Merlin-Arthur Classifiers: Formal Interpretability with Interactive Black Boxes

DMV 2022
Kartikey Sharma
15th September 2022
Results are joint work with...

Stephan Wäldchen  
Zuse Institute Berlin

Sebastian Pokutta  
Zuse Institute Berlin

Max Zimmer  
Zuse Institute Berlin
Merlin-Arthur Classifiers

1. Introduction

2. Theoretical Framework

3. Experiments
1. Introduction

**Explainable AI**

**Motivation**
- Neural Networks form a key part of AI
- Outcomes difficult to explain

**Consequences**
- Existence of hidden biases and vulnerabilities
- Lower trust

**Our Contribution**
- Interpretable classification system with theoretical guarantees on features.
1. Introduction

**Explainable AI**

**Motivation**
- Neural Networks form a key part of AI
- Outcomes difficult to explain

**Consequences**
- Existence of hidden biases and vulnerabilities
- Lower trust

**Our Contribution**
- Interpretable classification system with theoretical guarantees on features.
1. Introduction

**Explainable AI**

**Motivation**
- Neural Networks form a key part of AI
- Outcomes difficult to explain

**Consequences**
- Existence of hidden biases and vulnerabilities
- Lower trust

**Our Contribution**
- Interpretable classification system with theoretical guarantees on features.
1. Introduction

Existing Literature

Heuristic Approaches

- Saliency maps (Mohseni, Zarei, and Ragan 2021), Mechanistic interpretability (Olah et al. 2018).
- Their success cannot be verified. Can be manipulated by a clever design of the NN (Slack, Hilgard, Lakkaraju, et al. 2021; Slack, Hilgard, Jia, et al. 2020; Anders et al. 2020).

Formal Approaches

- Can run into complexity problems, require an exponential amount of time (Macdonald et al. 2020; Ignatiev, Narodytska, and Marques-Silva 2019).
1. Introduction

Existing Literature

Heuristic Approaches

- Saliency maps (Mohseni, Zarei, and Ragan 2021), Mechanistic interpretability (Olah et al. 2018).
- Their success cannot be verified. Can be manipulated by a clever design of the NN (Slack, Hilgard, Lakkaraju, et al. 2021; Slack, Hilgard, Jia, et al. 2020; Anders et al. 2020).

Formal Approaches

- Can run into complexity problems, require an exponential amount of time (Macdonald et al. 2020; Ignatiev, Narodytska, and Marques-Silva 2019).
1. Introduction

Background

- **Task:** For \( x \in D \), select feature \( \phi \in \Sigma \).
- \( \phi \) should have high mutual information to the class \( c(x) \in C \).
- Can we lower-bound \( I(c(x); "x \text{ contains } \phi") \)?
- **Problem:** Would require model of data manifold with bound on error
- **Idea:** Retrain on selected features
- **Problem:** Cheating!

![Diagram showing data space and feature space](image-url)
1. Introduction

Background

- **Task:** For $x \in D$, select feature $\phi \in \Sigma$.
- $\phi$ should have high mutual information to the class $c(x) \in C$.
- Can we lower-bound $I(c(x); \text{“x contains } \phi\text{”})$?
- **Problem:** Would require model of data manifold with bound on error.
- **Idea:** Retrain on selected features.
- **Problem:** Cheating!
1. Introduction

Background

- **Task:** For \( x \in D \), select feature \( \phi \in \Sigma \).
- \( \phi \) should have high mutual information to the class \( c(x) \in C \).
- Can we lower-bound \( I(c(x); \text{"x contains } \phi\")?\)
- **Problem:** Would require model of data manifold with bound on error
- **Idea:** Retrain on selected features
- **Problem:** Cheating!

![Diagram](image)

**Feature Selector:**

Data Space \( D \)

**Feature Space \( \Sigma \)**

**Feature Classifier:**

Class Space \( C \)

“Bird”
1. Introduction

Background

- **Task:** For $x \in D$, select feature $\phi \in \Sigma$.
- $\phi$ should have high mutual information to the class $c(x) \in C$.
- Can we lower-bound $I(c(x); \text{"x contains $\phi$")}$?
- **Problem:** Would require model of data manifold with bound on error.
- **Idea:** Retrain on selected features.
- **Problem:** Cheating!
1. Introduction
Cheating

**Original Images:**

\[
P(C = \text{"boat"} | \text{"sea"}) = 0.5 \\
P(C = \text{"isle"} | \text{"sea"}) = 0.5
\]

\[
I(C ; \text{"sea"}) = 0
\]

**Masked Images:**

\[
P(C = \text{"boat"} | \text{"sea"}) = 1 \\
P(C = \text{"isle"} | \text{"sea"}) = 0
\]

\[
I(C ; \text{"sea"}) = 1
\]
1. Introduction

Methodology: Merlin-Arthur Classification

- Based on Merlin-Arthur protocols from Interactive Proof Systems
  - Cooperative feature selector / Prover (Merlin): $M$
  - Adversarial feature selector / Prover (Morgana): $\hat{M}$
  - Classifier / Verifier (Arthur): $A$
- Arthur should leverage Merlin but not be misled by Morgana
1. Introduction

**Methodology: Merlin-Arthur Classification**

- Based on Merlin-Arthur protocols from Interactive Proof Systems
- Cooperative feature selector / Prover (Merlin): \( M \)
- Adversarial feature selector / Prover (Morgana): \( \hat{M} \)
- Classifier / Verifier (Arthur): \( A \)
- Arthur should leverage Merlin but not be misled by Morgana
1. Introduction

**Methodology: Merlin-Arthur Classification**

- Based on Merlin-Arthur protocols from Interactive Proof Systems
- Cooperative feature selector / Prover (Merlin): $M$
- Adversarial feature selector / Prover (Morgana): $\hat{M}$
- Classifier / Verifier (Arthur): $A$
- Arthur should leverage Merlin but not be misled by Morgana
1. Introduction

Methodology: Merlin-Arthur Classification

- Based on Merlin-Arthur protocols from Interactive Proof Systems
- Cooperative feature selector / Prover (Merlin): $M$
- Adversarial feature selector / Prover (Morgana): $\hat{M}$
- Classifier / Verifier (Arthur): $A$
- Arthur should **leverage Merlin** but not be misled by Morgana
1. Introduction

**Methodology: Merlin-Arthur Classification**

- Based on Merlin-Arthur protocols from Interactive Proof Systems
- Cooperative feature selector / Prover (Merlin): $M$
- Adversarial feature selector / Prover (Morgana): $\widehat{M}$
- Classifier / Verifier (Arthur): $A$
- Arthur should **leverage Merlin** but **not be misled by Morgana**
1. Introduction

Cheating with Morgana

- "Isle!" × "Boat!" ×
- "Don’t know!" √ "Don’t know!" √
- "Don’t know!" × "Don’t know!" ×
- "Boat!" √ "Isle!" √
Merlin-Arthur Classifiers

1. Introduction

2. Theoretical Framework

3. Experiments
2. Theoretical Framework

**Average Precision**

**Definition**

Completeness: \( \min_{l \in \{-1, 1\}} \mathbb{P}_{x \sim D} [A(M(x)) = c(x)] \geq 1 - \epsilon_c, \)

Soundness: \( \max_{l \in \{-1, 1\}} \mathbb{P}_{x \sim D} [A(\hat{M}(x)) = -c(x)] \leq \epsilon_s. \)

**Definition**

Given a feature selector \( M \in \mathcal{M}(D) \), the average precision of \( M \) with respect to the data distribution \( D \) is

\[
Q_D(M) := \mathbb{E}_{x \sim D} \left[ \mathbb{P}_{y \sim D} [c(y) = c(x) | y \in M(x)] \right]
\]
2. Theoretical Framework

**Average Precision**

**Definition**

Completeness: \( \min_{l \in \{-1, 1\}} \mathbb{P}_{x \sim D}[A(M(x)) = c(x)] \geq 1 - \epsilon_c, \)

Soundness: \( \max_{l \in \{-1, 1\}} \mathbb{P}_{x \sim D}[A(\hat{M}(x)) = -c(x)] \leq \epsilon_s. \)

**Definition**

Given a feature selector \( M \in \mathcal{M}(\mathcal{D}) \), the average precision of \( \mathcal{M} \) with respect to the data distribution \( \mathcal{D} \) is

\[
Q_\mathcal{D}(M) := \mathbb{E}_{x \sim \mathcal{D}}[\mathbb{P}_{y \sim \mathcal{D}}[c(y) = c(x) | y \in M(x)]]
\]
2. Theoretical Framework

Performance of Merlin

Classifier: $A$, Feature Selectors: Merlin $M$ and Morgana $\hat{M}$

$$E_{M,\hat{M},A} := \left\{ x \in D \mid A(M(x)) \neq c(x) \lor A(\hat{M}(x)) = -c(x) \right\},$$

Theorem (Wäldchen 2022+)

Let $M \in \mathcal{M}(D)$ be a feature selector and let

$$\epsilon_M = \min_{A \in A} \max_{\hat{M} \in \mathcal{M}} \mathbb{P}_{x \sim D}[x \in E_{M,\hat{M},A}].$$

Then there exists a set $D' \subset D$ with $\mathbb{P}_{x \sim D}[x \in D'] \geq 1 - \epsilon_M$ such that for $\mathcal{D}' = D|_{D'}$ we have

$$Q_{\mathcal{D}'}(M) = 1 \quad \text{and thus} \quad H_{x,y \sim \mathcal{D}'}(c(y) \mid y \in M(x)) = 0.$$
2. Theoretical Framework

**Performance of Merlin**

Classifier: $A$, Feature Selectors: Merlin $M$ and Morgana $\hat{M}$

$$E_{M, \hat{M}, A} := \{ x \in D \mid A(M(x)) \neq c(x) \lor A(\hat{M}(x)) = -c(x) \},$$

**Theorem (Wäldchen 2022+)**

Let $M \in \mathcal{M}(D)$ be a feature selector and let

$$\epsilon_M = \min_{A \in A} \max_{\hat{M} \in \mathcal{M}} \mathbb{P}_{x \sim D} \left[ x \in E_{M, \hat{M}, A} \right].$$

Then there exists a set $D' \subset D$ with $\mathbb{P}_{x \sim D}[x \in D'] \geq 1 - \epsilon_M$ such that for $D' = D|_{D'}$ we have

$$Q_{D'}(M) = 1 \quad \text{and thus} \quad H_{x, y \sim D'}(c(y) \mid y \in M(x)) = 0.$$
2. Theoretical Framework

**Average Precision Bound**

**Theorem (Wäldchen 2022+)**

Let $\mathcal{D} = ((D, \sigma, \mathcal{D}), c, \Sigma)$ be a two-class data space with AFC of $\kappa$ and class imbalance $B$. Let $A \in \mathcal{A}$, $M$ and $\hat{M} \in \mathcal{M}(\mathcal{D})$ such that $\hat{M}$ has a context impact of $\alpha$ with respect to $A$, $M$ and $\mathcal{D}$. Then it follows that

$$Q_D(M) \geq 1 - \epsilon_c - \frac{\alpha \kappa \epsilon_s}{1 - \epsilon_c + \alpha \kappa \epsilon_s B^{-1}}.$$ 

**Corollary**

$$\mathbb{E}_{x \sim D}[l_{y \sim D}(c(y); y \in M(x))] \geq H_{y \sim D}(c(y)) - H_b(Q_D(M)).$$
2. Theoretical Framework

**Average Precision Bound**

**Theorem (Wäldchen 2022+)**

Let $\mathcal{D} = ((D, \sigma, D), c, \Sigma)$ be a two-class data space with AFC of $\kappa$ and class imbalance $B$. Let $A \in \mathcal{A}$, $M$ and $\hat{M} \in \mathcal{M}(D)$ such that $\hat{M}$ has a context impact of $\alpha$ with respect to $A$, $M$ and $\mathcal{D}$. Then it follows that

$$Q_D(M) \geq 1 - \epsilon_c - \frac{\alpha \kappa \epsilon_s}{1 - \epsilon_c + \alpha \kappa \epsilon_s B^{-1}}.$$ 

**Corollary**

$$\mathbb{E}_{x \sim D}[I_{y \sim D}(c(y); y \in M(x))] \geq H_{y \sim D}(c(y)) - H_b(Q_D(M)).$$
Merlin-Arthur Classifiers

1. Introduction
2. Theoretical Framework
3. Experiments
3. Experiments

Experimental Setup

- **MNIST Dataset**

- **Models:**
  - Merlin and Morgana (Feature Selectors): FW-Classifier and U-Net
  - Arthur (Classifier): Convolutional Neural Network

- **Training process:**
  - Alternate between gradients steps for the masks and for the classifier
  - Alternate between epochs over masked images and regular images
3. Experiments

Experimental Setup

- **MNIST Dataset**
- **Models:**
  - Merlin and Morgana (Feature Selectors): FW-Classifier and U-Net
  - Arthur (Classifier): Convolutional Neural Network
- **Training process:**
  - Alternate between gradients steps for the masks and for the classifier
  - Alternate between epochs over masked images and regular images
3. Experiments

Experimental Setup

- **MNIST Dataset**
- **Models:**
  - Merlin and Morgana (Feature Selectors): FW-Classifier and U-Net
  - Arthur (Classifier): Convolutional Neural Network
- **Training process:**
  - Alternate between gradients steps for the masks and for the classifier
  - Alternate between epochs over masked images and regular images
3. Experiments

Selected Features

Key Point: Merlin features which tend to be unique to the class when Morgana is present.
3. Experiments

Experimental Results

Key Point: Very low error rates even at small pixel sizes for Merlin only. Errors increase when Morgana is present.
3. Experiments

**Experimental Results**

**Key Point**: Very low error rates even at small pixel sizes for Merlin only. Errors increase when Morgana is present.
3. Experiments

**Experimental Results**

Key Point: Very low error rates even at small pixel sizes for Merlin only. Errors increase when Morgana is present.
3. Experiments

**Experimental Results**

**Key Point:** Very low error rates even at small pixel sizes for Merlin only. Errors increase when Morgana is present.
Key Point: As mask size increases the bound becomes tighter and the completeness and soundness increase.
3. Experiments

Conclusion

- We provide an interpretable classification framework inspired by interactive proof systems.
- We achieve guarantees on the mutual information of the features with the class by expressing it in terms of measurable criteria such as completeness and soundness.
- We evaluate our results numerically on the MNIST data set. We observe high quality features which also demonstrate good agreement between our theoretical bounds and the experimental quality of the exchanged features.
Thank you for your attention!
3. Experiments

References

Anders, Christopher et al. (2020). “Fairwashing explanations with off-manifold detergent”. In: *International Conference on Machine Learning*. PMLR, pp. 314–323.

Ignatiev, Alexey, Nina Narodytska, and Joao Marques-Silva (2019). “Abduction-based explanations for machine learning models”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01, pp. 1511–1519.

Macdonald, Jan et al. (2020). “Explaining neural network decisions is hard”. In: *XXAI Workshop, 37th ICML*.

Mohseni, Sina, Niloofar Zarei, and Eric D Ragan (2021). “A multidisciplinary survey and framework for design and evaluation of explainable AI systems”. In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 11.3-4, pp. 1–45.

Olah, Chris et al. (2018). “The building blocks of interpretability”. In: *Distill 3.3*, e10.
References

- Slack, Dylan, Sophie Hilgard, Emily Jia, et al. (2020). “Fooling lime and shap: Adversarial attacks on post hoc explanation methods”. In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 180–186.
- Slack, Dylan, Sophie Hilgard, Himabindu Lakkaraju, et al. (2021). “Counterfactual Explanations Can Be Manipulated”. In: *arXiv preprint arXiv:2106.02666*. 
3. Experiments

**Training Algorithm**

**Data:** Dataset: $D$, Epochs: $N$, $\gamma$

**Result:** Classifier ($A$), Optional: Masking Networks Merlin ($M$) and Morgana ($\hat{M}$)

for $i \in [N]$ do

for $x_j, y_j \in D$ do

$s_M \leftarrow M(x_j, y_j), s_{\hat{M}} \leftarrow \hat{M}(x_j, y_j)$, $M$, $\hat{M}$ can be optimiser or NN

$A \leftarrow \arg \min_A (1 - \gamma)L_M(A(s_M \cdot x_j), y_j) + \gamma L_{\hat{M}}(A(s_{\hat{M}} \cdot x_j), y_j)$ Update classifier using masked images

$M \leftarrow \arg \min L_M(A(M(x_j) \cdot x_j), y_j)$ Update only if $M$ is a NN

$\hat{M} \leftarrow \arg \max L_{\hat{M}}(A(\hat{M}(x_j) \cdot x_j), y_j)$ Update only if $\hat{M}$ is a NN

end

for $x_j, y_j \in D$ do

$A \leftarrow \arg \min_A L(A(x_j), y_j)$ Update classifier using regular images

end
3. Experiments

**Training Algorithm**

**Data:** Dataset: $D$, Epochs: $N$, $\gamma$

**Result:** Classifier ($A$), Optional: Masking Networks Merlin ($M$) and Morgana ($\hat{M}$)

for $i \in [N]$ do

    for $x_j, y_j \in D$ do

        $s_M \leftarrow M(x_j, y_j)$, $s_{\hat{M}} \leftarrow \hat{M}(x_j, y_j)$, $M$, $\hat{M}$ can be optimiser or NN

        $A \leftarrow \arg \min_A (1 - \gamma)L_M(A(s_M \cdot x_j), y_j) + \gamma L_{\hat{M}}(A(s_{\hat{M}} \cdot x_j), y_j)$ Update classifier using masked images

        $M \leftarrow \arg \min L_M(A(M(x_j) \cdot x_j), y_j)$ Update only if $M$ is a NN

        $\hat{M} \leftarrow \arg \max L_{\hat{M}}(A(\hat{M}(x_j) \cdot x_j), y_j)$ Update only if $\hat{M}$ is a NN

    end

end

for $x_j, y_j \in D$ do

    $A \leftarrow \arg \min_A L(A(x_j), y_j)$ Update classifier using regular images

end
3. Experiments

**Training Algorithm**

**Data:** Dataset: $D$, Epochs: $N$, $\gamma$

**Result:** Classifier ($A$), Optional: Masking Networks Merlin ($M$) and Morgana ($\widehat{M}$)

for $i \in [N]$ do
  for $x_j, y_j \in D$ do
    $s_M \leftarrow M(x_j, y_j), s_{\widehat{M}} \leftarrow \widehat{M}(x_j, y_j)$
    $M, \widehat{M}$ can be optimiser or NN
    $A \leftarrow \arg\min_{A} (1 - \gamma)L_M(A(s_M \cdot x_j), y_j) + \gamma L_{\widehat{M}}(A(s_{\widehat{M}} \cdot x_j), y_j)$ Update classifier using masked images
    $M \leftarrow \arg\min L_M(A(M(x_j) \cdot x_j), y_j)$ Update only if $M$ is a NN
    $\widehat{M} \leftarrow \arg\max L_{\widehat{M}}(A(\widehat{M}(x_j) \cdot x_j), y_j)$ Update only if $\widehat{M}$ is a NN
  end
end

for $x_j, y_j \in D$ do
  $A \leftarrow \arg\min_{A} L(A(x_j), y_j))$ Update classifier using regular images
end