Explanations for Monotonic Classifiers

Joao Marques-Silva[1] Thomas Gerspacher[1]
Martin Cooper[1] Alexey Ignatiev[2] Nina Narodytska[3]

1. CNRS/IRIT, University of Toulouse

2. Monash University, Australia

3. VMWare, USA

ICML 2021

Monotonic classifiers

Feature domains & the set of classes assumed totally ordered.

Definition

A classifier κ is *monotonic* if $a \leq b \Rightarrow \kappa(a) \leq \kappa(b)$ (where, given two feature vectors a, b, $a \leq b$ if $a_i \leq b_i$ $(i = 1, \ldots, n)$).

Monotonic classifiers

Feature domains & the set of classes assumed totally ordered.

Definition

A classifier κ is *monotonic* if $a \leq b \Rightarrow \kappa(a) \leq \kappa(b)$ (where, given two feature vectors a, b, $a \leq b$ if $a_i \leq b_i$ $(i = 1, \ldots, n)$).

Example

A student is accepted on a CS Masters course if $\kappa = 1$, where

$$\kappa = (CS \vee M \vee EE) \wedge (X \geq 60 \vee W \geq 1) \wedge (P + A + OR \geq 2)$$

where CS, M, EE indicates whether they have a degree in CS, Maths, EEng; X is the final exam mark, W is years of work experience; P, A, OR indicate whether they have taken classes in Programming, Algorithmics, OR.

Clearly, κ is monotonic (increasing any feature cannot decrease the value of κ).

Explanations of a specific decision

We want to explain a specific decision $\kappa(v) = c$ by giving a set of features which are important for this decision.

Definition

A prime implicant/abductive explanation (*AXp*) is a minimal set of features that are sufficient to explain the decision $\kappa(v) = c$.

Example

An AXp of $\kappa(1, 0, 0, 65, 1.5, 1, 1, 0) = 1$ is $\{CS, X, P, A\}$

Explanations of a specific decision

We want to explain a specific decision $\kappa(v) = c$ by giving a set of features which are important for this decision.

Definition

A prime implicant/abductive explanation (*AXp*) is a minimal set of features that are sufficient to explain the decision $\kappa(v) = c$.

Example

An AXp of $\kappa(1, 0, 0, 65, 1.5, 1, 1, 0) = 1$ is $\{CS, X, P, A\}$

Definition

A contrastive explanation (*CXp*) is a minimal set of features which, if changed, can lead to a change of class.

Example

CXp's of $\kappa(1, 0, 0, 65, 1.5, 1, 1, 0) = 1$: $\{CS\}$, $\{X, W\}$, $\{P\}$, $\{A\}$.

Finding one explanation

Proposition

It is possible to find one AXp (CXp) in polynomial time

```
**findOneAXp** $(v, c)$ :
    $\mathcal{S} \leftarrow \{1, \ldots, n\}$ ;   $v_L \leftarrow v$ ;
    **for** $i = 1, \ldots, n$ :
        fix $i$th feature in $v_L$ to lowest value in domain ;
        **if** $\kappa(v_L) = c$
        **then** $\mathcal{S} \leftarrow \mathcal{S} \setminus \{i\}$ :
        **else** reinstate previous value of $v_L$ ;
    **return** $\mathcal{S}$ ;
```

Hitting-set duality of AXp's and CXp's

Proposition

Every AXp intersects every CXp.

Proposition

$\exists$ an algorithm to enumerate all AXp's and all CXp's which requires 1 call to a SAT oracle per explanation (AXp or CXp).

For example, a new AXp must satisfy the constraints:

- intersect all already-found CXps
- not be a subset of any already-found AXp

and any set satisfying these constraints is a superset of a new AXp (which can be found by a version of **findOneAXp**).

Experiments and Conclusion

Experiments

- *Average size of explanations is short for both AXp's and CXp's.*
- *Average runtime is almost entirely taken up by calls to the classifier which shows that despite NP-completeness, the SAT oracle is very fast.*
- *Compared to Anchor, our approach produces shorter explanations on average, is faster (approx. 5 times faster) due to the lower number of calls to the classifier, and provides formal guarantees.*

Experiments and Conclusion

Experiments

- *Average size of explanations is short for both AXp's and CXp's.*
- *Average runtime is almost entirely taken up by calls to the classifier which shows that despite NP-completeness, the SAT oracle is very fast.*
- *Compared to Anchor, our approach produces shorter explanations on average, is faster (approx. 5 times faster) due to the lower number of calls to the classifier, and provides formal guarantees.*

Conclusion

We have an **efficient** *method for finding* **formally-correct** *explanations if the classifier is* **monotonic**.