DIGITAL FINANCE

This project has received funding from the Horizon Europe research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 101119635

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

State Secretariat for Education, Research and Innovation SERI

Funded by the European Union

MARIE CURIE ACTIONS

DIGITAL

Reading

Laguna, S., Heidenreich, J. N., Sun, J., Cetin, N., Al-Hazmi, I., Schlegel, U., Cheng, F., & El-Assady, M. (2023). ExplLIMEable: An exploratory framework for LIME. In XAI in Action: Past, Present, and Future Applications (NeurIPS 2023 Workshop).

DIGITAL

Lime & ExpLIMEable

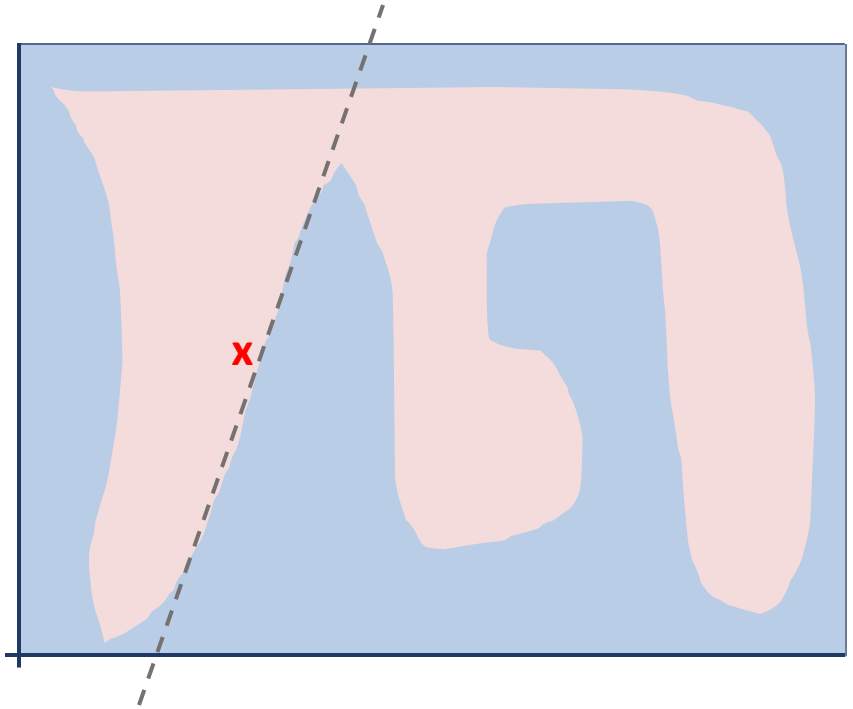
MSCA Digital

Laguna, S., Heidenreich, J. N., Sun, J., Cetin, N., Al-Hazmi, I., Schlegel, U., Cheng, F., & El-Assady, M. (2023). ExplLIMEable: An exploratory framework for LIME. In XAI in Action: Past, Present, and Future Applications (NeurIPS 2023 Workshop).

DIGITAL

Funded by the European Union

Recap: LIME



"Why Should I Trust You?" Explaining the Predictions of Any Classifier

Marco Tulio Ribeiro
University of Washington
Seattle, WA 98105, USA
marcotcr@cs.uw.edu

Sameer Singh
University of Washington
Seattle, WA 98105, USA
sameer@cs.uw.edu

Carlos Guestrin
University of Washington
Seattle, WA 98105, USA
guestrin@cs.uw.edu

ABSTRACT

Despite widespread adoption, machine learning models remain mostly black boxes. Understanding the reasons behind predictions is, however, quite important in assessing *trust*, which is fundamental if one plans to take action based on a prediction, or when choosing whether to deploy a new model. Such understanding also provides insights into the model, which can be used to transform an untrustworthy model or prediction into a trustworthy one.

In this work, we propose LIME, a novel explanation technique that explains the predictions of *any* classifier in an interpretable and faithful manner, by learning an interpretable model locally around the prediction. We also propose a method to explain models by presenting representative individual predictions and their explanations in a non-redundant way, framing the task as a submodular optimization problem. We demonstrate the flexibility of these methods by explaining different models for text (e.g. random forests) and image classification (e.g. neural networks). We show the utility of explanations via novel experiments, both simulated and with human subjects, on various scenarios that require trust: deciding if one should trust a prediction, choosing between models, improving an untrustworthy classifier, and identifying why a classifier should not be trusted.

1. INTRODUCTION

Machine learning is at the core of many recent advances in science and technology. Unfortunately, the important role of humans is an oft-overlooked aspect in the field. Whether humans are directly using machine learning classifiers as tools, or are deploying models within other products, a vital concern remains: *if the users do not trust a model or a prediction, they will not use it.* It is important to differentiate between two different (but related) definitions of trust: (1) *trusting a prediction*, i.e. whether a user trusts an individual prediction sufficiently to take some action based on it, and (2) *trusting a model*, i.e. whether the user trusts a model to behave in reasonable ways if deployed. Both are directly impacted by how much the human understands a model's behaviour, as opposed to seeing it as a black box.

Determining trust in individual predictions is an important problem when the model is used for decision making. When using machine learning for medical diagnosis [6] or terrorism detection, for example, predictions cannot be acted upon on blind faith, as the consequences may be catastrophic.

Apart from trusting individual predictions, there is also a need to evaluate the model as a whole before deploying it "in the wild". To make this decision, users need to be confident that the model will perform well on real-world data, according to the metrics of interest. Currently, models are evaluated using accuracy metrics on an available validation dataset. However, real-world data is often significantly different, and further, the evaluation metric may not be indicative of the product's goal. Inspecting individual predictions and their explanations is a worthwhile solution, in addition to such metrics. In this case, it is important to aid users by suggesting which instances to inspect, especially for large datasets.

In this paper, we propose providing explanations for individual predictions as a solution to the "trusting a prediction" problem, and selecting multiple such predictions (and explanations) as a solution to the "trusting the model" problem. Our main contributions are summarized as follows.

- LIME, an algorithm that can explain the predictions of *any* classifier or regressor in a faithful way, by approximating it locally with an interpretable model.
- SP-LIME, a method that selects a set of representative instances with explanations to address the "trusting the model" problem, via submodular optimization.
- Comprehensive evaluation with simulated and human subjects, where we measure the impact of explanations on trust and associated tasks. In our experiments, non-experts using LIME are able to pick which classifier from a pair generalizes better in the real world. Further, they are able to greatly improve an untrustworthy classifier trained on 20 newsgroups, by doing feature engineering using LIME. We also show how understanding the predictions of a neural network on images helps practitioners know when and why they should not trust a model.

2. THE CASE FOR EXPLANATIONS

By "explaining a prediction", we mean presenting textual or visual artifacts that provide qualitative understanding of the relationship between the instance's components (e.g. words in text, patches in an image) and the model's prediction. We argue that explaining predictions is an important aspect in

KDD 2016 San Francisco, CA, USA

ISBN 978-1-4503-4232-2/16/08…$15.00
DOI: http://dx.doi.org/10.1145/2939672.2939778

DIGITAL

LIME: **Formally**

$$\xi(x) = argmin_{g \in G} L(f, g, \pi_x) + \Omega(g)$$

- $\xi(x)$ is the **explanation function.**
- f is the **black-box model** we want to explain.
- $g \in G$ represents the set of **interpretable models** (e.g., linear regression, decision trees).
- $L(f, g, \pi_x)$ is the **loss function.**
- π_x is the **proximity function.**
- $\Omega(g)$ is a **complexity penalty**.

DIGITAL

Molnar (2023)

LIME: **Formally**

$$\xi(x) = argmin_{g \in G} L(f, g, \pi_x) + \Omega(g) \qquad (1)$$

$$\boldsymbol{L(f, g, \pi_x)} = \sum_i \pi_x(x_i)\big(f(x_i) - g(x_i)\big)^2 \qquad (2)$$

- We want to ensure that the interpretable model g approximates the black-box model f **locally**. The typical choice is the **weighted squared error**.
- x_i are the perturbed samples around x.
- $\pi_x(x_i)$ are their proximity weights.

DIGITAL

Molnar (2023)

LIME: **Formally**

$$\xi(x) = argmin_{g \in G} L(f, g, \pi_x) + \boldsymbol{\Omega(g)} \qquad (1)$$

- Complexity parameter.
- Prevents the local model g from being too complex.
- Encourages simpler explanations (e.g., fewer features in a linear model).
  - Example: If g is a linear model, $\boldsymbol{\Omega(g)}$ could be the number of non-zero coefficients.

DIGITAL

Molnar (2023)

LIME: **Algo**

Step 1. Initialize an empty dataset

Algorithm 1 Sparse Linear Explanations using LIME

Require: Classifier f, Number of samples N
Require: Instance x, and its interpretable version x'
Require: Similarity kernel π_x, Length of explanation K
$\mathcal{Z} \leftarrow \{\}$
for $i \in \{1, 2, 3, ..., N\}$ **do**
$z'_i \leftarrow sample\_around(x')$
$\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z'_i, f(z_i), \pi_x(z_i) \rangle$
end for
$w \leftarrow$ K-Lasso$(\mathcal{Z}, K)$ ▷ with z'_i as features, $f(z)$ as target
return w

DIGITAL

LIME: **Algo**

Algorithm 1 Sparse Linear Explanations using LIME

Require: Classifier f, Number of samples N
Require: Instance x, and its interpretable version x'
Require: Similarity kernel π_x, Length of explanation K
$\mathcal{Z} \leftarrow \{\}$
for $i \in \{1, 2, 3, ..., N\}$ **do**
$z_i' \leftarrow sample\_around(x')$
$\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z_i', f(z_i), \pi_x(z_i) \rangle$
end for
$w \leftarrow$ K-Lasso$(\mathcal{Z}, K)$ ▷ with z_i' as features, $f(z)$ as target
return w

Step 1. Initialize an empty dataset

Step 2. For each of the N samples:
- Generate perturbed sample z_i' around x' using sample_around(x')
- Get:
  - The prediction
  - The similarity $\pi_x(z_i)$

DIGITAL

LIME: **Algo**

Algorithm 1 Sparse Linear Explanations using LIME

Require: Classifier f, Number of samples N
Require: Instance x, and its interpretable version x'
Require: Similarity kernel π_x, Length of explanation K

$\mathcal{Z} \leftarrow \{\}$
for $i \in \{1, 2, 3, ..., N\}$ **do**
  $z'_i \leftarrow sample\_around(x')$
  $\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z'_i, f(z_i), \pi_x(z_i) \rangle$
end for
$w \leftarrow$ K-Lasso$(\mathcal{Z}, K)$ ▷ with z'_i as features, $f(z)$ as target
return w

Step 1. Initialize an empty dataset

Step 2. For each of the N samples:
- Generate perturbed sample z'_i around x' using sample_around(x')
- Get:
  - The prediction
  - The similarity $\pi_x(z_i)$

Step 3. Fit the weighted linear model using K-Lasso
- Use z'_i as features
- Use $f(z_i)$ as target
- Weigh each sample using the $\pi_x(z_i)$
- Restrict to k features

DIGITAL

LIME: **Algo**

Algorithm 1 Sparse Linear Explanations using LIME

Require: Classifier f, Number of samples N
Require: Instance x, and its interpretable version x'
Require: Similarity kernel π_x, Length of explanation K

$\mathcal{Z} \leftarrow \{\}$
for $i \in \{1, 2, 3, ..., N\}$ **do**
  $z'_i \leftarrow sample\_around(x')$
  $\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z'_i, f(z_i), \pi_x(z_i) \rangle$
end for
$w \leftarrow$ K-Lasso$(\mathcal{Z}, K)$ ▷ with z'_i as features, $f(z)$ as target
return w

Step 1. Initialize an empty dataset

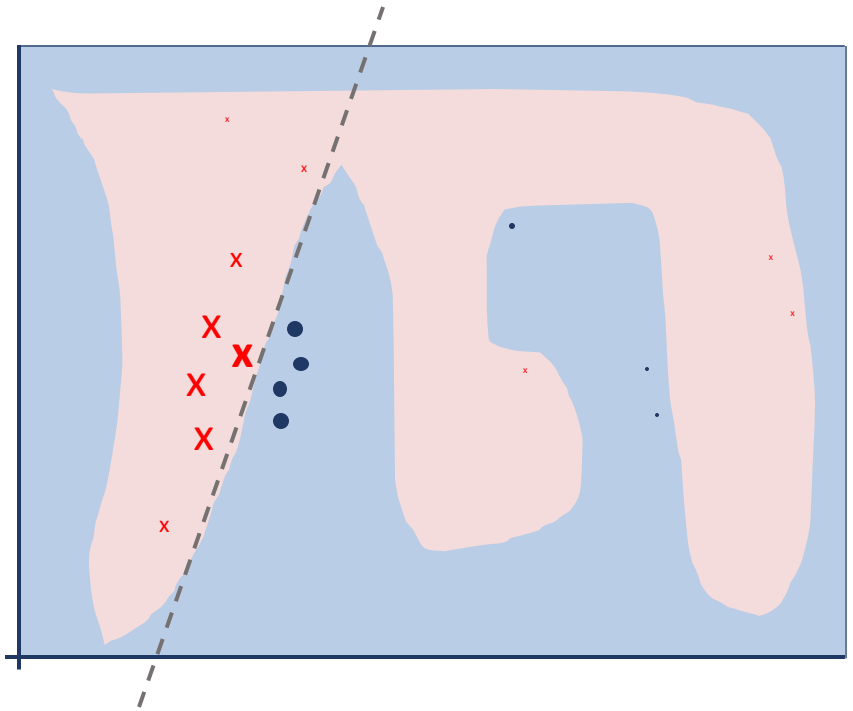
Step 2. For each of the N samples:
- Generate perturbed sample z'_i around x' using sample_around(x')
- Get:
  - The prediction
  - The similarity $\pi_x(z_i)$

Step 3. Fit the weighted linear model using K-Lasso
- Use z'_i as features
- Use $f(z_i)$ as target
- Weigh each sample using the $\pi_x(z_i)$
- Restrict to k features

Step 4. Return w: weights i.e. local explanations

DIGITAL

LIME: So, what are **all the steps**?



Pick an observation, **create and permute data**

Calculate similarity between the original observations and the permutations

Make predictions on new data using your black box

Fit a simple model to the permuted data with k features and similarity scores as weights

Coefficients from the simple model serve as an explanation of the model behavior at the local level

DIGITAL

Core problem

- **LIME is popular but fragile.**
- LIME explanations can change substantially depending on:
  - how images are **segmented** into superpixels,
  - how **perturbations** are generated,
  - which **interpretable surrogate model** is fitted,
  - random sampling effects.

- This instability is especially problematic in **high-stakes domains like medical imaging**, where explanations are used to support trust and decision-making.

- The key issue: ***Users usually see only one LIME explanation, without understanding how sensitive it is to modeling choices.***

DIGITAL

Laguna, S., Heidenreich, J. N., Sun, J., Cetin, N., Al-Hazmi, I., Schlegel, U., Cheng, F., & El-Assady, M. (2023). ExplLIMEable: An exploratory framework for LIME. In XAI in Action: Past, Present, and Future Applications (NeurIPS 2023 Workshop).

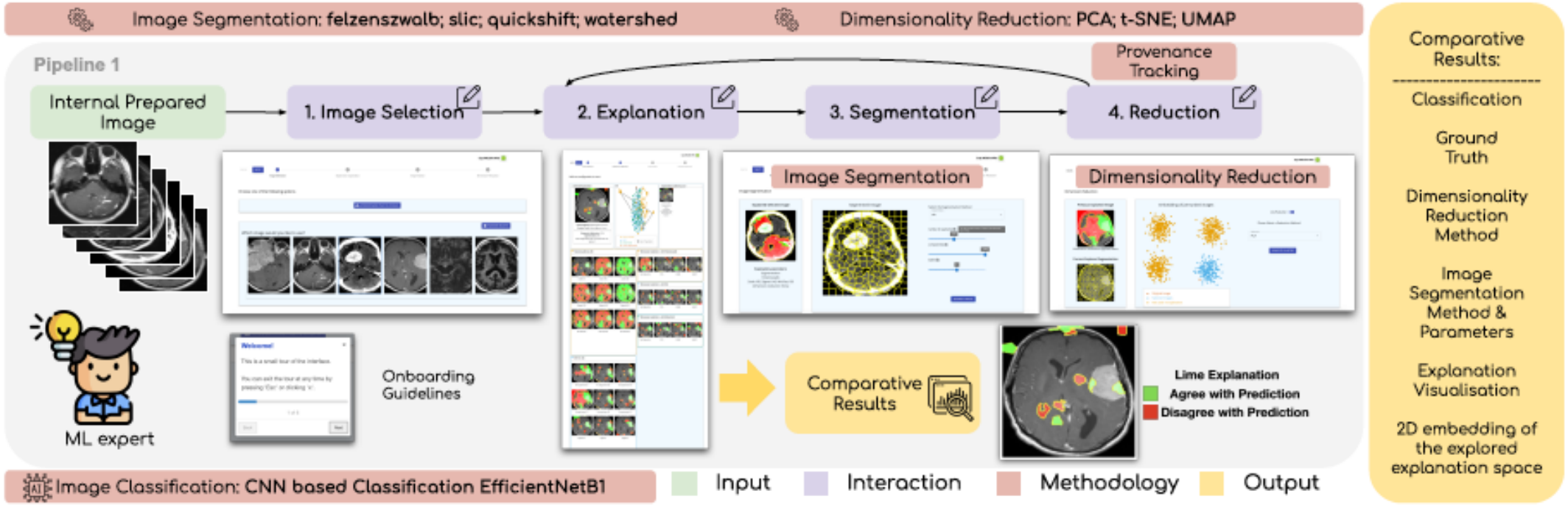
ExpLIMEable

- The authors propose **ExpLIMEable**, an **interactive visual analytics framework** that allows users to:
- **Systematically explore the space of LIME explanations**
- **Understand how parameter choices affect explanations**
- **Track provenance** of explanation decisions
- **Improve robustness** by controlling which perturbations are used
- Rather than "fixing" LIME algorithmically, they **make its weaknesses explicit and explorable**.

DIGITAL

Laguna, S., Heidenreich, J. N., Sun, J., Cetin, N., Al-Hazmi, I., Schlegel, U., Cheng, F., & El-Assady, M. (2023). ExplLIMEable: An exploratory framework for LIME. In XAI in Action: Past, Present, and Future Applications (NeurIPS 2023 Workshop).

What ExpLIMEable actually does?



DIGITAL

Laguna, S., Heidenreich, J. N., Sun, J., Cetin, N., Al-Hazmi, I., Schlegel, U., Cheng, F., & El-Assady, M. (2023). ExplLIMEable: An exploratory framework for LIME. In XAI in Action: Past, Present, and Future Applications (NeurIPS 2023 Workshop).

LIME vs **ExpLIMEable**

Standard LIME problem:

- Perturbations are sampled randomly
- Many samples are unrealistic or uninformative
- The "local neighborhood" is poorly defined

This is *not* changing LIME's regression step — it changes **what data LIME learns from**.

ExpLIMEable solution:

- Generate many perturbed images
- Project them into a **low-dimensional space** using:
  - PCA
  - t-SNE
  - UMAP
- **Cluster the perturbations**
- Keep only perturbations **close to the original image** in embedding space
- This creates a **more informative and controlled local neighborhood**.

DIGITAL

Laguna, S., Heidenreich, J. N., Sun, J., Cetin, N., Al-Hazmi, I., Schlegel, U., Cheng, F., & El-Assady, M. (2023). ExplLIMEable: An exploratory framework for LIME. In XAI in Action: Past, Present, and Future Applications (NeurIPS 2023 Workshop).

DIGITAL

Funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Horizon Europe: Marie Skłodowska-Curie Actions. Neither the European Union nor the granting authority can be held responsible for them.

DIGITAL

This project has received funding from the European Union's Horizon Europe research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101119635