Explanations for ML Models LIME and ANCHOR

Keyu Li and Jason Lee

March 28, 2023

CompSci 590.01presentation

Duke Computer Science

LIME: “Why Should I Trust You?” Explaining the Predictions of Any Classifier

Ribeiro et al. (2016)

Table of Content

Why

Why?

Current Problem: Black Box Model vs. Trustworthy Model

- Machine learning models: mostly black boxes
  - “Powerful, high performance?” “Yes!”
  - “Do you trust this model?” “Umm…”
  - “Do you really want to use this model for decision making?” “Not really…”

Why?

Current Problem: Black Box Model vs. Trustworthy Model

- Machine learning models: mostly black boxes
  - "Powerful, high performance?" "Yes!"
  - "Do you trust this model?" "Umm…"
  - "Do you really want to use this model for decision making?" "Not really…"
- Problem? Yes!
  - The important role of humans is often overlooked
    - *If the users do not trust a model or prediction, they will not use it*
      - E.g. Using ML classifiers as tools/deploying models within other products

Why?

Current Problem: Black Box Model vs. Trustworthy Model

- Machine learning models: mostly black boxes
    - “Powerful, high performance?” “Yes!”
    - “Do you trust this model?” “Umm…”
    - “Do you really want to use this model for decision making?” “Not really…”
- Problem? Yes!
    - The important role of humans is usually oft-overlooked
        - *If the users do not trust a model or prediction, they will not use it*
            - E.g. Using ML classifiers as tools/deploying models within other products
- Definition of “Trust”?
    - 1) Trusting a prediction
        - i.e. whether a user trust an individual prediction sufficiently to take some action based on it

Why?

Current Problem: Black Box Model vs. Trustworthy Model

- Machine learning models: mostly black boxes
  - “Powerful, high performance?” “Yes!”
  - “Do you trust this model?” “Umm…”
  - “Do you really want to use this model for decision making?” “Not really…”
- Problem? Yes!
  - The important role of humans is usually oft-overlooked
    - *If the users do not trust a model or prediction, they will not use it*
      - E.g. Using ML classifiers as tools/deploying models within other products
- Definition of “Trust”?
  - 1) Trusting a prediction
    - i.e. whether a user trust an individual prediction sufficiently to take some action based on it
  - 2) Trusting a model
    - i.e. whether the user trust a model to behave in reasonable ways if deployed

Why?

Current Problem: Black Box Model vs. Trustworthy Model

- Definition of “Trust” - Examples and Explanation
    - 1) Trusting a prediction
        - i.e. whether a user trust an individual prediction sufficiently to take some action based on it
        - E.g. Medical diagnosis

Why?

Current Problem: Black Box Model vs. Trustworthy Model

- Definition of “Trust” - Examples and Explanation
    - 1) Trusting a prediction
        - i.e. whether a user trust an individual prediction sufficiently to take some action based on it
        - E.g. Medical diagnosis
        - E.g. Terrorism detection

Why?

Current Problem: Black Box Model vs. Trustworthy Model

- Definition of “Trust” - Examples and Explanation
  - 1) Trusting a prediction
    - i.e. whether a user trust an individual prediction sufficiently to take some action based on it
    - E.g. Medical diagnosis
    - E.g. Terrorism detection
  - 2) Trusting a model
    - i.e. whether the user trust a model to behave in reasonable ways if deployed

Why?

The Case for Explanations

- 1) Trusting a prediction

Question:
Generally speaking,
what can make you trust a
prediction?

The Case for Explanations

- 1) Trusting a prediction
  - Qualitative understanding of the relationship between the instance's components and the model's prediction.

Why?

The Case for Explanations

- 1) Trusting a prediction
  - Qualitative understanding of the relationship between the instance’s components and the model’s prediction.
  - Example

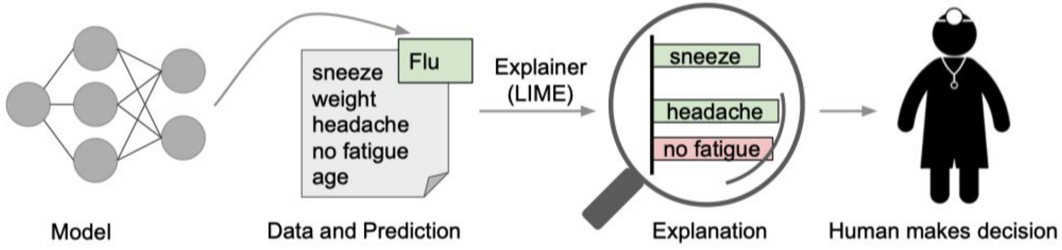


Figure 1

Why?

The Case for Explanations

- 2) Trusting a model
    - ML applications: requires a certain measure of overall trust
    - ML practitioners: “a lot of alternatives…which one to choose?”
        - Example

Why?

The Case for Explanations

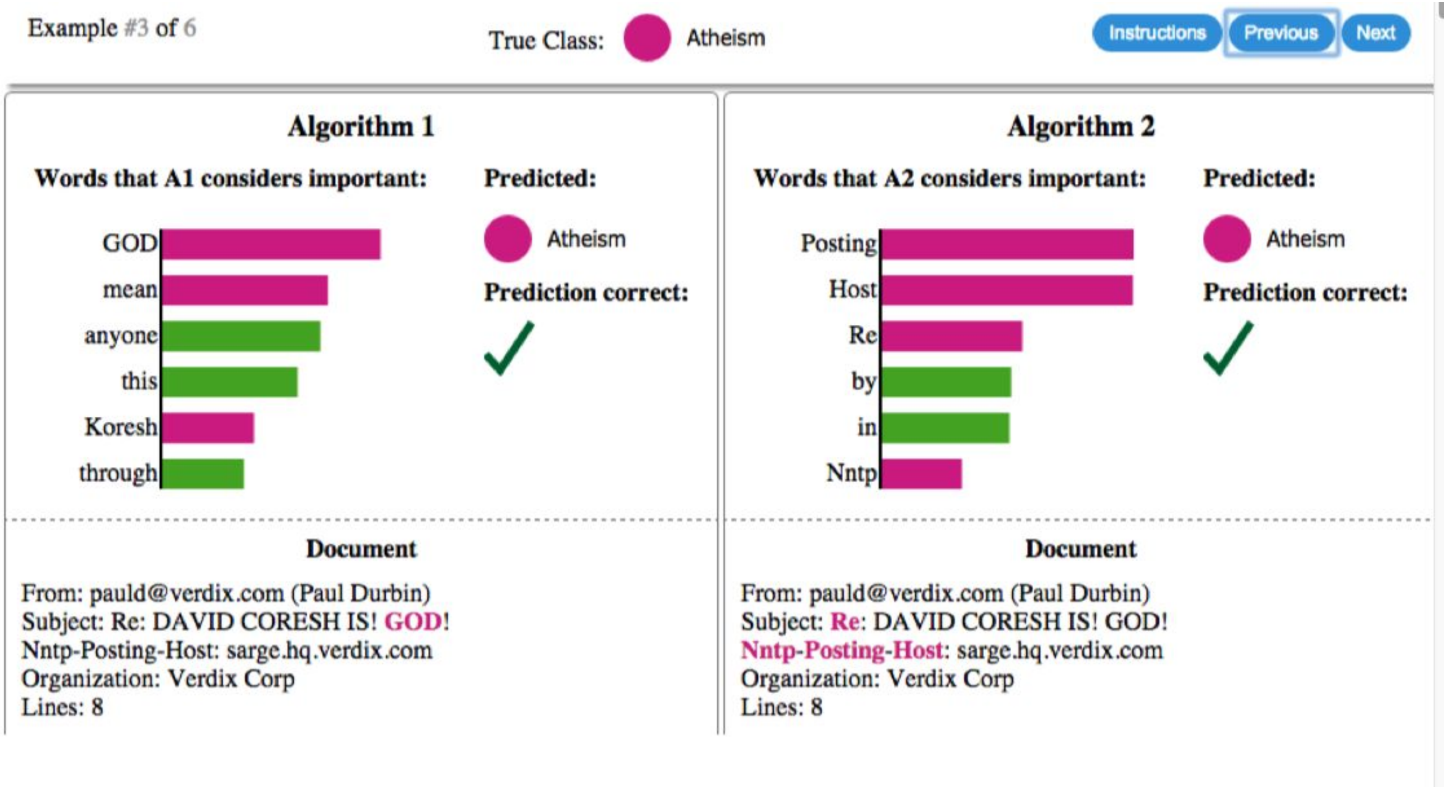


Figure 2

Why?

The Case for Explanations

Question:
Which one is better?

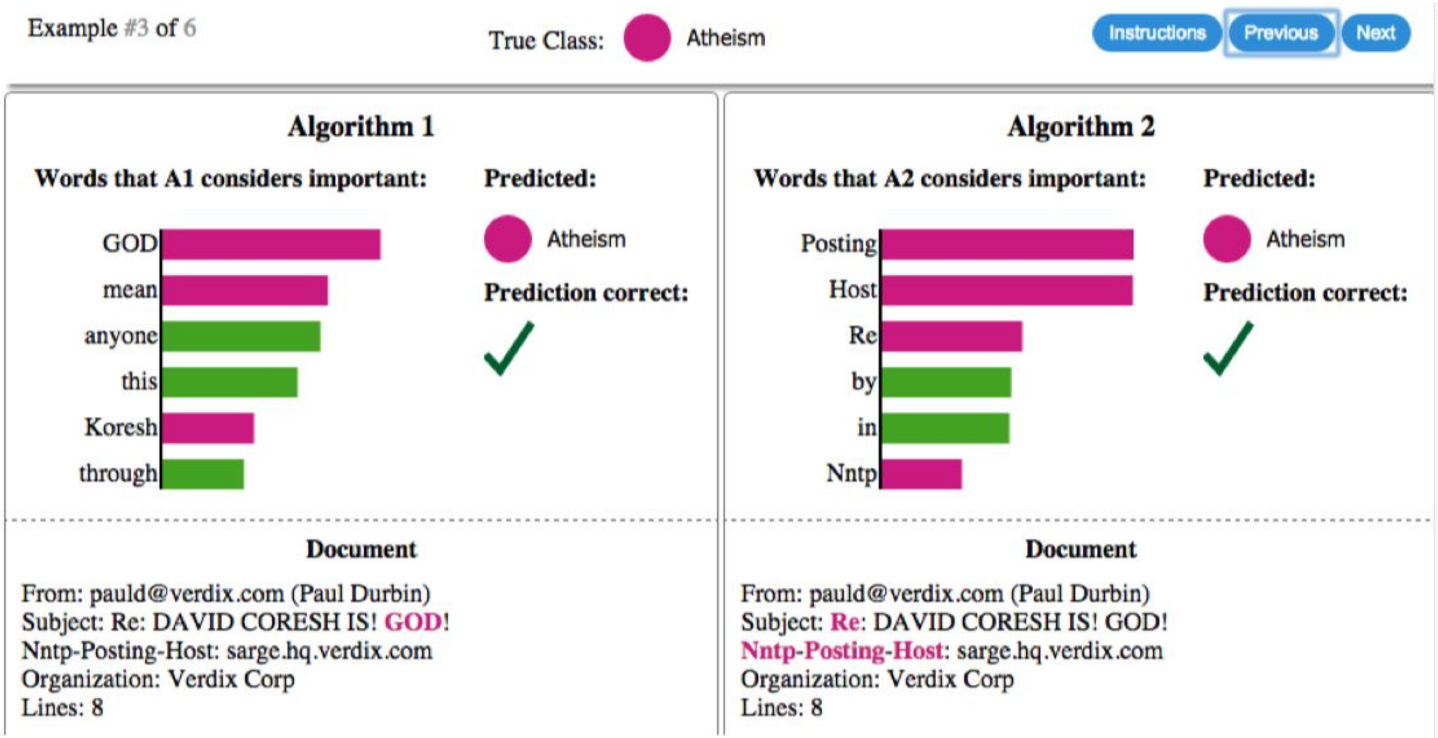


Figure 2

Why?

Desired Characteristics for Explainers

- Interpretable
    - i.e. provide qualitative understanding **between the input variables and the response**
    - Must consider the user's **limitations**
        - A linear model may not be interpretable…

Why?

Desired Characteristics for Explainers

- Interpretable
  - i.e. provide qualitative understanding **between the input variables and the response**
  - Must consider the user's **limitations**
    - A linear model may not be interpretable
- Local fidelity
  - i.e. explanation must correspond to how the model behaves **in the vicinity of** the instance being predicted

Desired Characteristics for Explainers

- Interpretable
  - i.e. provide qualitative understanding **between the input variables and the response**
  - Must consider the user's **limitations**
    - A linear model may not be interpretable
- Local fidelity
  - i.e. explanation must correspond to how the model behaves **in the vicinity of** the instance being predicted
- Model-agnostic
  - An explainer should be able to explain ***any*** model

Why?

Desired Characteristics for Explainers

- Interpretable
  - i.e. provide qualitative understanding **between the input variables and the response**
  - Must consider the user's **limitations**
    - A linear model may not be interpretable
- Local fidelity
  - i.e. explanation must correspond to how the model behaves **in the vicinity of** the instance being predicted
- Model-agnostic
  - An explainer should be able to explain ***any*** model
- Global perspective
  - A good explainer should provide a global perspective so as to ascertain trust in the model
    - *Explain **the model***

What

What?

Proposed Solution

- Trustworthy predictions
    - **Local Interpretable Model-Agnostic Explanations (LIME)**
    - Overall goal of LIME: identify an **interpretable** model over the ***interpretable representation*** that is **locally faithful** to the classifier

What?

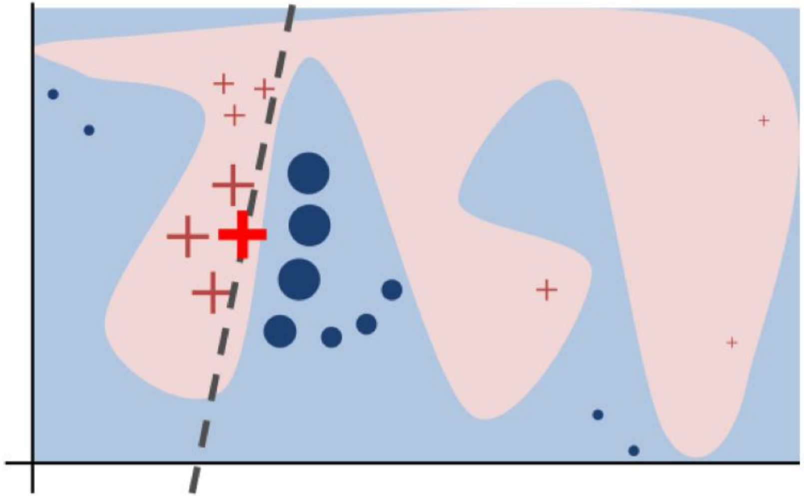
Proposed Solution

- For trustworthy predictions
  - **Local Interpretable Model-Agnostic Explanations (LIME)**
  - Overall goal of LIME: identify an **interpretable** model over the *interpretable representation* that is **locally faithful** to the classifier
- For trustworthy models
  - Submodular Optimization-LIME (SP-LIME)
  - Key ideas of SP-LIME: Choose **a set of representative instances with explanations** to address the “trust the model” problem, via **submodular optimization**

What?

Local Interpretable Model-Agnostic Explanations(LIME)

- Key idea in one sentence
  - For a prediction of a given individual, we learn **a explainable model** (e.g. linear models, decision trees) that uses **interpretable representations** to generate the prediction which **mimic** the **local behaviors** of the original black-box model (in terms of prediction results), while **controlling the complexity** of the learned explainable model.



What?

Submodular Pick For Explaining Models(SP-LIME)

- Key idea in one sentence
    - **Wait ! ! What is submodular ? ?**
        - **Submodular optimization is…**

What?

Submodular Pick For Explaining Models(SP-LIME)

- Key idea in one sentence
    - **Wait ! ! What is submodular ? ?**
        - **Submodular optimization is…**
    - **Key idea:**
        - **Based on the explanations** that accompany each prediction, this method pick a **diverse**, **representative** set of explanations to show the user - i.e. **non-redundant** explanations that represent how the model behaves **globally**.

How

A Closer Look: A Toy Example (LIME)

- Text classifier:
  - Classify a given personal comment to “good” (as 1) or “bad” (as 0).
  - Learned black-box classifier: f
- **Interpretable** Data Representation

x

x’

Raw Input:

“You are a very nice person” (Label: 1)

Features: Word embedding (Feed into the original model)

(0.123, -0.982), (-0.672, 0.251), (0.464, 0.294), (0.456, -0.627), (0.111, 0.957), (-0.832, -0.517)

Interpretable representation: Binary vector indicating the presence or absence of a word (Feed into the explainable model)

0 0 0 0 0 0

How?

A Closer Look: A Toy Example (LIME)

- **Sampling** for local exploration
    - Sampling instances around x’ by drawing nonzero elements of x’ uniformly at random

Raw Input:

“You are a very nice person” (Label: 1)

Features: Word embedding (Feed into the original model) x

(0.123, -0.982), (-0.672, 0.251), (0.464, 0.294), (0.456, -0.627), (0.111, 0.957), (-0.832, -0.517)

Interpretable representation: Binary vector indicating the presence or absence of a word (Feed into the explainable model) x’

0 0 0 0 0 0

A Closer Look: A Toy Example (LIME)

- **Sampling** for local exploration
  - Sampling instances around x' by drawing nonzero elements of x' uniformly at random

Raw Input:

"You are a very nice person" (Label: 1)

Features: Word embedding (Feed into the original model)

(0.123, -0.982), (-0.672, 0.251), (0.464, 0.294), (0.456, -0.627), (0.111, 0.957), (-0.832, -0.517)

X

Interpretable representation: Binary vector indicating the presence or absence of a word (Feed into the explainable model)

0 0 0 0 0 0

X'

N: # of samples. Assume N=5 here
K: length of explanation. Assume K=3 here

Number of such draw: Uniformly sampled. Assume ~ U(2,4) here

A Closer Look: A Toy Example (LIME)

- Sampling for local exploration
  - Sampling instances around x' by drawing nonzero elements of x' uniformly at random

Raw Input:

"You are a very nice person"
(Label: 1)

Interpretable representation:
Binary vector indicating the presence or absence of a word

0 0 0 0 0 0

x'

sampling →

Perturbed sample:
Interpretable binary vector indicating the presence or absence of a word (Feed into the explainable model)

z1': 0 1 1 0 0 0

z2': 0 0 1 1 1 0

z3': 1 0 0 1 0 0

z4': 1 1 1 1 0 0

z5': 0 1 0 1 1 0

N: # of samples. Assume N=5 here
K: length of explanation. Assume K=3 here
Number of such draw: Uniformly sampled. Assume ~ U(2,4)

How?

A Closer Look: A Toy Example (LIME)

- Sampling for local exploration
  - Sampling instances around x’ by drawing nonzero elements of x’ uniformly at random

Raw Input:

“You are a very nice person” (Label: 1)

Interpretable representation:
Binary vector indicating the presence or absence of a word

0 0 0 0 0 0

x’

sampling →

Perturbed sample:
Interpretable binary vector indicating the presence or absence of a word (Feed into the explainable model)

z1’: 0 1 1 0 0 0

z2’: 0 0 1 1 1 0

z3’: 1 0 0 1 0 0

z4’: 1 1 1 1 0 0

z5’: 0 1 0 1 1 0

N: # of samples. Assume N=5 here
K: length of explanation. Assume K=3 here
Number of such draw: Uniformly sampled. Assume ~ U(2,4)
z: The features of the corresponding z’ → f(z)
E.g. z’ = 111100 → z: word vector of “You are a very”
$\pi_x(z)$: **Proximity measure** between an instance z to x (define **the locality around x**)

How?

A Closer Look: A Toy Example (LIME)

- Sampling for local exploration
  - Sampling instances around x' by drawing nonzero elements of x' uniformly at random

Raw Input:

"You are a very nice person" (Label: 1)

Interpretable representation:
Binary vector indicating the presence or absence of a word

0 0 0 0 0 0

x'

sampling →

Perturbed sample:
Interpretable binary vector indicating the presence or absence of a word (Feed into the explainable model)

z1': 0 1 1 0 0 0

z2': 0 0 1 1 1 0

z3': 1 0 0 1 0 0

z4': 1 1 1 1 0 0

z5': 0 1 0 1 1 0

Z ← {}

Z ← Z U (z1', f(z1), πx(z1)).

Z ← Z U (z2', f(z2), πx(z2)).

Z ← Z U (z3', f(z3), πx(z3)).

Z ← Z U (z4', f(z4), πx(z4)).

Z ← Z U (z5', f(z5), πx(z5)).

N: # of samples. Assume N=5 here
K: length of explanation. Assume K=3 here
Number of such draw: Uniformly sampled. Assume ~ U(2,4)
z: The features of the corresponding z'→ **f(z)**
E.g. z' = 111100 → z: word vector of "You are a very"
$\pi_x(z)$: Proximity measure between an instance z to x (define **the locality around x**)

How?

A Closer Look: A Toy Example

N: # of samples. Assume N=5 here
K: length of explanation. Assume K=3 here
Number of such draw: Uniformly sampled. Assume ~ U(2,4)
z: The features of the corresponding z'
E.g. z' = 111100 → z: "You are a very"
$\pi_x(z)$: Proximity measure between an instance z to x (define **the locality around x**)

- **Sparse Linear Explanations**

Explainable model g:
Choose linear models here
Dataset:
Z (contains data & label & additional distance metric)

Raw Input:

"You are a very nice person" (Label: 1)

Perturbed sample:
Interpretable binary vector indicating the presence or absence of a word (Feed into the explainable model)

z1': 0 1 1 0 0 0

z2': 0 0 1 1 1 0

z3: 1 0 0 1 0 0

z4': 1 1 1 1 0 0

z5': 0 1 0 1 1 0

Z ← {}

Z ← Z U (z1', f(z1), πx(z1)).

Z ← Z U (z2', f(z2), πx(z2)).

Z ← Z U (z3', f(z3), πx(z3)).

Z ← Z U (z4', f(z4), πx(z4)).

Z ← Z U (z5', f(z5), πx(z5)).

How?

A Closer Look: A Toy Example

N: # of samples. Assume N=5 here
K: length of explanation. Assume K=3 here
Number of such draw: Uniformly sampled. Assume ~ U(2,4)
z: The features of the corresponding z'
E.g. z' = 111100 → z: "You are a very"
$\pi_x(z)$: Proximity measure between an instance z to x (define **the locality around x**)

- **Sparse Linear Explanations**

Explainable model g:
Choose linear models here
Dataset:
Z (contains data & label & additional distance metric)
Objective function:

$$\min \underbrace{\mathcal{L}(f, g, \pi_x)}_{\text{Local fidelity}} + \underbrace{\Omega(g)}_{\text{Interpretability}}$$

Ω(g): A measure of complexity (as opposed to interpretability)

$$\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z) \left(f(z) - g(z')\right)^2$$

Explanation:

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \ \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

(Corresponding weight for each feature)

Raw Input:

"You are a very nice person"
(Label: 1)

Perturbed sample:
Interpretable binary vector indicating the presence or absence of a word (Feed into the explainable model)

z1': 0 1 1 0 0 0
z2': 0 0 1 1 1 0
z3: 1 0 0 1 0 0
z4': 1 1 1 1 0 0
z5': 0 1 0 1 1 0

Z ← {}
Z ← Z U (z1', f(z1), πx(z1)).
Z ← Z U (z2', f(z2), πx(z2)).
Z ← Z U (z3', f(z3), πx(z3)).
Z ← Z U (z4', f(z4), πx(z4)).
Z ← Z U (z5', f(z5), πx(z5)).

How?

A Closer Look: A Toy Example

N: # of samples. Assume N=5 here
K: length of explanation. Assume K=3 here
Number of such draw: Uniformly sampled. Assume ~ U(2,4)
z: The features of the corresponding z'
E.g. z' = 111100 → z: "You are a very"
$\pi_x(z)$: Proximity measure between an instance z to x (define **the locality around x**)

- Sparse Linear Explanations

Explainable model g:
Choose linear models here
Dataset:
Z (contains data & label & additional distance metric)
Objective function:

Local fidelity Interpretability

$$\min \quad \underbrace{\mathcal{L}(f, g, \pi_x)} + \underbrace{\Omega(g)}$$

Ω(g): A measure of complexity (as opposed to interpretability)

$$\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z) \left(f(z) - g(z')\right)^2$$

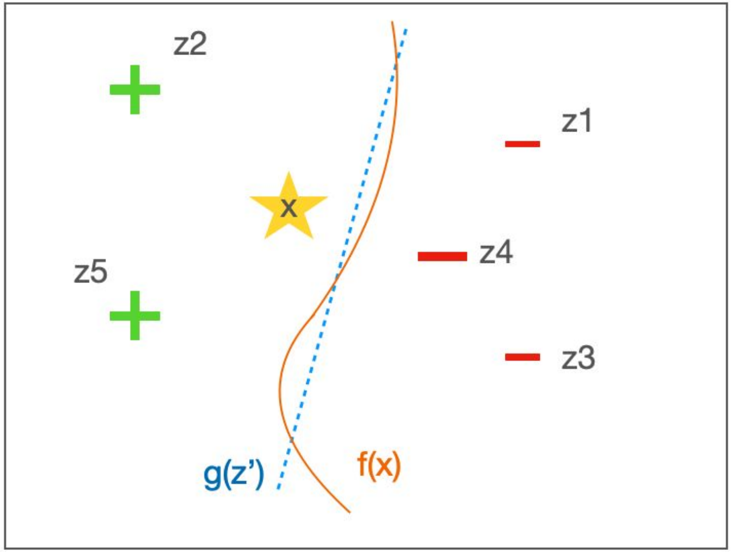
Explanation:

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \quad \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

(Corresponding weight for each feature)

For the toy example:
Choose K-Lasso to limit # of explanations (K=3), i.e. we can only choose up to 3 words here for explanation

Explainable model g vs original model f



Explanation:

Nice 0.96
Very 0.56
You 0.43

A Closer Look: A Toy Example (LIME)

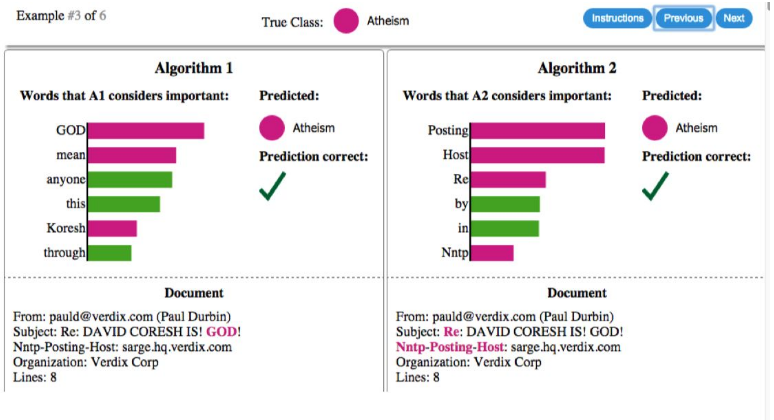
- LIME Algorithm

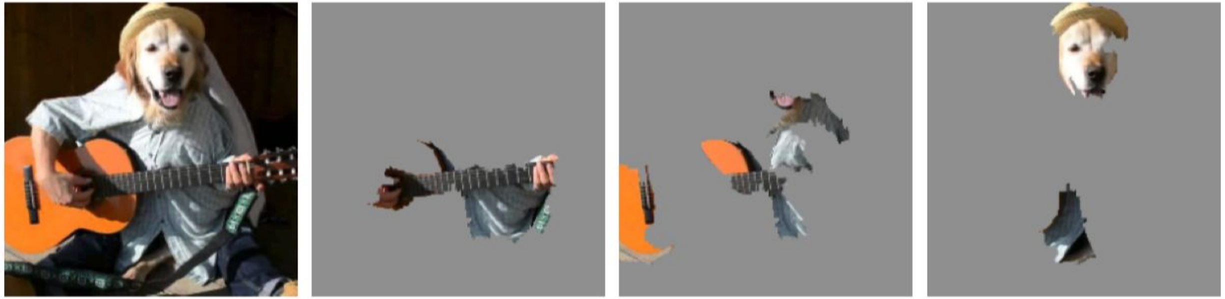
Algorithm 1 Sparse Linear Explanations using LIME

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

A Closer Look: A Toy Example (LIME)

- More examples





(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

A Closer Look: A Toy Example (SP-LIME)

- Submodular Pick for Explaining Models
    - Explanations generated for x1, x2,...xn
    - Key idea: Pick a **diverse**, **representative** set of **explanations** to show the user

A Closer Look: A Toy Example (SP-LIME)

- Submodular Pick for Explaining Models
  - Explanations generated for x1, x2,...xn
  - Key idea: Pick a diverse, representative set of explanations to show the user
- How to find such a set
  - We don't want to show hundreds of instances (What a nightmare…)
    - **Budget** is needed: B

A Closer Look: A Toy Example (SP-LIME)

- Submodular Pick for Explaining Models
  - Explanations generated for x1, x2,...xn
  - Key idea: Pick a diverse, representative set of explanations to show the user
- How to find such a set
  - We don’t want to show hundreds of instances (What a nightmare…)
    - **Budget** is needed: B
  - We don’t want a **redundant** explanation set
    - Redundant…?
  - We want to have a **representative** set of explanations
    - Create **Importance function**: I

A Closer Look: A Toy Example (SP-LIME)

- Should have budget: B
- Should maximize the interpretable representation diversity
- Should have a representative set
    - Using importance function: I

A Closer Look: A Toy Example (SP-LIME)

- Should have **budget**: B
- Should maximize the interpretable representation **diversity**
- Should have a **representative** set
  - Using **importance function**: I

Explanation matrix: W (n*d’)
Represent the local importance of the interpretable components for each instance

How?

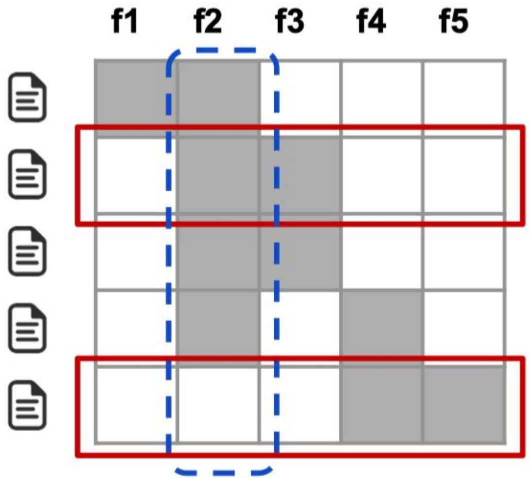
A Closer Look: A Toy Example (SP-LIME)

- Should have budget: B
- Should maximize the interpretable representation diversity
- Should have a representative set
    - Using importance function: I

Explanation matrix: W (n*d')
Represent the local importance of the interpretable components for each instance

<u>Toy example:</u>

Row: Different sentences/text (Individual instance)
Column: Features



A Closer Look: A Toy Example (SP-LIME)

- Formalize the **non-redundant coverage** with **importance function**

$$c(V, \mathcal{W}, I) = \sum_{j=1}^{d'} \mathbb{1}_{[\exists i \in V : \mathcal{W}_{ij} > 0]} I_j$$

How?

A Closer Look: A Toy Example (SP-LIME)

- Formalize the non-redundant coverage with importance function

$$c(V, \mathcal{W}, I) = \sum_{j=1}^{d'} \mathbb{1}_{[\exists i \in V : \mathcal{W}_{ij} > 0]} I_j$$

- Pick problem:
  - Goal is to **achieve highest coverage**

$$Pick(\mathcal{W}, I) = \underset{V, |V| \leq B}{\operatorname{argmax}} \, c(V, \mathcal{W}, I)$$

A Closer Look: A Toy Example (SP-LIME)

- Submodular Pick for Explaining Models

Algorithm 2 Submodular pick (SP) algorithm

Require: Instances X, Budget B
  for all $x_i \in X$ **do**
    $\mathcal{W}_i \leftarrow$ **explain**(x_i, x'_i) ▷ Using Algorithm 1
  end for
  for $j \in \{1 \ldots d'\}$ **do**
    $I_j \leftarrow \sqrt{\sum_{i=1}^{n} |\mathcal{W}_{ij}|}$ ▷ Compute feature importances
  end for
  $V \leftarrow \{\}$
  while $|V| < B$ **do** ▷ Greedy optimization of Eq (4)
    $V \leftarrow V \cup \mathrm{argmax}_i\, c(V \cup \{i\}, \mathcal{W}, I)$
  end while
  return V

Demonstration

Demonstration

Experiments

- Stimulated user experiment
  - **Are explanation faithful to the model?**

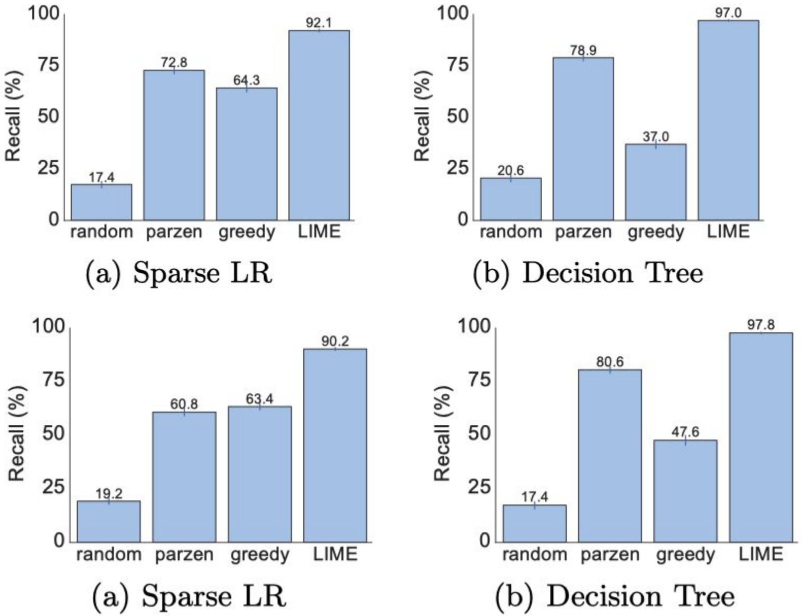


Figure 3

- **Should I trust this prediction?**

	Books				DVDs			
	LR	NN	RF	SVM	LR	NN	RF	SVM
Random	14.6	14.8	14.7	14.7	14.2	14.3	14.5	14.4
Parzen	84.0	87.6	94.3	92.3	87.0	81.7	94.2	87.3
Greedy	53.7	47.4	45.0	53.3	52.4	58.1	46.6	55.1
LIME	**96.6**	**94.5**	**96.2**	**96.7**	**96.6**	**91.8**	**96.1**	**95.6**

Table 1

- **Can I trust this model?**

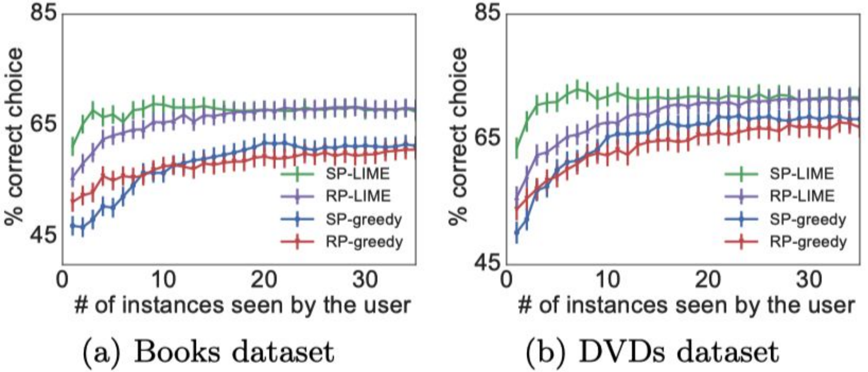


Figure 4

Experiments

- Evaluation with human subjects
  - **Can user select the best classifier?**

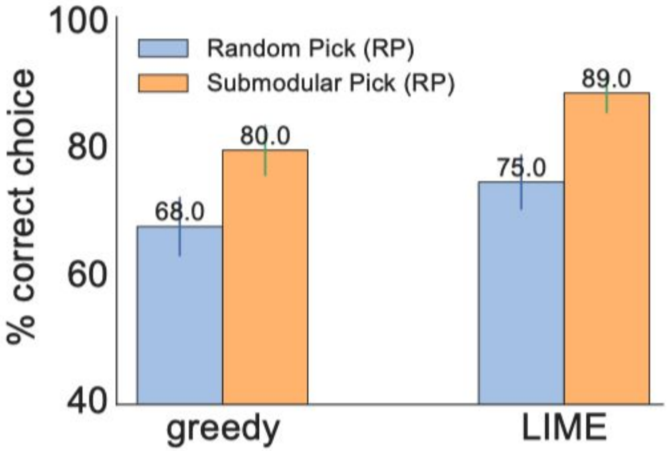


Figure 5

- **Can non-experts improve a classifier?**

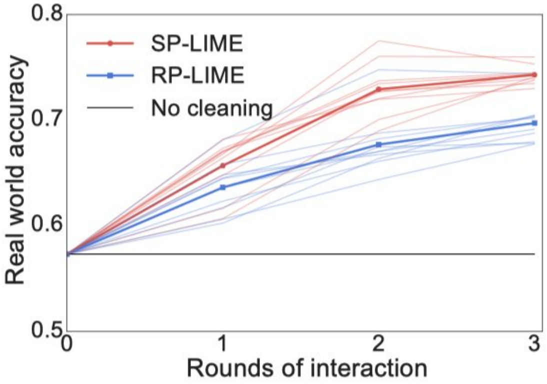


Figure 6

- **Do explanations lead to insights?**

(a) Husky classified as wolf (b) Explanation

Figure 7

	Before	After
Trusted the bad model	10 out of 27	3 out of 27
Snow as a potential feature	12 out of 27	25 out of 27

Table 2

Conclusion

- Argument
  - Argued that trust is crucial for effective human interaction with ML systems
  - Explaining individual predictions is important in assessing trust
- Proposed LIME:
  - a modular and extensible approach to faithfully explain the predictions of *any* model in an interpretable manner
- Introduced SP-LIME:
  - a method to select representative and non-redundant predictions, providing a global view of the model to users

Future Work

- A comparative study on **different explainable models** with **real** users
- Address the limitation of how to perform the **pick up for images**
- **Explore a variety of applications** like speech, video, recommendation systems, and medical domains
- Explore **theoretical properties** (such as the appropriate number of samples) and **computational optimization**

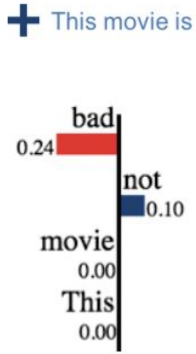
Anchors: High-Precision Model-Agnostic Explanations

Ribeiro et al. (2018)

Problem with LIME

- unclear **coverage** ← unclear when an explanation applies to unseen case
    - e.x. the word ‘not’ have opposite meanings based on its context
- this leads to worse human **precision**

Explanation given for sentiment analysis:



User prediction on unseen case:

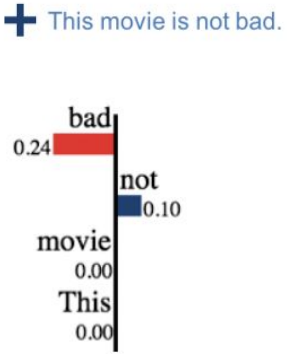
“This movie is not very good”

What can go wrong?

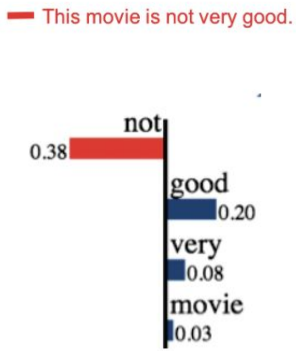
Problem with LIME

- unclear **coverage** ← unclear when an explanation applies to unseen case
    - e.x. the word 'not' have opposite meanings based on its context
- this leads to worse human **precision**

Explanation given for sentiment analysis:



User prediction on unseen case:



Introducing… Anchors

- Anchors: a **rule** (if… then statements) such that when the anchor holds, the prediction stays the same with high probability

Example anchor for given instance

Why Anchors?

1. Intuitive: Easy for users to understand if… then statements
2. Clear Coverage: Very clear when & where explanations apply
3. High Precision: By design, users can accurately predict model behavior

Example anchor for given instance

Notations 😫

Given a black box model $f : X \rightarrow Y$ and instance $x \in X$, the goal of explanations is to explain the behavior of $f(x)$ to a user.

In LIME, we obtain local, model-agnostic explanations by perturbing instance **x** using perturbation distribution $\mathcal{D}_{\mathbf{x}}$ (noted as $\mathcal{D}$ for simplicity)

instance x

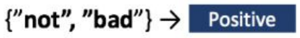
perturbation $\mathcal{D}$

Notations 😫

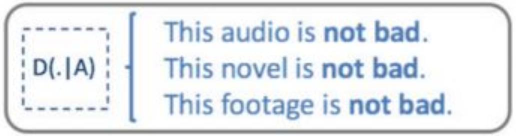
Given a black box model f : X → Y and instance x ∈ X, the goal of explanations is to explain the behavior of f(x) to a user.

For given instance **x**, rule (predicate set) **A**, and conditional distribution **𝒟(•|A)**, we can obtain perturbed samples **z** where **A** still holds

instance x,
where f(x) = positive



Rule A, where A(x) = 1 if rule applies for x



conditional distribution 𝒟 (•|A) ← set of perturbed samples where A applies

Theoretical Definition of Anchors

Anchors*: a **rule** (if… then statements) such that <u>when the anchor holds</u>, the <u>prediction stays the same</u> with high probability*

- Rule **A** is an anchor if a) the rule applies for the given instance **x**, and b) if the model prediction stays the same for <u>most</u> perturbed samples **z** from $\mathcal{D}$(z|A)

$$\mathbb{E}_{\mathcal{D}(z|A)}[\mathbb{1}_{f(x)=f(z)}] \geq \tau, A(x) = 1.$$

for each perturbation of x
where rule A applies

Theoretical Definition of Anchors

Anchors*: a **rule** (if… then statements) such that <u>when the anchor holds</u>, the <u>prediction stays the same</u> with high probability*

- Rule **A** is an anchor if a) the rule applies for the given instance **x**, and b) if the model prediction stays the same for <u>most</u> perturbed samples **z** from $\mathcal{D}$(z|A)

$$\mathbb{E}_{\mathcal{D}(z|A)}[\mathbb{1}_{f(x)=f(z)}] \geq \tau, A(x) = 1.$$

the prediction for the perturbed samples f(z) <u>stays the same</u> e.x. = f(x) with probability greater than **τ**

Theoretical Definition of Anchors

Anchors*: a **rule** (if… then statements) such that <u>when the anchor holds</u>, the <u>prediction stays the same</u> with high probability*

- Rule **A** is an anchor if a) the rule applies for the given instance **x**, and b) if the model prediction stays the same for <u>most</u> perturbed samples **z** from $\mathcal{D}$(z|A)

$$\mathbb{E}_{\mathcal{D}(z|A)}[\mathbb{1}_{f(x)=f(z)}] \geq \tau, A(x) = 1.$$

and obviously, the rule applies for x itself

Example Anchors — POS tagging

- Explanations for **part-of-speech tagging** of the word “play”
- We want to explain why “play” was classified as verb or noun
- Define “**predicate set**” to be the part of speech of neighbouring words

Instance	If	Predict
I want to play(V) ball.	previous word is PARTICLE	play is VERB.
I went to a play(N) yesterday.	previous word is DETERMINER	play is NOUN.
I play(V) ball on Mondays.	previous word is PRONOUN	play is VERB.

Table 1: Anchors for Part-of-Speech tag for the word “play”

Example Anchors — Machine Translation

- Explanations for **english-to-portuguese translation** of word "this"
- We want to explain why the word "this" was translated into esta, este, or isso
- Define "**predicate set**" to be presence/absence of specific tokens

English	Portuguese
This is the **question** we must address	Esta é a questão que temos que enfrentar
This is the **problem** we must address	Este é o problema que temos que enfrentar
This is what we must address	É isso que temos de enfrentar

Table 2: Anchors (in bold) of a machine translation system for the Portuguese word for "This" (in pink).

Example Anchors — Tabular Datasets (Classic ML)

- Explanations for **ML prediction** (e.x. predict income, recidivism, loan)
- We want to explain why an individual was classified into a specific label
- Define "**predicate set**" to be features in the machine learning model

	If	**Predict**
adult	No capital gain or loss, never married	$\leq$ 50K
	Country is US, married, work hours $>$ 45	$>$ 50K
rcdv	No priors, no prison violations and crime not against property	Not rearrested
	Male, black, 1 to 5 priors, not married, and crime not against property	Re-arrested
lending	FICO score $\leq$ 649	Bad Loan
	649 $\leq$ FICO score $\leq$ 699 and $\$5,400 \leq$ loan amount $\leq \$10,000$	Good Loan

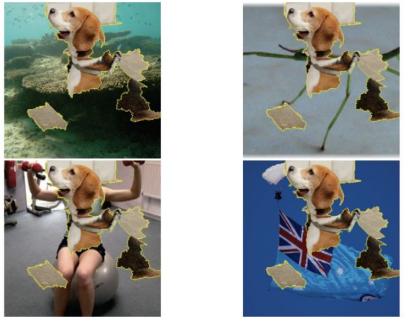
Table 3: Generated anchors for Tabular datasets

Example Anchors — Image Classification

- Explanations for **computer vision prediction** (e.x. is the image of a beagle?)
- We want to explain which superpixels of the image was relevant
- Define "**predicate set**" to be a set of superpixels
  - Unlike LIME, we superimpose a set of superpixels onto a random image and determine if prediction on superimposed image meets precision criterion.

(a) Original image

(b) Anchor for "beagle"



(c) Images where Inception predicts $P(\text{beagle}) > 90\%$

Our definition of anchors ensures high precision

Reminder: Given $\mathcal{D}$**(z|A)**, **A** is an anchor if model predictions for perturbed samples z are the <u>same as that of instance x</u> in most cases

- This is equivalent to our notion of **high precision**
  - Recall — "High Precision: By design, users can accurately predict model behavior" ← e.x. explanations are generalizable to other cases (refer back to sentiment analysis case)

$$\text{prec}(A) = \mathbb{E}_{\mathcal{D}(z|A)}\left[\mathbb{1}_{f(x)=f(z)}\right].$$

Searching for an anchor – defining the problem (1/2)

- Calculating prec(A) directly is intractable, we introduce a **probabilistic definition**: anchors satisfy the precision constraint with high probability

$$P\left(\mathrm{prec}(A) \geq \tau\right) \geq 1 - \delta$$

- What if multiple anchors meet this criterion? <u>We prefer anchors with high coverage</u>: the anchor applies to a greater number of samples (more practical)

$$\mathrm{cov}(A) = \mathbb{E}_{\mathcal{D}(z)}[A(z)].$$

Searching for an anchor – optimization problem (2/2)

Therefore, the search for an anchor is the same as the following optimization problem: for all rule A that satisfies the precision constraint, our anchor is the rule that maximizes coverage

$$\max_{A \text{ s.t. } P(\text{prec}(A) \geq \tau) \geq 1-\delta} \text{cov}(A).$$

This is prohibitive!!

Bottom-up (Greedy) Search

1. Start with an empty rule A = {} i.e. one that applies to all instances
2. For each iteration, find the set of all **candidate** rules that extend A by one predicate $\{a_i\}$ (e.x. candidate rules have one more word than current rule)
3. Identify the candidate rule with highest **estimated precision**, replace A with this candidate
4. Terminate when A satisfies the probabilistic precision constraint

Bottom-up (Greedy) Search

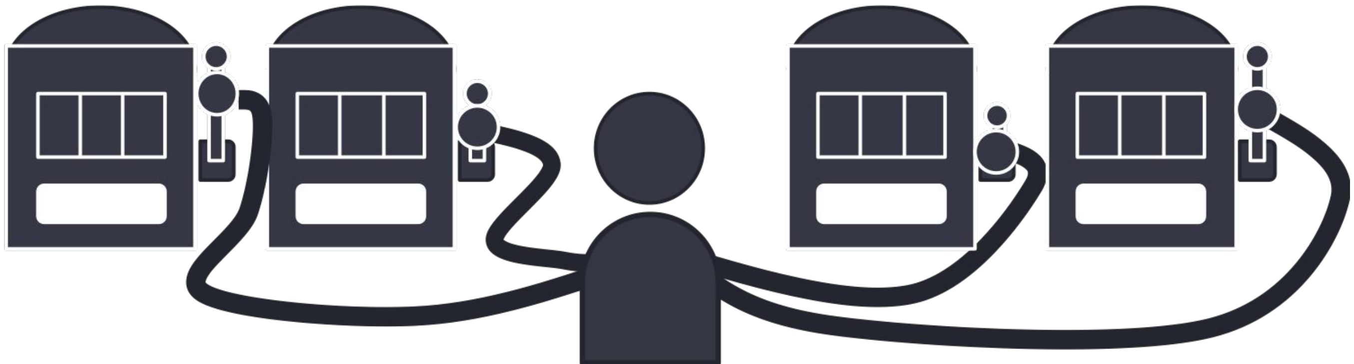
Two things to keep in mind

1. Shorter rules will generally have higher coverage; a bottom-up, greedy approach is **inherently a proxy** for maximizing coverage
2. We estimate precision by drawing samples from Ɗ(•|A), but how do we know how many samples is appropriate?
   a. What is the **minimal** calls to f, or the fewest samples drawn from D, such that we can estimate which candidate rule has the highest true precision ← this is a **multi-armed bandit problem**

Introducing: multi-armed bandit formulation

Wikipedia definition: the multi-armed bandit problem is a problem in which a fixed limited set of resources must be allocated between competing (alternative) choices in a way that maximizes their expected gain ← reinforcement learning

- each candidate rule is an arm, and each pull of the arm is an evaluation of whether f(x) = f(z) ← draw a sample from Ɗ(•|A)
- authors propose using the KL-LUCB algorithm to identify arm with highest precision



Beam Search of Anchors

Greedy approach has two shortcomings:

- Maintain a single rule at a time; suboptimal choice irreversible
- Does not directly consider coverage

Author's Solution: Beam Search ← graph search algorithm

- given set of candidates, identify best B (w.r.t. precision) using KL-LUCB
- generate next set of **candidates from the best B of previous iteration**
- among set of best B, identify best rule A* with **highest coverage**
    - allows us to prune unnecessary candidates

Generalizing instance explanations to model explanations

Like LIME, leverage **submodular pick** (denoted SP-Anchor)

- identify an optimal set of anchors across validation set that best represent global behavior
- selects K anchors that cover as many instances in the validation set as possible (i.e. highest coverage of validation set)

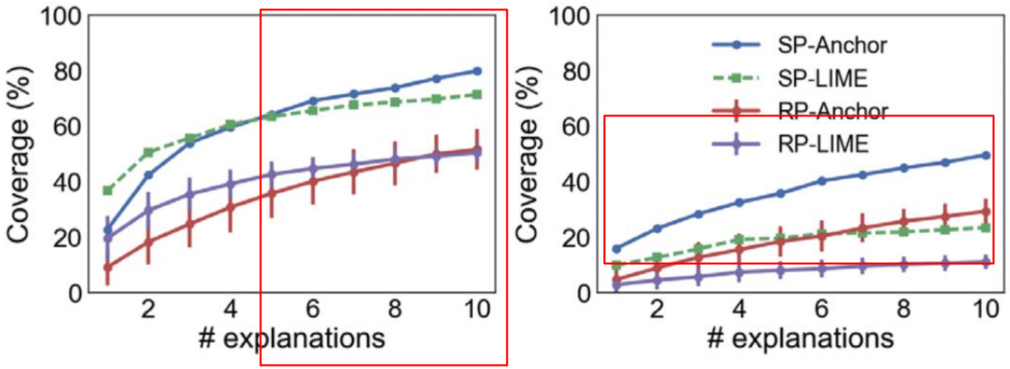
Anchor delivers higher precision on individual instances

- Anchor delivers on high precision, LIME has lower + more inconsistent precision values ← lack of generalizability
- Comparison of coverage between models standardized to have equal precision; inconclusive results generated

		Precision		Coverage	
		anchor	**lime-n**	**anchor**	**lime-t**
adult	logistic	95.6	81.0	10.7	21.6
	gbt	96.2	81.0	9.7	20.2
	nn	95.6	79.6	7.6	17.3
rcdv	logistic	95.8	76.6	6.8	17.3
	gbt	94.8	71.7	4.8	2.6
	nn	93.4	65.7	1.1	1.5
lending	logistic	99.7	80.2	28.6	12.2
	gbt	99.3	79.9	28.4	9.1
	nn	96.7	77.0	16.6	5.4

Coverage of **model explanations** is higher with Anchor

- In real life, users prefer a set of explanations that explain most of model with <u>minimal amount of effort</u> (least # of explanations needed)
- Given the same # of explanations, carefully curated Anchor explanations achieve higher coverage than LIME explanations



(a) *adult* dataset (b) *rcdv* dataset

User studies show Anchor requires less time

- User studies yield similar results as simulated experiments
- We also know users make quicker (and more accurate) decisions with Anchor explanations

Method	**Precision**				**Coverage** (perceived)				**Time/pred** (seconds)			
	adult	rcdv	vqa1	vqa2	adult	rcdv	vqa1	vqa2	adult	rcdv	vqa1	vqa2
No expls	54.8	83.1	61.5	68.4	79.6	63.5	39.8	30.8	29.8 ±14	35.7±26	18.7±20	13.9±20
LIME(1)	68.3	98.1	57.5	76.3	89.2	55.4	71.5	54.2	28.5±10	24.6±6	8.6±3	11.1±8
Anchor(1)	100.0	97.8	93.0	98.9	43.1	24.6	31.9	27.3	13.0±4	14.4±5	5.4±2	3.7±1
LIME(2)	89.9	72.9	-	-	78.5	63.1	-	-	37.8±20	24.4±7	-	-
Anchor(2)	87.4	95.8	-	-	62.3	45.4	-	-	10.5±3	19.2±10	-	-

Limitations?

1. **Overly specific anchors** for predictions near the decision boundary ← lower coverage. LIME may be better here
2. (Potentially but unlikely) **conflicting anchors**: in the wild, multiple anchors with different predictions may apply to the same instance ← unlikely given high precision, submodular pick algorithm
3. Generating **realistic perturbation distributions** that are expressive & interpretable (e.x. image perturbations)