Towards Trustworthy AI

Namrita Varshney

with Prof. Ashutosh Gupta, Prof S. Akshay, Prof S. Krishna, Prof Kaushik Mallik*, A. Ahmed, Tanay Tayal
IIT Bombay, SBI Foundation Hub project, *IMDEA Spain

AI for mission-critical systems

AI is being deployed everywhere, including within mission-critical systems.

- Examples: airport security, loan dispersal, self-driving car, online medical advice

AI for mission-critical systems

AI is being deployed everywhere, including within mission-critical systems.

- Examples: airport security, loan dispersal, self-driving car, online medical advice

AI makes mistakes.

AI for mission-critical systems

AI is being deployed everywhere, including within mission-critical systems.

- Examples: airport security, loan dispersal, self-driving car, online medical advice

AI makes mistakes.

We need quality assurance for AI.

Outline

- Deep inspection of AI model
  - Robustness property
  - Sensitivity property
  - A novel property

AI Model

$$\text{Input} \rightarrow \text{AI Model} \rightarrow \text{Output}$$

AI Model

$$\text{Input} \rightarrow \text{AI Model} \rightarrow \text{Output}$$

AI model is a function that approximates the relationship of input and output

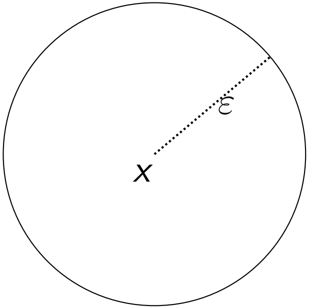
Deep inspection of AI models

- Check models for robustness, adversarial attacks, sensitivity, data poisoning, and fairness

Deep inspection of AI models

- Check models for robustness, adversarial attacks, sensitivity, data poisoning, and fairness

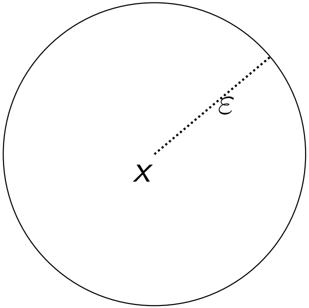
Robustness



Deep inspection of AI models

- Check models for robustness, adversarial attacks, sensitivity, data poisoning, and fairness

Robustness

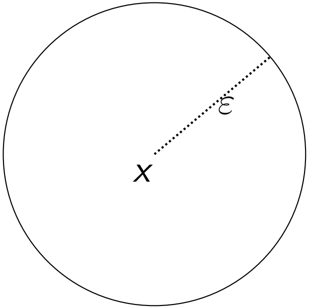


$$B_\varepsilon(x) = \{\, z \mid \|z - x\| = \varepsilon \,\} \; \forall z \in B_\varepsilon(x),\; f(x) = f(z)$$

Deep inspection of AI models

- Check models for robustness, adversarial attacks, sensitivity, data poisoning, and fairness

Robustness



$$B_\varepsilon(x) = \{\, z \mid \|z - x\| = \varepsilon \,\} \; \forall z \in B_\varepsilon(x),\; f(x) = f(z)$$

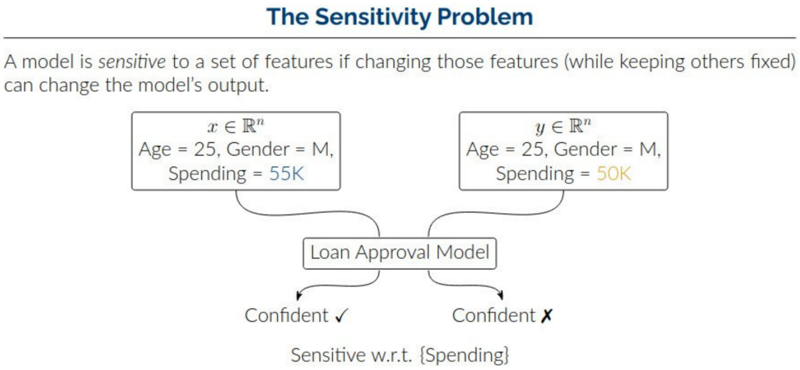
A lot of work has been done in the literature.

Deep inspection of AI models

- Check models for robustness, adversarial attacks, sensitivity, data poisoning, and fairness

Deep inspection of AI models

- Check models for robustness, adversarial attacks, sensitivity, data poisoning, and fairness



Deep inspection of AI models

- Check models for robustness, adversarial attacks, sensitivity, data poisoning, and fairness

Deep inspection of AI models

- Check models for robustness, adversarial attacks, sensitivity, data poisoning, and fairness

We will discuss a newly discovered anomaly.

Topic 1.1

Sensitivity property

Sensitivity of a loan dispersal model

Intuitive description: a small set of features can alter the decision.

In other words, all decisions of the model are broad-based decisions.

Sensitivity of a loan dispersal model

Intuitive description: a small set of features can alter the decision.

In other words, all decisions of the model are broad-based decisions.

Example 1.1

Someone should not be able to manipulate their age to change the decision of the AI.

Sensitivity of a loan dispersal model

Intuitive description: a small set of features can alter the decision.

In other words, all decisions of the model are broad-based decisions.

Example 1.1

Someone should not be able to manipulate their age to change the decision of the AI.

- To give formal guarantees, we need to first define the problem mathematically.

Formal definition of sensitivity

The sensitivity problem

Given the model X and feature set $F \subseteq \mathcal{F}$, are there two loan applications x^1, x^2 such that

- x^1 and x^2 differ only on F (same on all the other features)
- but, outputs/decisions are significantly different, i.e., $X(x^1) < -gap$ and $X(x^2) > gap$ for some given $gap > 0$.

Formal definition of sensitivity

The sensitivity problem

Given the model X and feature set $F \subseteq \mathcal{F}$, are there two loan applications x^1, x^2 such that

- x^1 and x^2 differ only on F (same on all the other features)
- but, outputs/decisions are significantly different, i.e., $X(x^1) < -gap$ and $X(x^2) > gap$ for some given $gap > 0$.

Example 1.2

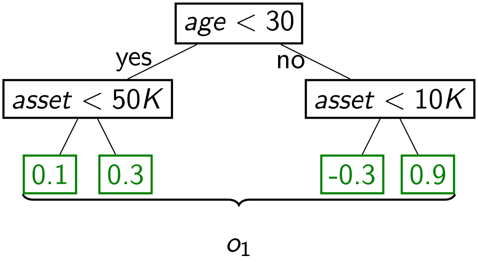
Is it possible to change the decision of the model by only changing the age?

$$F = \{age\}$$

Topic 1.2

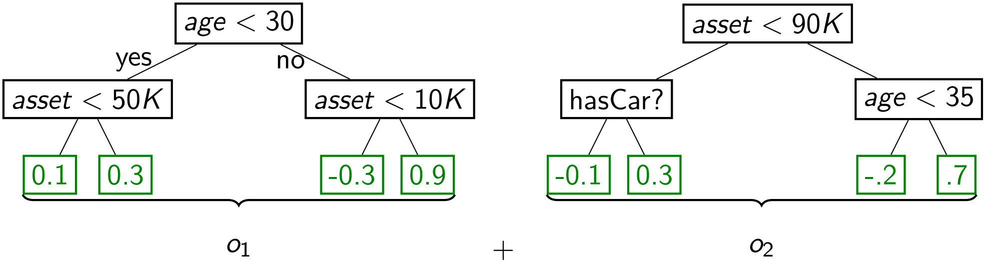
Models under analysis: tree ensembles

Tree ensemble models



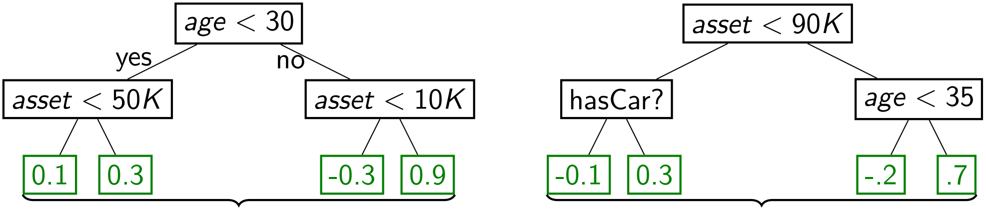
Widely used in the financial industry for learning on tabular data.

Tree ensemble models



Widely used in the financial industry for learning on tabular data.

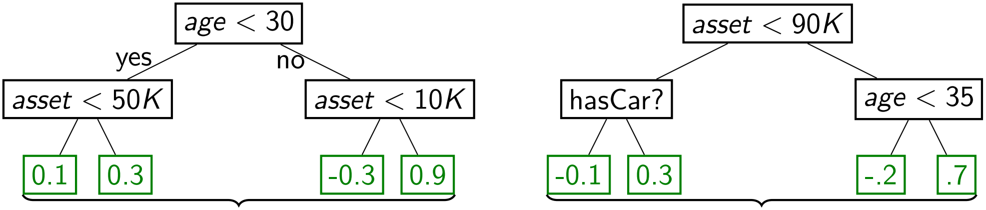
Tree ensemble models



$$X(age, asset, hasCar) = o_1 + o_2$$

Give a loan if $X(age, asset, hasCar) > 0$.

Tree ensemble models

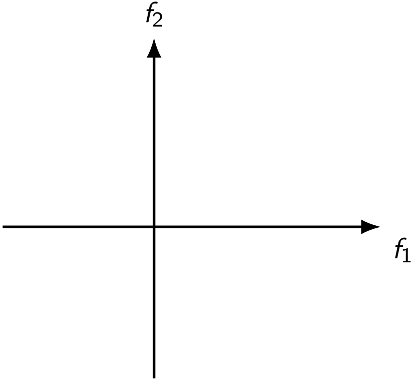


$$X(age, asset, hasCar) = o_1 + o_2$$

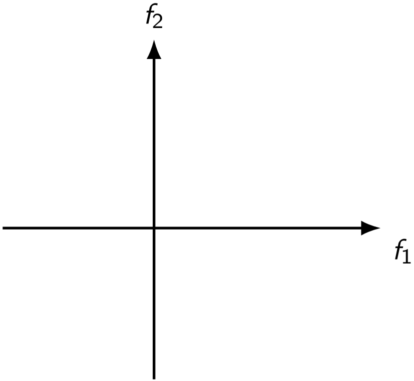
Give a loan if $X(age, asset, hasCar) > 0$.

Widely used in the financial industry for learning on tabular data.

Tree ensemble models

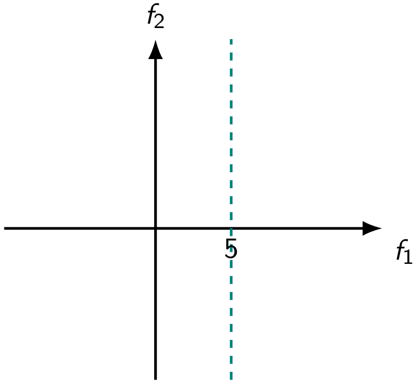


Tree ensemble models



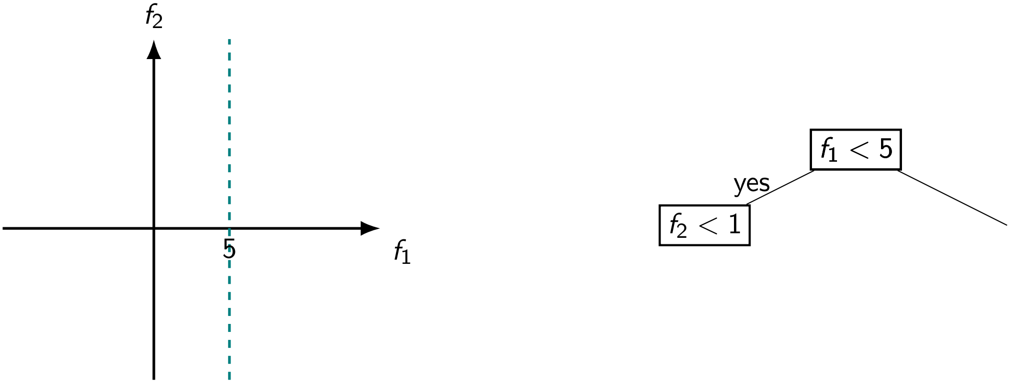
$f_1 < 5$

Tree ensemble models

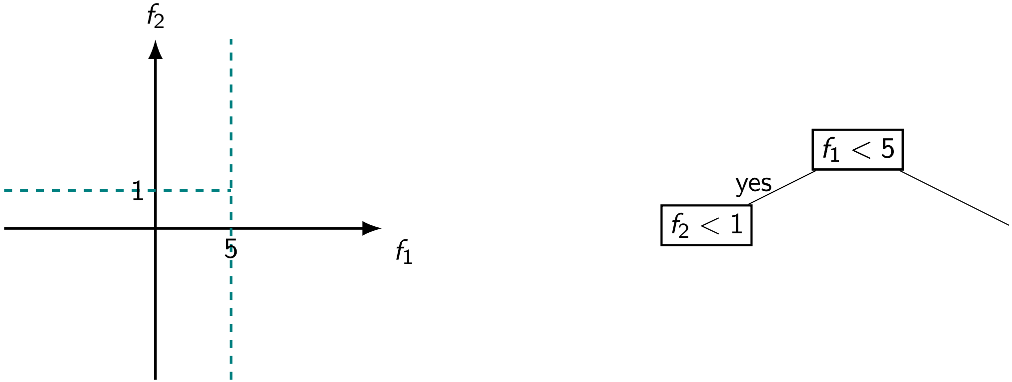


$f_1 < 5$

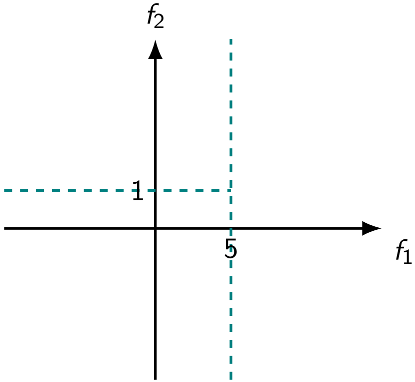
Tree ensemble models

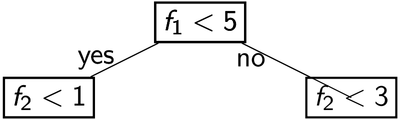


Tree ensemble models

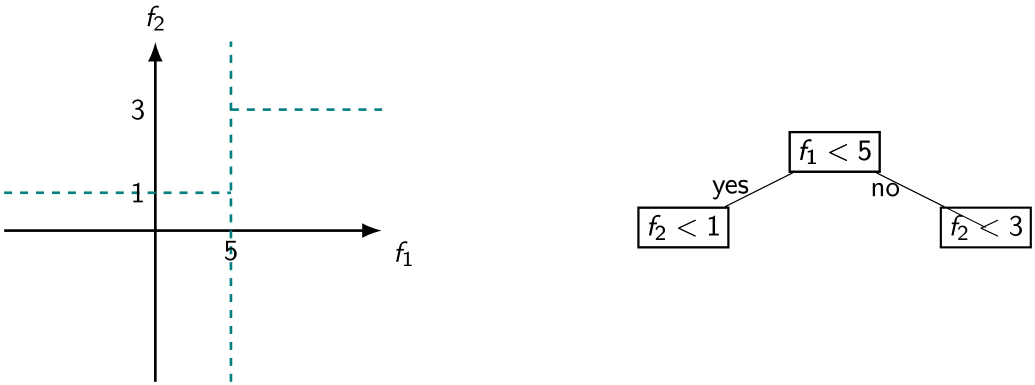


Tree ensemble models

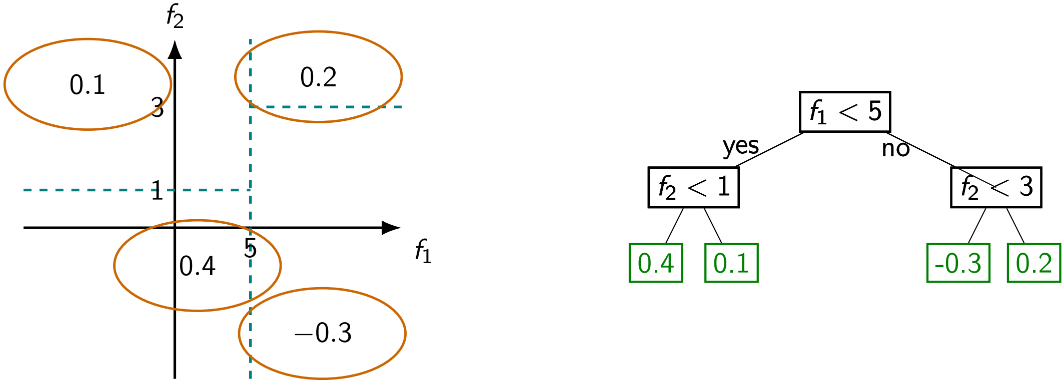




Tree ensemble models



Tree ensemble models



Examples of tree ensembles

Tree ensembles have variations.

- XGBoost : level-wise growth during training
- LightGBM : leaf-wise growth during training
- Random Forest : decision via majority vote

Examples of tree ensembles

Tree ensembles have variations.

- XGBoost : level-wise growth during training
- LightGBM : leaf-wise growth during training
- Random Forest : decision via majority vote

In our verification question, their differences do not matter.

Example: sensitive pair

The following pair is for an xgboost model with 200 trees, 5 depth, and 9 features on the dataset pimadiabetes from UCI, varying feature “BloodPressure”.

Example: sensitive pair

The following pair is for an xgboost model with 200 trees, 5 depth, and 9 features on the dataset pimadiabetes from UCI, varying feature "BloodPressure".

Point1: {'Pregnancies': 17, 'Glucose': 188, 'BloodPressure': 122, 'SkinThickness': 33, 'Insulin': 846, 'BMI': 67.1, 'DiabetesPedigreeFunction': 2.42, 'Age': 81},

Output: 0.579602

Point2: {'Pregnancies': 17.0, 'Glucose': 188, 'BloodPressure': 76, 'SkinThickness': 33, 'Insulin': 846, 'BMI': 67.1, 'DiabetesPedigreeFunction': 2.42, 'Age': 81}

Output: -0.557419

Is the sensitivity of tree ensembles an NP-hard problem?

Is the sensitivity of tree ensembles an NP-hard problem?

Yes, it's an NP-Hard problem

Is the sensitivity of tree ensembles an NP-hard problem?

Yes, it's an NP-Hard problem

This will be covered at the end.

Existence to quality

We are able to solve the problem, but what about the quality of the sensitive pairs?

Existence to quality

We are able to solve the problem, but what about the quality of the sensitive pairs?

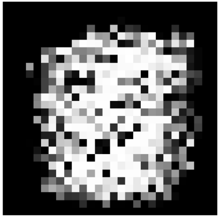
Existing literature looks at "what". [kantchelian et al. ICML'16, Ahmad et al. ICLR '25].

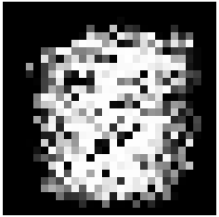
Existence to quality

We are able to solve the problem, but what about the quality of the sensitive pairs?

Existing literature looks at "what". [kantchelian et al. ICML'16, Ahmad et al. ICLR '25].

Here is a sensitive pair found by existing tool for a model that classifies letters between 3 and 8.



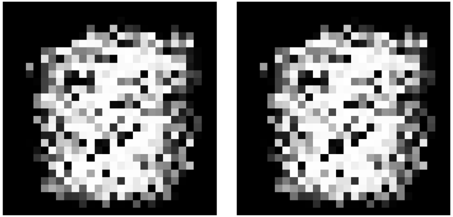


Existence to quality

We are able to solve the problem, but what about the quality of the sensitive pairs?

Existing literature looks at "what". [kantchelian et al. ICML'16, Ahmad et al. ICLR '25].

Here is a sensitive pair found by existing tool for a model that classifies letters between 3 and 8.



Proposed : "Find where not just what"

Search guided by marginal data distribution

Let $\mathcal{D}$ be the set of data points and K_f the number of guards for feature f.

Search guided by marginal data distribution

Let $\mathcal{D}$ be the set of data points and K_f the number of guards for feature f.

The following defines the marginal distribution of the data points.

$$\pi_f(v) = \sum_{k=2}^{K_f} \left(\mathbf{1}_{(\tau_{f(k-1)} \leq v < \tau_{fk})} \cdot \frac{|\{x \in \mathcal{D} \mid \tau_{f(k-1)} \leq x_f < \tau_{fk})\}|}{|\mathcal{D}|} \right)$$

Search guided by marginal data distribution

Let $\mathcal{D}$ be the set of data points and K_f the number of guards for feature f.

The following defines the marginal distribution of the data points.

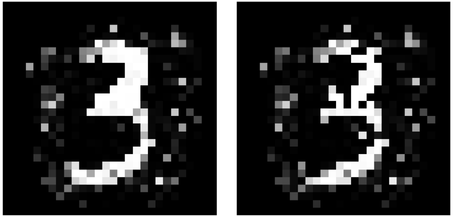
$$\pi_f(v) = \sum_{k=2}^{K_f} \left(\mathbf{1}_{(\tau_{f(k-1)} \leq v < \tau_{fk})} \cdot \frac{|\{x \in \mathcal{D} \mid \tau_{f(k-1)} \leq x_f < \tau_{fk})\}|}{|\mathcal{D}|} \right)$$

We modify our constraints to optimize the following objective function.

$$u\left(x^{(1)}, x^{(2)}\right) = \prod_{i=1}^{f} \pi_i\left(x^{(1)}, x^{(2)}\right).$$

Example(1): Data-aware sensitivity search

We made our search data aware, which resulted in finding the following sensitive pair for the model.



Example(2): data-aware vs. without data-aware analysis

Without data-aware Analysis

Point1: {'Pregnancies': 17, 'Glucose': 188, 'BloodPressure': 122, 'SkinThickness': 33, 'Insulin': 846, 'BMI': 67.1, 'DiabetesPedigreeFunction': 2.42, 'Age': 81},

Point2: {'Pregnancies': 17.0, 'Glucose': 188, 'BloodPressure': 76, 'SkinThickness': 33, 'Insulin': 846, 'BMI': 67.1, 'DiabetesPedigreeFunction': 2.42, 'Age': 81}

Distance from data: 0.3534358888
Nearest Training Datapoint:
{'Pregnancies': 10, 'Glucose': 148, 'BloodPressure': 84, 'SkinThickness': 48, 'Insulin': 237, 'BMI': 37.6, 'DiabetesPedigreeFunction': 1.001, 'Age': 51}

Example(2): data-aware vs. without data-aware analysis

Without data-aware Analysis

Point1: {'Pregnancies': 17, 'Glucose': 188, 'BloodPressure': 122, 'SkinThickness': 33, 'Insulin': 846, 'BMI': 67.1, 'DiabetesPedigreeFunction': 2.42, 'Age': 81},
Point2: {'Pregnancies': 17.0, 'Glucose': 188, 'BloodPressure': 76, 'SkinThickness': 33, 'Insulin': 846, 'BMI': 67.1, 'DiabetesPedigreeFunction': 2.42, 'Age': 81}

Distance from data: 0.3534358888
Nearest Training Datapoint:
{'Pregnancies': 10, 'Glucose': 148, 'BloodPressure': 84, 'SkinThickness': 48, 'Insulin': 237, 'BMI': 37.6, 'DiabetesPedigreeFunction': 1.001, 'Age': 51}

Data-aware Analysis

Point 1: {'Pregnancies': 0, 'Glucose': 139, 'BloodPressure':70, 'SkinThickness': 0, 'Insulin': 0, 'BMI': 32.75, 'DiabetesPedigreeFunction': 0.3595, 'Age': 21}
Point2: {'Pregnancies': 0, 'Glucose': 139, 'BloodPressure':79, 'SkinThickness': 0, 'Insulin': 0, 'BMI': 32.75, 'DiabetesPedigreeFunction': 0.3595, 'Age': 21}

Distance from data: 0.03051399
Nearest Training Datapoint:
{'Pregnancies': 0, 'Glucose': 132, 'BloodPressure': 78, 'SkinThickness': 0, 'Insulin': 0, 'BMI': 32.4, 'DiabetesPedigreeFunction': 0.393, 'Age': 21}

The insensitive features in the training data points that are far away from the sensitive pair are highlighted with cyan color.

Better quality results!

After adding the objective function, we found a sensitive pair closer to the data.

Method	Win%	Draw%	Loss%
Objective function vs No-Objective function	76.6	1.15	22.1

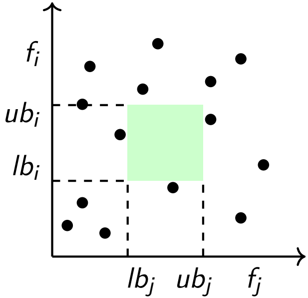
Improvement: correlation aware guidance

Marginal distribution ignores the correlation between features.

Improvement: correlation aware guidance

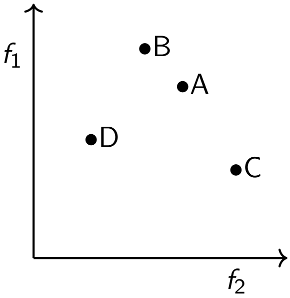
Marginal distribution ignores the correlation between features.

For highly correlated data distributions, we add cavity avoidance constraints.

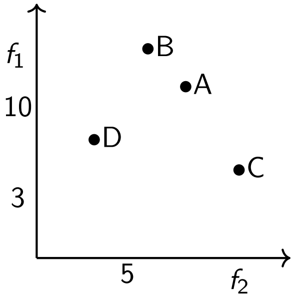


We search for the cavities in the training data and remove them from our search space.

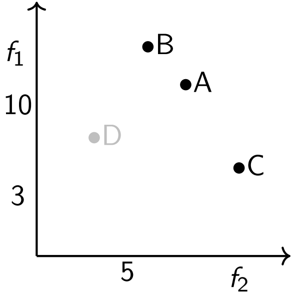
Synthesis of cavities



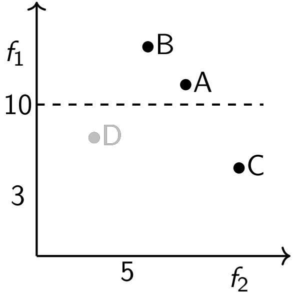
Synthesis of cavities



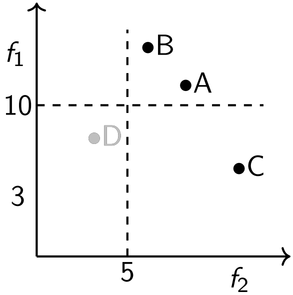
Synthesis of cavities



Synthesis of cavities

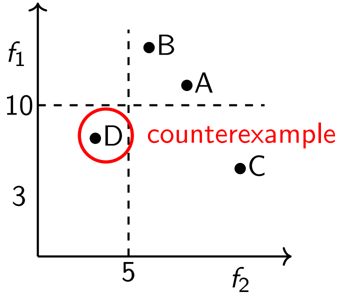


Synthesis of cavities



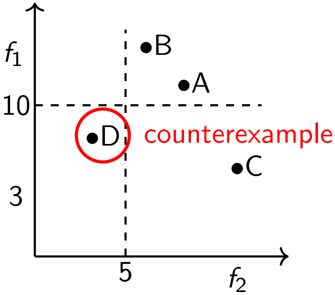
$$f_1 < 10 \wedge f_2 < 5$$

Synthesis of cavities



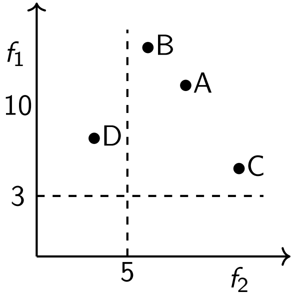
$$f_1 < 10 \wedge f_2 < 5$$

Synthesis of cavities



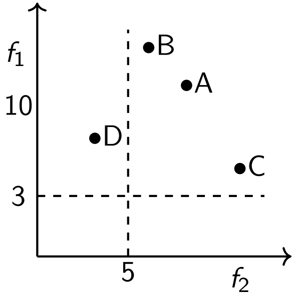
$f_1 < 10 \wedge f_2 < 5$ **X**

Synthesis of cavities



$$f_1 < 3 \wedge f_2 < 5$$

Synthesis of cavities



$$f_1 < 3 \land f_2 < 5 \qquad \checkmark$$

Experiments: after adding cavity constraints

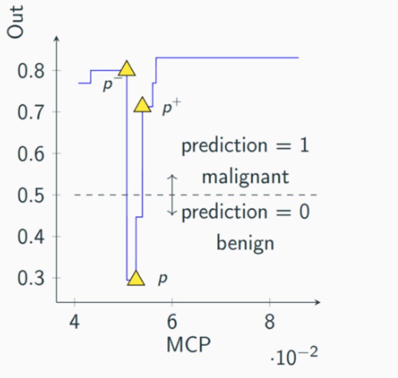
Method	Win%	Draw%	Loss%
Cavity Constraints vs Unguided Search	86.7	1.1	12.1

Topic 1.3

A novel property: Glitch

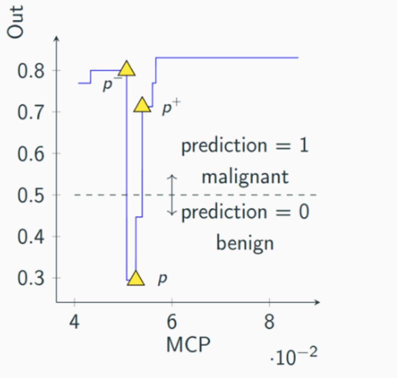
A newly discovered inconsistency:Glitches

The following glitch is found in an xgboost model with 100 trees, 5 depth, and 21 features on the breastcancer dataset from UCI.



A newly discovered inconsistency:Glitches

The following glitch is found in an xgboost model with 100 trees, 5 depth, and 21 features on the breastcancer dataset from UCI.



We call it "Glitches".

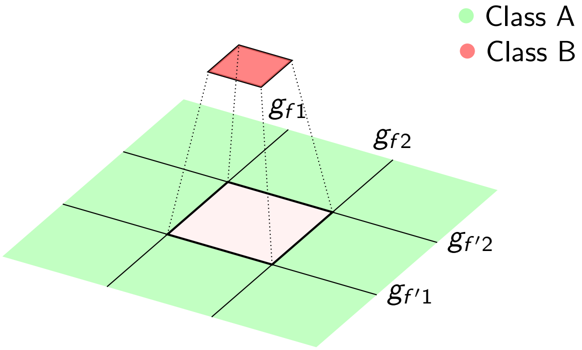
Formalisation of glitches in tree ensemble models

Let $\mathcal{F}\colon \mathbb{R}^m \to \mathbb{R}$ be a tree ensemble model.

Let $(\mathbf{x}^-, \mathbf{x}, \mathbf{x}^+)$ be an input triple such that there exists an i with $\mathbf{x}^-_i < \mathbf{x}_i < \mathbf{x}^+_i$, and for every $j \neq i$, $\mathbf{x}^-_j = \mathbf{x}_j$ and $\mathbf{x}_j = \mathbf{x}^+_j$. The triple $(\mathbf{x}^-, \mathbf{x}, \mathbf{x}^+)$ is a glitch in the dimension i with magnitude $\alpha > 0$ if α is the largest constant that satisfies:

$$\mathcal{F}(\mathbf{x}^-) > \mathcal{F}(\mathbf{x}) \wedge \mathcal{F}(\mathbf{x}) < \mathcal{F}(\mathbf{x}^+) \quad \text{or} \quad \mathcal{F}(\mathbf{x}^-) < \mathcal{F}(\mathbf{x}) \wedge \mathcal{F}(\mathbf{x}) > \mathcal{F}(\mathbf{x}^+) \tag{1}$$

$$\frac{\min\{d(\mathcal{F}(\mathbf{x}), \mathcal{F}(\mathbf{x}^-)),\ d(\mathcal{F}(\mathbf{x}), \mathcal{F}(\mathbf{x}^+))\}}{d(\mathbf{x}^-, \mathbf{x}^+)} \geq \alpha \tag{2}$$



Evidence of Glitches in Neural Networks

plastic_bag
(p=0.861) ε=0

bib
(p=0.6882),ε=0.06171

plastic_bag
(p=0.6702),ε=0.15457

Conclusion

- We have developed methods to verify AI systems
- Technology exists that can analyze small to mid-size AI systems
- Call to action: develop analysis technology that scales to large AI systems.

Conclusion

- We have developed methods to verify AI systems
- Technology exists that can analyze small to mid-size AI systems
- Call to action: develop analysis technology that scales to large AI systems.

Questions

Topic 1.4

Is the sensitivity of tree ensembles an NP-hard problem?

The sensitivity of tree ensembles is NP-hard

Theorem 1.1

The single feature sensitivity problem, i.e., checking whether a given tree ensemble classifier is F-sensitive for —F— $= 1$, is NP-hard.

The sensitivity of tree ensembles is NP-hard

Theorem 1.1

The single feature sensitivity problem, i.e., checking whether a given tree ensemble classifier is F-sensitive for —F— $= 1$, is NP-hard.

Proof.

Take the 3CNF formula $\phi = c_1 \wedge ... \wedge c_m$ with m clauses and $v_1, ..., v_n$ variables.

The sensitivity of tree ensembles is NP-hard

Theorem 1.1

The single feature sensitivity problem, i.e., checking whether a given tree ensemble classifier is F-sensitive for —F— $= 1$, is NP-hard.

Proof.

Take the 3CNF formula $\phi = c_1 \wedge ... \wedge c_m$ with m clauses and $v_1, ..., v_n$ variables.

We construct a sensitivity problem for a tree ensemble X such that X is sensitive iff ϕ is satisfiable.

The sensitivity of tree ensembles is NP-hard

Theorem 1.1

The single feature sensitivity problem, i.e., checking whether a given tree ensemble classifier is F-sensitive for —F— $= 1$, is NP-hard.

Proof.

Take the 3CNF formula $\phi = c_1 \wedge ... \wedge c_m$ with m clauses and $v_1, ..., v_n$ variables.

We construct a sensitivity problem for a tree ensemble X such that X is sensitive iff ϕ is satisfiable.

We consider a formula $\phi' = \phi \wedge v_{n+1}$, where v_{n+1} is a fresh variable. ...

Proof: a tree for each clause

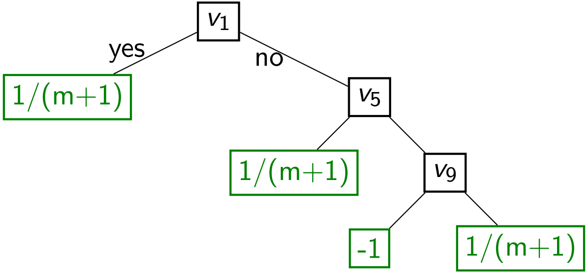
In our tree ensemble X, we construct a decision tree for each clause of ϕ'.

Proof: a tree for each clause

In our tree ensemble X, we construct a decision tree for each clause of ϕ'.

Example 1.3

Let us suppose $(v_1 \vee v_5 \vee \neg v_9) \in \phi'$. We construct the following tree for the clause.

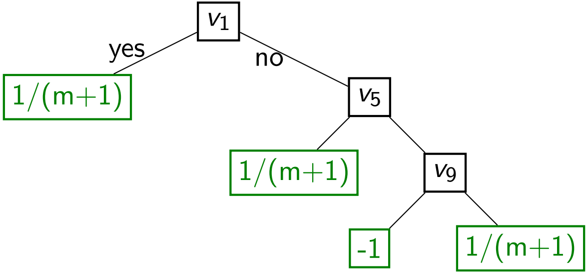


Proof: a tree for each clause

In our tree ensemble X, we construct a decision tree for each clause of ϕ'.

Example 1.3

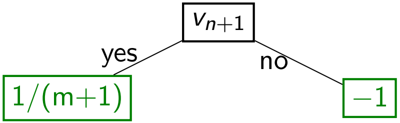
Let us suppose $(v_1 \vee v_5 \vee \neg v_9) \in \phi'$. We construct the following tree for the clause.



Recall ϕ' has $m + 1$ clauses.

Proof: consider the last clause of ϕ'

The tree for the last clause of ϕ' is v_{n+1}.



Proof: Decision vs satisfaction

Theorem 1.2

$x \models \phi'$ iff $X(x) > 0$.

Proof.

If $x \models \phi'$, all of the trees in X produce positive output. Therefore, $X(x) > 0$.

Proof: Decision vs satisfaction

Theorem 1.2

$x \models \phi'$ iff $X(x) > 0$.

Proof.

If $x \models \phi'$, all of the trees in X produce positive output. Therefore, $X(x) > 0$.

If $x \not\models \phi'$,

- at least one of the tree in X produce -1 output, and
- the sum of outputs of all the other trees is at most $m/(m+1)$.

Therefore, $X(x) < 0$. □

Proof: Sensitivity vs satisfiability

Theorem 1.3

ϕ is satisfiable iff X is $\{v_{n+1}\}$-sensitive.

Proof.

Assume ϕ is satisfiable. Let $x \models \phi$ for some x.

Proof: Sensitivity vs satisfiability

Theorem 1.3

ϕ is satisfiable iff X is $\{v_{n+1}\}$-sensitive.

Proof.

Assume ϕ is satisfiable. Let $x \models \phi$ for some x.

Iff, $x[v_{n+1} \mapsto 1] \models \phi'$ and $x[v_{n+1} \mapsto 0] \not\models \phi'$.

Proof: Sensitivity vs satisfiability

Theorem 1.3

ϕ is satisfiable iff X is $\{v_{n+1}\}$-sensitive.

Proof.

Assume ϕ is satisfiable. Let $x \models \phi$ for some x.

Iff, $x[v_{n+1} \mapsto 1] \models \phi'$ and $x[v_{n+1} \mapsto 0] \not\models \phi'$.

Iff, $X(x[v_{n+1} \mapsto 1]) > 0$ and $X(x[v_{n+1} \mapsto 0]) < 0$.

Proof: Sensitivity vs satisfiability

Theorem 1.3

ϕ is satisfiable iff X is $\{v_{n+1}\}$-sensitive.

Proof.

Assume ϕ is satisfiable. Let $x \models \phi$ for some x.

Iff, $x[v_{n+1} \mapsto 1] \models \phi'$ and $x[v_{n+1} \mapsto 0] \not\models \phi'$.

Iff, $X(x[v_{n+1} \mapsto 1]) > 0$ and $X(x[v_{n+1} \mapsto 0]) < 0$.

Iff, X is $\{v_{n+1}\}$-sensitive. □