating methods for explaining these black box models will alleviate some of these problems, but trying to *explain* black box models, rather than creating models that are *interpretable* in the first place, is likely to perpetuate bad practices and can potentially cause catastrophic harm to society. There is a way forward – it is to design models that are inherently interpretable.

Outline

MUSE

- Model Understanding through Subspace Explanations
- "Differentiable explanation": how does model vary across feature subspaces?
- Uses 2-level decision sets

MUSE

- Model Understanding through Subspace Explanations
- "Differentiable explanation": how does model vary across feature subspaces?
- Uses 2-level decision sets

MUSE

- Model Understanding through Subspace Explanations
- "Differentiable explanation": how does model vary across feature subspaces?
- Uses 2-level decision sets

MUSE Output

Subspace descriptor

Decision logic rules

If Age <50 **and** Male $=$Yes:

- **If** Past-Depression $=$Yes **and** Insomnia $=$No **and** Melancholy $=$No, **then** Healthy
- **If** Past-Depression $=$Yes **and** Insomnia $=$Yes **and** Melancholy $=$Yes **and** Tiredness $=$Yes, **then** Depression

If Age $\geq$ 50 **and** Male $=$No:

- **If** Family-Depression $=$Yes **and** Insomnia $=$No **and** Melancholy $=$Yes **and** Tiredness $=$Yes, **then** Depression
- **If** Family-Depression $=$No **and** Insomnia $=$No **and** Melancholy $=$No **and** Tiredness $=$No, **then** Healthy

Default:

- **If** Past-Depression $=$Yes **and** Tiredness $=$No **and** Exercise $=$No **and** Insomnia $=$Yes, **then** Depression
- **If** Past-Depression $=$No **and** Weight-Gain $=$Yes **and** Tiredness $=$Yes **and** Melancholy $=$Yes, **then** Depression
- **If** Family-Depression $=$Yes **and** Insomnia $=$Yes **and** Melancholy $=$Yes **and** Tiredness $=$Yes, **then** Depression

Explanation of the Black Box Model
(No user input)

MUSE Output

Decision logic rules

If Exercise $=$Yes **and** Smoking $=$No:

If Rapid-Weight-Gain $=$Yes **and** Tiredness $=$Yes **and** Melancholy $=$Yes **and** Insomnia $=$Yes **and** Age $<$50, **then** Depression

If Tiredness $=$Yes **and** Melancholy $=$Yes **and** Age $\geq$ 50, **then** Depression

If Tiredness $=$No **and** Melancholy $=$No, **then** Healthy

If Smoking $=$Yes:

If Rapid-Weight-Gain $=$Yes **and** Melancholy $=$Yes, **then** Depression

If Tiredness $=$No **and** Insomnia $=$No **and** Melancholy $=$No **and** Rapid-Weight-Gain $=$No, **then** Healthy

If Insomnia $=$Yes **and** Past-Depression $=$Yes **and** Tiredness $=$Yes, **then** Depression

Default:

If Past-Depression $=$Yes **and** Tiredness $=$Yes **and** Melancholy $=$Yes, **then** Depression

If Past-Depression $=$No **and** Rapid-Weight-Gain $=$Yes **and** Tiredness $=$No **and** Melancholy $=$Yes, **then** Depression

If Family-Depression $=$Yes **and** Age $\geq$ 50 **and** Male $=$No **and** Tiredness $=$Yes, **then** Depression

If Past-Depression $=$No **and** Melancholy $=$No **and** Rapid-Weight-Gain $=$No **and** Tiredness $=$No, **then** Healthy

If Melancholy $=$No **and** Overweight $=$No **and** Insomnia $=$No **and** Tiredness $=$No, **then** Healthy

Explanation of the Black Box Model
w.r.t. **Exercise & Smoking**

MUSE Framework

- Dataset $\mathcal{D} = \{x_1, x_2, ..., x_N\}$, black box $\mathcal{B}$, class labels $\mathcal{C}$
- Goal: create 2-level decision set $\mathcal{R} = \{(q_1, s_1, c_1), ..., (q_M, s_M, c_M)\}$
  - q_i: subspace description; a conjunction
  - s_i: inner logic; a conjunction
  - c_i: label assigned by $\mathcal{R}$
- $\mathcal{ND}$: candidate set of conjunctions; for the generating desctiprs
- $\mathcal{DL}$: same as ND but for inner logic

MUSE Framework

- Dataset $\mathcal{D} = \{x_1, x_2, ..., x_N\}$, black box $\mathcal{B}$, class labels $\mathcal{C}$
- Goal: create 2-level decision set $\mathcal{R} = \{(q_1, s_1, c_1), ..., (q_M, s_M, c_M)\}$
  - q_i: subspace description; a conjunction
  - s_i: inner logic; a conjunction
  - c_i: label assigned by $\mathcal{R}$
- $\mathcal{ND}$: candidate set of conjunctions; for the generating desctiprs
- $\mathcal{DL}$: same as ND but for inner logic

MUSE Framework

- Dataset $\mathcal{D} = \{x_1, x_2, ..., x_N\}$, black box $\mathcal{B}$, class labels $\mathcal{C}$
- Goal: create 2-level decision set $\mathcal{R} = \{(q_1, s_1, c_1), ..., (q_M, s_M, c_M)\}$
  - q_i: subspace description; a conjunction
  - s_i: inner logic; a conjunction
  - c_i: label assigned by $\mathcal{R}$
- $\mathcal{ND}$: candidate set of conjunctions; for the generating desctiprs
- $\mathcal{DL}$: same as ND but for inner logic

MUSE Framework

- Dataset $\mathcal{D} = \{x_1, x_2, ..., x_N\}$, black box $\mathcal{B}$, class labels $\mathcal{C}$
- Goal: create 2-level decision set $\mathcal{R} = \{(q_1, s_1, c_1), ..., (q_M, s_M, c_M)\}$
  - q_i: subspace description; a conjunction
  - s_i: inner logic; a conjunction
  - c_i: label assigned by $\mathcal{R}$
- $\mathcal{ND}$: candidate set of conjunctions; for the generating desctiprs
- $\mathcal{DL}$: same as ND but for inner logic

MUSE Framework

- Dataset $\mathcal{D} = \{x_1, x_2, ..., x_N\}$, black box $\mathcal{B}$, class labels $\mathcal{C}$
- Goal: create 2-level decision set $\mathcal{R} = \{(q_1, s_1, c_1), ..., (q_M, s_M, c_M)\}$
  - q_i: subspace description; a conjunction
  - s_i: inner logic; a conjunction
  - c_i: label assigned by $\mathcal{R}$
- $\mathcal{ND}$: candidate set of conjunctions; for the generating desctiprs
- $\mathcal{DL}$: same as ND but for inner logic

MUSE Framework

- Dataset $\mathcal{D} = \{x_1, x_2, ..., x_N\}$, black box $\mathcal{B}$, class labels $\mathcal{C}$
- Goal: create 2-level decision set $\mathcal{R} = \{(q_1, s_1, c_1), ..., (q_M, s_M, c_M)\}$
  - q_i: subspace description; a conjunction
  - s_i: inner logic; a conjunction
  - c_i: label assigned by $\mathcal{R}$
- $\mathcal{ND}$: candidate set of conjunctions; for the generating desctiprs
- $\mathcal{DL}$: same as ND but for inner logic

MUSE Framework

- Dataset $\mathcal{D} = \{x_1, x_2, ..., x_N\}$, black box $\mathcal{B}$, class labels $\mathcal{C}$
- Goal: create 2-level decision set $\mathcal{R} = \{(q_1, s_1, c_1), ..., (q_M, s_M, c_M)\}$
  - q_i: subspace description; a conjunction
  - s_i: inner logic; a conjunction
  - c_i: label assigned by $\mathcal{R}$
- $\mathcal{ND}$: candidate set of conjunctions; for the generating desctiprs
- $\mathcal{DL}$: same as ND but for inner logic

Label Assignment

- x satisfies some $q_i \wedge s_i \in \mathcal{R} \rightarrow$ label $= c_i$
- x doesn't satisfy any rules in $\mathcal{R} \rightarrow$ assigned with default function
- x satisfies multiple rules $\rightarrow$ one rule is picked via a tie-breaker

Label Assignment

- x satisfies some $q_i \wedge s_i \in \mathcal{R} \rightarrow$ label $= c_i$
- x doesn't satisfy any rules in $\mathcal{R} \rightarrow$ assigned with default function
- x satisfies multiple rules $\rightarrow$ one rule is picked via a tie-breaker

Label Assignment

- x satisfies some $q_i \wedge s_i \in \mathcal{R} \rightarrow$ label $= c_i$
- x doesn't satisfy any rules in $\mathcal{R} \rightarrow$ assigned with default function
- x satisfies multiple rules $\rightarrow$ one rule is picked via a tie-breaker

Quantifying Fidelity

- Simply how often $\mathcal{R}$ agrees with $\mathcal{B}$ over $\mathcal{D}$
- $disagreement(\mathcal{R}) = \sum |\{x | x \in \mathcal{D}, x \text{ satisfies } q_i \wedge s_i, \mathcal{B} \neq c_i\}|$

Quantifying Fidelity

- Simply how often $\mathcal{R}$ agrees with $\mathcal{B}$ over $\mathcal{D}$
- $\mathit{disagreement}(\mathcal{R}) = \sum |\{x | x \in \mathcal{D}, x \text{ satisfies } q_i \wedge s_i, \mathcal{B} \neq c_i\}|$

Quantifying Umambiguity

- Goal 1: prevent rules from overlapping too much
- goal 2: maximize coverage of the rules
- $ruleoverlap(\mathcal{R})$ = number of times a conjunction was repeated in $\mathcal{R}$
- $cover(\mathcal{R})$ = number of x covered by some rule in $\mathcal{R}$

Quantifying Umambiguity

- Goal 1: prevent rules from overlapping too much
- goal 2: maximize coverage of the rules
- $\mathit{ruleoverlap}(\mathcal{R})$ = number of times a conjunction was repeated in $\mathcal{R}$
- $\mathit{cover}(\mathcal{R})$ = number of x covered by some rule in $\mathcal{R}$

Quantifying Umambiguity

- Goal 1: prevent rules from overlapping too much
- goal 2: maximize coverage of the rules
- $ruleoverlap(\mathcal{R})$ = number of times a conjunction was repeated in $\mathcal{R}$
- $cover(\mathcal{R})$ = number of x covered by some rule in $\mathcal{R}$

Quantifying Umambiguity

- Goal 1: prevent rules from overlapping too much
- goal 2: maximize coverage of the rules
- $\textit{ruleoverlap}(\mathcal{R})$ = number of times a conjunction was repeated in $\mathcal{R}$
- $\textit{cover}(\mathcal{R})$ = number of x covered by some rule in $\mathcal{R}$

Quantifying Interpretability

- *size*($\mathcal{R}$) = number of rule triples
- *maxwidth*($\mathcal{R}$) = length of longest rule in number of predicates
- *numpreds*($\mathcal{R}$) = number of predicates in $\mathcal{R}$ (non-unique)
- *numdsets*($\mathcal{R}$) = number of descriptors
- *featuroverlap*($\mathcal{R}$) = overlap of features in descriptors and inner logic

Quantifying Interpretability

- *size*$(\mathcal{R})$ = number of rule triples
- *maxwidth*$(\mathcal{R})$ = length of longest rule in number of predicates
- *numpreds*$(\mathcal{R})$ = number of predicates in $\mathcal{R}$ (non-unique)
- *numdsets*$(\mathcal{R})$ = number of descriptors
- *featuroverlap*$(\mathcal{R})$ = overlap of features in descriptors and inner logic

Quantifying Interpretability

- *size*$(\mathcal{R})$ = number of rule triples
- *maxwidth*$(\mathcal{R})$ = length of longest rule in number of predicates
- *numpreds*$(\mathcal{R})$ = number of predicates in $\mathcal{R}$ (non-unique)
- *numdsets*$(\mathcal{R})$ = number of descriptors
- *featuroverlap*$(\mathcal{R})$ = overlap of features in descriptors and inner logic

Quantifying Interpretability

- $size(\mathcal{R})$ = number of rule triples
- $maxwidth(\mathcal{R})$ = length of longest rule in number of predicates
- $numpreds(\mathcal{R})$ = number of predicates in $\mathcal{R}$ (non-unique)
- $numdsets(\mathcal{R})$ = number of descriptors
- $featuroverlap(\mathcal{R})$ = overlap of features in descriptors and inner logic

Quantifying Interpretability

- $size(\mathcal{R})$ = number of rule triples
- $maxwidth(\mathcal{R})$ = length of longest rule in number of predicates
- $numpreds(\mathcal{R})$ = number of predicates in $\mathcal{R}$ (non-unique)
- $numdsets(\mathcal{R})$ = number of descriptors
- $featuroverlap(\mathcal{R})$ = overlap of features in descriptors and inner logic

Quantified Metrics

Fidelity	$disagreement(\mathcal{R}) = \sum_{i=1}^{M} \lvert\{\boldsymbol{x} \mid \boldsymbol{x} \in \mathcal{D}, \boldsymbol{x} \text{ satisfies } q_i \wedge s_i, \mathcal{B}(\boldsymbol{x}) \neq c_i\}\rvert$
Unambiguity	$ruleoverlap(\mathcal{R}) = \sum_{i=1}^{M} \sum_{j=1, i \neq j}^{M} overlap(q_i \wedge s_i, q_j \wedge s_j)$ $cover(\mathcal{R}) = \lvert\{x \mid x \in \mathcal{D}, x \text{ satisfies } q_i \wedge s_i \text{ where } i \in \{1 \cdots M\}\}\rvert$
Interpretability	$size(\mathcal{R})$: number of rules (triples of the form (q, s, c)) in $\mathcal{R}$ $maxwidth(\mathcal{R}) = \max_{e \in \bigcup_{i=1}^{M} (q_i \cup s_i)} width(e)$ $numpreds(\mathcal{R}) = \sum_{i=1}^{M} width(s_i) + width(q_i)$ $numdsets(\mathcal{R}) = \lvert dset(\mathcal{R})\rvert \text{ where } dset(\mathcal{R}) = \bigcup_{i=1}^{M} q_i$ $featureoverlap(\mathcal{R}) = \sum_{q \in dset(\mathcal{R})} \sum_{i=1}^{M} featureoverlap(q, s_i)$

Setting up the Objective Function

- Goal: maximize each f_i reward function
- W_{max} = max width of any rule in either $\mathcal{ND}$ or $\mathcal{DL}$

$$f_1(\mathcal{R}) = \mathcal{P}_{max} - numpreds(\mathcal{R}), \text{ where } \mathcal{P}_{max} = 2 * \mathcal{W}_{max} * |\mathcal{ND}| * |\mathcal{DL}|$$
$$f_2(\mathcal{R}) = \mathcal{O}_{max} - featureoverlap(\mathcal{R}), \text{ where } \mathcal{O}_{max} = \mathcal{W}_{max} * |\mathcal{ND}| * |\mathcal{DL}|$$
$$f_3(\mathcal{R}) = \mathcal{O'}_{max} - ruleoverlap(\mathcal{R}), \text{ where } \mathcal{O'}_{max} = N \times (|\mathcal{ND}| * |\mathcal{DL}|)^2$$
$$f_4(\mathcal{R}) = cover(\mathcal{R})$$
$$f_5(\mathcal{R}) = \mathcal{F}_{max} - disagreement(\mathcal{R}), \text{ where } \mathcal{F}_{max} = N \times |\mathcal{ND}| * |\mathcal{DL}|$$

Objective Function

Find $\mathcal{R} \subseteq \mathcal{ND} \times \mathcal{DL} \times \mathcal{C}$ to maximize:

$$\sum_{i=1}^{M} \lambda_i f_i(\mathcal{R})$$

subject to:

$$size(\mathcal{R}) \leq \epsilon_1$$

$$maxwidth(\mathcal{R}) \leq \epsilon_2$$

$$numdsets(\mathcal{R}) \leq \epsilon_3$$

λ_i: non-negative weight set by user or found via CV.
ϵ_i: set by user.

Objective Function Optimization

- Optimization is NP-Hard; instance of Budgeted Maximum Coverage Problem
- Use "approximate local search" algo (Lee at al. 2009) for 1/5-approximation
- Gist: select a rule that maximizes the objective; repeatedly perform delete or exchange operations to optimize the solution set

Objective Function Optimization

- Optimization is NP-Hard; instance of Budgeted Maximum Coverage Problem
- Use "approximate local search" algo (Lee at al. 2009) for 1/5-approximation
- Gist: select a rule that maximizes the objective; repeatedly perform delete or exchange operations to optimize the solution set

Objective Function Optimization

- Optimization is NP-Hard; instance of Budgeted Maximum Coverage Problem
- Use “approximate local search” algo (Lee at al. 2009) for 1/5-approximation
- Gist: select a rule that maximizes the objective; repeatedly perform delete or exchange operations to optimize the solution set

Algorithm 1 Optimization Procedure (Lee et al. 2009)

Input: Objective f, domain $\mathcal{ND} \times \mathcal{DL} \times \mathcal{C}$, parameter δ, number of constraints k

$V_1 = \mathcal{ND} \times \mathcal{DL} \times \mathcal{C}$

for $i \in \{1, 2 \cdots k + 1\}$ **do** ▷ Approximation local search procedure

$X = V_i; n = |X|; S_i = \emptyset$

Let v be the element with the maximum value for f and set $S_i = v$

while there exists a delete/update operation which increases the value of S_i by a factor of at least $(1 + \frac{\delta}{n^4})$ **do**

Delete Operation: If $e \in S_i$ such that $f(S_i \backslash \{e\}) \geq (1 + \frac{\delta}{n^4}) f(S_i)$, then $S_i = S_i \backslash e$

Exchange Operation If $d \in X \backslash S_i$ and $e_j \in S_i$ (for $1 \leq j \leq k$) such that

$(S_i \backslash e_j) \cup \{d\}$ (for $1 \leq j \leq k$) satisfies all the k constraints and

$f(S_i \backslash \{e_1, e_2 \cdots e_k\} \cup \{d\}) \geq (1 + \frac{\delta}{n^4}) f(S_i)$, then $S_i = S_i \backslash \{e_1, e_2, \cdots e_k\} \cup \{d\}$

end while

$V_{i+1} = V_i \backslash S_i$

end for

return the solution corresponding to $\max\{f(S_1), f(S_2), \cdots f(S_{k+1})\}$

Outline

Datasets

1. Bail outcomes (released on bail or not) for 86K defendants
2. High school performance (graduated on time or not) for 21K students
3. Depression diagnoses (depressed or not) 33K patients

Datasets

1. Bail outcomes (released on bail or not) for 86K defendants
2. High school performance (graduated on time or not) for 21K students
3. Depression diagnoses (depressed or not) 33K patients

Datasets

1. Bail outcomes (released on bail or not) for 86K defendants
2. High school performance (graduated on time or not) for 21K students
3. Depression diagnoses (depressed or not) 33K patients

Baselines

- Locally Interpretable Model agnostic Explanations (LIME) (Ribeiro, Singh, and Guestrin 2016)
- Interpretable Decision Sets (IDS) (Lakkaraju, Bach, and Leskovec 2016)
- Bayesian Decision Lists (BDL) (Letham et al. 2015)

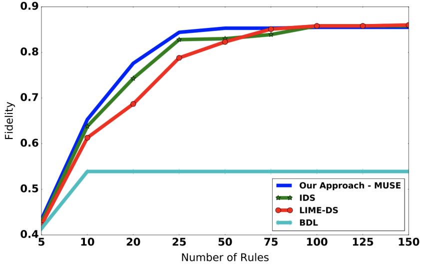
Baselines

- Locally Interpretable Model agnostic Explanations (LIME) (Ribeiro, Singh, and Guestrin 2016)
- Interpretable Decision Sets (IDS) (Lakkaraju, Bach, and Leskovec 2016)
- Bayesian Decision Lists (BDL) (Letham et al. 2015)

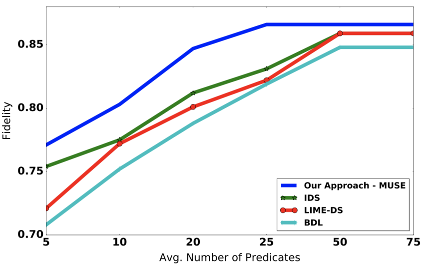
Baselines

- Locally Interpretable Model agnostic Explanations (LIME) (Ribeiro, Singh, and Guestrin 2016)
- Interpretable Decision Sets (IDS) (Lakkaraju, Bach, and Leskovec 2016)
- Bayesian Decision Lists (BDL) (Letham et al. 2015)

Results



(a) Number of Rules



(b) Avg. Number of Predicates

Results

Approach	Human Accuracy	Avg. Time (in secs.)
MUSE (No customization)	94.5%	160.1
IDS	89.2%	231.1
BDL	83.7%	368.5
MUSE (Customization)	98.3%	78.3

(c) Results of User Study

Outline

1. Background
2. Related Work
3. The MUSE Framework
4. Results
5. Conclusion

Shortcomings

- Can't work on image classifiers; needs to be combined with feature extraction from middle layers of NN
- What if we value some features more than others?
- Asks end users to do a lot of work
  - create $\mathcal{D}, \mathcal{NL}$, and $\mathcal{DL}$ sets
  - Set objective function weights
  - Set ϵ constraint values
- Is 85% tolerable in high stakes situations?
- Possibly encourages bad practice
- Might be better as an analysis tool for ML developers

Shortcomings

- Can't work on image classifiers; needs to be combined with feature extraction from middle layers of NN
- What if we value some features more than others?
- Asks end users to do a lot of work
    - create $\mathcal{D}, \mathcal{NL}$, and $\mathcal{DL}$ sets
    - Set objective function weights
    - Set ϵ constraint values
- Is 85% tolerable in high stakes situations?
- Possibly encourages bad practice
- Might be better as an analysis tool for ML developers

Shortcomings

- Can't work on image classifiers; needs to be combined with feature extraction from middle layers of NN
- What if we value some features more than others?
- Asks end users to do a lot of work
  - create $\mathcal{D}, \mathcal{NL},$ and $\mathcal{DL}$ sets
  - Set objective function weights
  - Set ϵ constraint values
- Is 85% tolerable in high stakes situations?
- Possibly encourages bad practice
- Might be better as an analysis tool for ML developers

Shortcomings

- Can't work on image classifiers; needs to be combined with feature extraction from middle layers of NN
- What if we value some features more than others?
- Asks end users to do a lot of work
  - create $\mathcal{D}, \mathcal{NL},$ and $\mathcal{DL}$ sets
  - Set objective function weights
  - Set ϵ constraint values
- Is 85% tolerable in high stakes situations?
- Possibly encourages bad practice
- Might be better as an analysis tool for ML developers

Shortcomings

- Can't work on image classifiers; needs to be combined with feature extraction from middle layers of NN
- What if we value some features more than others?
- Asks end users to do a lot of work
  - create $\mathcal{D}, \mathcal{NL},$ and $\mathcal{DL}$ sets
  - Set objective function weights
  - Set ϵ constraint values
- Is 85% tolerable in high stakes situations?
- Possibly encourages bad practice
- Might be better as an analysis tool for ML developers

Shortcomings

- Can't work on image classifiers; needs to be combined with feature extraction from middle layers of NN
- What if we value some features more than others?
- Asks end users to do a lot of work
  - create $\mathcal{D}, \mathcal{NL},$ and $\mathcal{DL}$ sets
  - Set objective function weights
  - Set ϵ constraint values
- Is 85% tolerable in high stakes situations?
- Possibly encourages bad practice
- Might be better as an analysis tool for ML developers

Shortcomings

- Can't work on image classifiers; needs to be combined with feature extraction from middle layers of NN
- What if we value some features more than others?
- Asks end users to do a lot of work
  - create $\mathcal{D}, \mathcal{NL},$ and $\mathcal{DL}$ sets
  - Set objective function weights
  - Set ϵ constraint values
- Is 85% tolerable in high stakes situations?
- Possibly encourages bad practice
- Might be better as an analysis tool for ML developers

Shortcomings

- Can't work on image classifiers; needs to be combined with feature extraction from middle layers of NN
- What if we value some features more than others?
- Asks end users to do a lot of work
  - create $\mathcal{D}, \mathcal{NL},$ and $\mathcal{DL}$ sets
  - Set objective function weights
  - Set ϵ constraint values
- Is 85% tolerable in high stakes situations?
- Possibly encourages bad practice
- Might be better as an analysis tool for ML developers

Shortcomings

- Can't work on image classifiers; needs to be combined with feature extraction from middle layers of NN
- What if we value some features more than others?
- Asks end users to do a lot of work
    - create $\mathcal{D}, \mathcal{NL},$ and $\mathcal{DL}$ sets
    - Set objective function weights
    - Set ϵ constraint values
- Is 85% tolerable in high stakes situations?
- Possibly encourages bad practice
- Might be better as an analysis tool for ML developers

Lessons Learned

- Potentially good idea to build an interpretable approximation of your model using logic rules
- Valuable for sanity checking or helping others use model
- More work is needed on interpretable black box algorithms

Lessons Learned

- Potentially good idea to build an interpretable approximation of your model using logic rules
- Valuable for sanity checking or helping others use model
- More work is needed on interpretable black box algorithms

Lessons Learned

- Potentially good idea to build an interpretable approximation of your model using logic rules
- Valuable for sanity checking or helping others use model
- More work is needed on interpretable black box algorithms