Why Should I Trust You?

Explaining the Predictions of Any Classifier

Casper Vestergaard Kristensen Alexander Munch-Hansen

November 28, 2019

Aarhus University

Outline

1. Meta information
2. Article
   - The LIME framework
   - Explaining Predictions
   - Explaining Models
3. Experiments
   - Simulated User Experiments
   - Human user experiments
4. Conclusion
5. Recap

Meta information

Authors

- Marco Tulio Ribeiro, PhD from University of Washington, Currently a researcher for Microsoft
- Sameer Singh, PhD from University of Massachusetts Amherst, adviser for Marco
- Carlos Guestrin, Professor at University of Washington, adviser for Marco

- This paper won the Audience appreciation award
- These also wrote “Model-Agnostic Interpretability of Machine Learning”
- Marco’s research focus for his PhD was making it easier for humans to understand and interact with machine learning models.

Publishing

- Conference Paper, Research
- KDD ’16 Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining
  - A premier interdisciplinary conference, brings together researchers and practitioners from data science, data mining, knowledge discovery, large-scale data analytics, and big data.
  - Sigkdd has the highest h5 index of any conference involving databases or data in general
  - Highly trusted source

- Main take-away is that this paper was shown at a respected conference

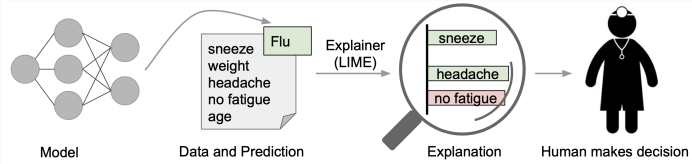
Article

Problem definition

- People often use Machine Learning models for predictions
- Blindly trusting a prediction can lead to poor decision making
- We seek to understand the reasons behind predictions
    - As well as the model doing the predictions

Problem definition

- People often use Machine Learning models for predictions
- Blindly trusting a prediction can lead to poor decision making
- We seek to understand the reasons behind predictions
    - As well as the model doing the predictions

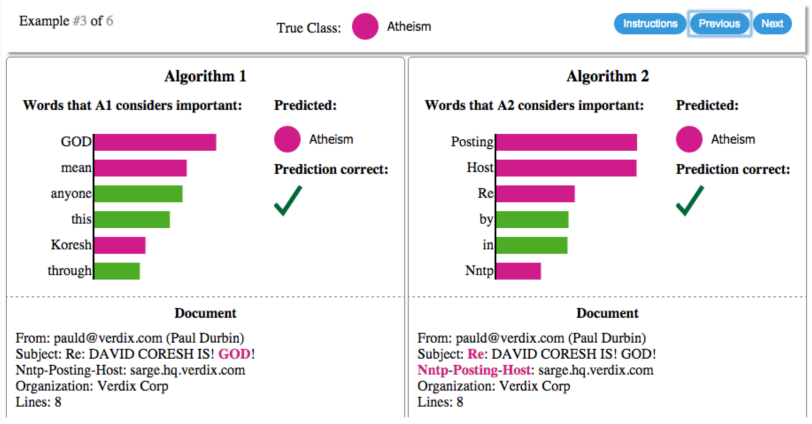


Previous Solutions

- Relying on accuracy based on validation set
- Recognizing the utility of explanations in assessing trust, many have proposed using interpretable models
  - May generalize poorly, if data can't be explained in few dimensions
  - So interpretability, in these cases, comes at the cost of flexibility, accuracy, or efficiency

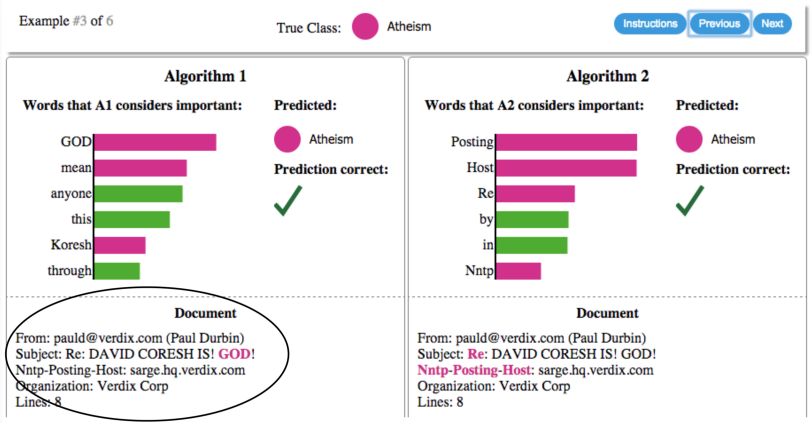
Practitioners consistently overestimate their models accuracy [20], propagate feedback loops [23], or fail to notice data leaks

A look into two predictions



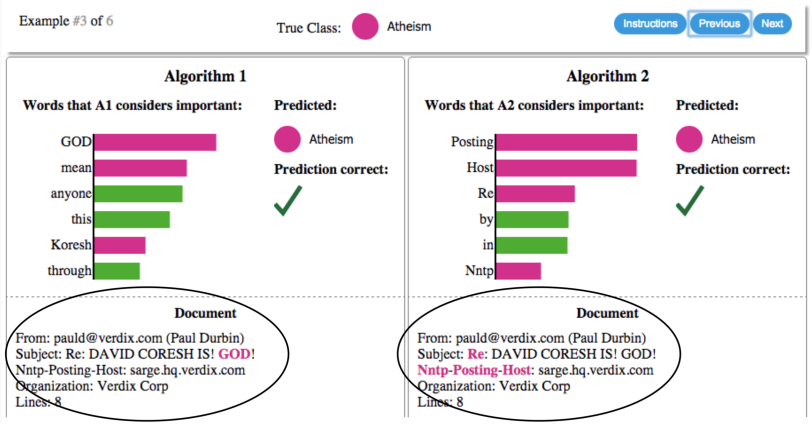
It becomes clear the dataset has issues, as there is a fake correlation between the header information and the class Atheism. It is also clear what the problems are, and the steps that can be taken to fix these issues and train a more trustworthy classifier.

A look into two predictions



It becomes clear the dataset has issues, as there is a fake correlation between the header information and the class Atheism. It is also clear what the problems are, and the steps that can be taken to fix these issues and train a more trustworthy classifier.

A look into two predictions



It becomes clear the dataset has issues, as there is a fake correlation between the header information and the class Atheism. It is also clear what the problems are, and the steps that can be taken to fix these issues and train a more trustworthy classifier.

LIME

- The algorithm created
- Explains the predictions of *any* classifier or regressor in a faithful way, by approximating it locally with an *interpretable* model.

Everything you want

Properties of a good explanation

- It should be *intepretable*
  - They must provide qualitative understanding between the input variables and the response
  - They must take into account the users limitations
- It should have *fidelity*
  - Essentially means the model should be faithful.
- It should be *model-agnostic*
  - Should treat model as a black box

Interpretable

Use a representation understandable to humans

Could be a binary vector indicating presence or absence of a word

Could be a binary vector indicating presence of absence of super-pixels in an image

Fidelity

Essentially means the model should be faithful.

Local fidelity does not imply global fidelity

The explanation should aim to correspond to how the model behaves in the vicinity of the instance being predicted

Model-agnostic

The explanation should be blind to what model is underneath

The Fidelity-Interpretability Trade-off

We want a simple explanation, still capable of displaying fidelity

- Let an explanation be defined as a model $g \in \{0,1\}^{d'} \in G$, where G is a class of *potentially interpretable* models
- $\Omega(g)$ explains the *complexity* of an explanation g
- The model we try to explain is $f : \mathbb{R}^d \to \mathbb{R}$
- $\pi_x(z)$ is a proximity measure between instance z and x and defines the locality around x
- $\mathcal{L}(f, g, \pi_x)$ defines how *unfaithful* g is in approximating f in the locality around π_x.
- To ensure both *interpretability* and *local fidelity*, we minimize $\mathcal{L}$ while having $\Omega(g)$ be low as well

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

Intepretable models could be:

Linear models, decision trees

g is a vector showing presence or absence of *interpretable components*

$\Omega(g)$ could be height of a DT or number of non-zero weights of linear model

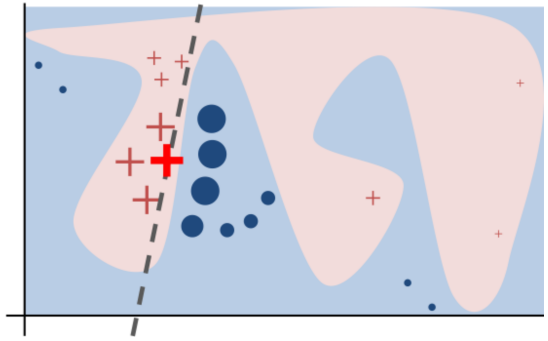
In classification, $f(x)$ is the probability or binary indicator that x belongs to a certain class

So a more complex g will achieve a more faithful interpretation (a lower L), but will increase the value of Omega(g)

Sampling for Local Exploration

Goal: Minimizing $\mathcal{L}(f, g, \pi_x)$ without making assumptions on f

- For a sample x', we need to draw samples around x'
- Accomplished by drawing non-zero elements of x, resulting in perturbed samples z'
- Given $z' \in \{0, 1\}^{d'}$, we compute un-pertubed $z \in R^d$, $f(z)$, so we have a label for z'.



WTF is x' here? - An interpretable version of x
g acts in d' while f acts in d, so when we say that we have z' in dimension d', it's the model g, we can recover the z in the original representation i.e. explained by f in dimension d.

Specifics for linear models

- They focus only on linear explanations
- $G =$ Class of linear models: $g(z') = w_g \cdot z'$
- $L =$ The locally weighted square loss
- $\pi_x(z) = \exp(-D(x,z)^2/\sigma^2)$
  - An exponential kernel function based on some distance function D (could be L2 distance for images)
- Thus; $L(f, g, \pi_x) = \sum_{z,z' \in \mathcal{Z}} \pi_x(z)(f(z) - g(z'))^2$

Explaining an individual prediction

- Solving eq $\text{argmin}_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g)$ is intractable, but this algo approximates it.
- K-Lasso is the procedure of picking K features with Lasso and then using Least Squares to compute weights (features).

Algorithm 1: Sparse Linear Explanations using LIME

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

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

Talk through the algorithm, discussing the sampling and K-Lasso (least absolute shrinkage and selection operator), which is used for feature selection
This algorithm approximates the minimization problem of computing a single individual explanation of a prediction.
K-Lasso is the procedure of learning the weights via least squares. Wtf are these weights??? - The features

Explaining models

Idea: We give a global understanding of the model by explaining a set of individual instances

- Still model agnositc (since the individual explanations are)
- Instances need to be selected in a clever way, as people won't have time to look through all explanations
- Some definitions
  - Time/patience of humans is explained by a budget *B* which denotes number of explanations a human will sit through.
  - Given a set of instances **X**, we define the *pick step* as the task of selecting **B** instances for the user to inspect.

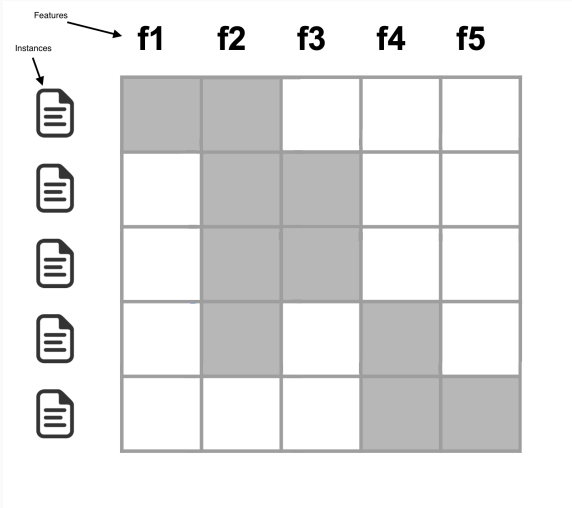
The pick step

The task of selecting B instances for the user to inspect

- Should return the instances which best explains the model
- Looking at raw data is not enough to understand predicitions and get insights
- Should take into account the explanations that accompany each prediction

Should pick a diverse, representative set of explanations to show the user, so non-redundant explanations that represent how the model behaves globally.

Picking instances



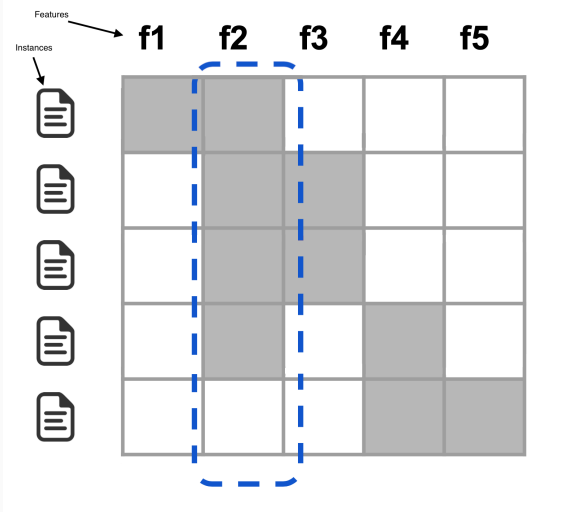
This is a matrix explaining instances and their features explained by a binary list s.t. an instance either has a feature or does not.

The blue line explains the most inherent feature, which is important, as it is found in most of the instances.

The red lines indicate the two samples which are most important in explaining the model.

Thus, explaining importance, is done by: $I_j = \sqrt{\sum_{i=1}^{n} W_{ij}}$

Picking instances



- $I_j = \sqrt{\sum_{i=1}^{n} W_{ij}}$

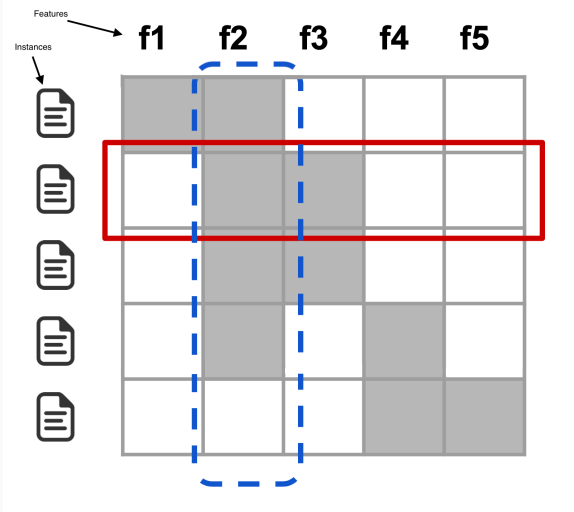
This is a matrix explaining instances and their features explained by a binary list s.t. an instance either has a feature or does not.

The blue line explains the most inherent feature, which is important, as it is found in most of the instances.

The red lines indicate the two samples which are most important in explaining the model.

Thus, explaining importance, is done by: $I_j = \sqrt{\sum_{i=1}^{n} W_{ij}}$

Picking instances



- $I_j = \sqrt{\sum_{i=1}^{n} W_{ij}}$

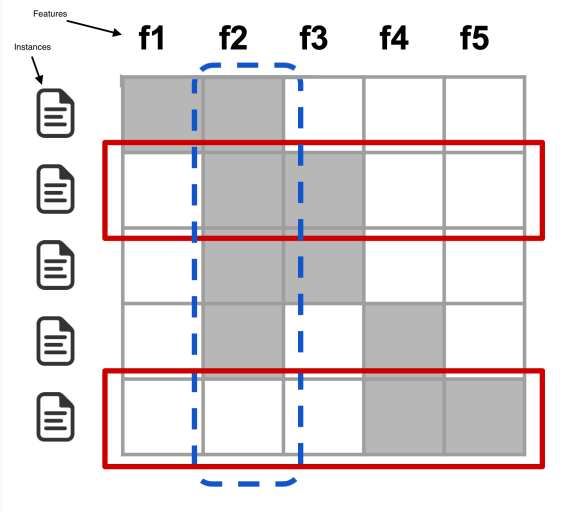
This is a matrix explaining instances and their features explained by a binary list s.t. an instance either has a feature or does not.

The blue line explains the most inherent feature, which is important, as it is found in most of the instances.

The red lines indicate the two samples which are most important in explaining the model.

Thus, explaining importance, is done by: $I_j = \sqrt{\sum_{i=1}^{n} W_{ij}}$

Picking instances



- $I_j = \sqrt{\sum_{i=1}^{n} W_{ij}}$

This is a matrix explaining instances and their features explained by a binary list s.t. an instance either has a feature or does not.

The blue line explains the most inherent feature, which is important, as it is found in most of the instances.

The red lines indicate the two samples which are most important in explaining the model.

Thus, explaining importance, is done by: $I_j = \sqrt{\sum_{i=1}^{n} W_{ij}}$

Submodular Picks

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

Algorithm 2: Submodular pick (SP) algorithm

Require: Instances X, Budget B

forall $x_i \in X$ **do**
  $W_i \leftarrow$ **explain**(x_i, x_i') // Using Algorithm 1
end
for $j \in \{1 \ldots d'$ **do**
  $I_j \leftarrow \sqrt{\sum_{i=1}^{n} |W_{ij}|}$ // Compute feature importances
end
$V \leftarrow \{\}$
while $|V| < B$ **do** // Greedy optimisation of Eq 4
  $V \leftarrow V \cup \text{argmax}_i \, c(V \cup \{i\}, W, i)$
end
return V

Note: maximizing a weighted coverage function is NP-hard, but the version used in the algorithm is iterativily greedy, so it just adds the one with the maximum gain, which offers a constant-factor approximation guarantee of $1 - 1/e$ to the optimum.

Experiments

Experiments

Interested in three questions:

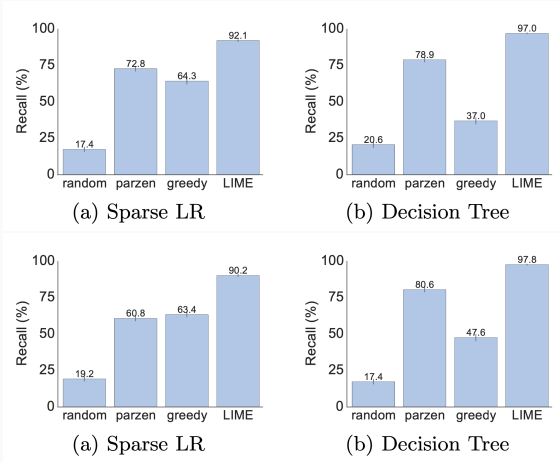
- Are the explanations faithful to the model?
- Can the explanations aid users in ascertaining trust in the individual predictions?
- Are the explanations useful for evaluating the model as a whole?

Faithfulness and golden standard

- Explanations of **LIME** are compared with **parzen** as well as greedy and random algorithms.
  - **parzen** approximates black box classifier globally and explains individual predictions by taking the gradient of the prediction probability function.
- Faithfulness of explanations is measured on classifiers that are interpretable: **Logistic Regression** and **Decision Tree**.
  - Both find 10 features, which are the *gold standard* features
- For each prediction on the test set, explanations are produced and the fraction of the gold features found, is computed.

- Train logistic regression and decision tree classifiers, so that they use a maximum of 10 features to classify each instance.
- These 10 features are the gold set of features that are actually considered important by the model.
- The explanations should recover these features.

Faithfulness



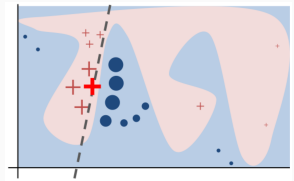
- We observe that the greedy approach is comparable to parzen on logistic regression, but is significantly worse on decision trees, since changing a single feature at a time often does not have an effect on the prediction.
- The overall recall by parzen is low, likely due to the difficulty in approximating the original highdimensional classifier.
- LIME consistently provides > 90% recall for both classifiers on both datasets, demonstrating that LIME explanations are faithful to the models.

Should I trust this prediction?

- Randomly select 25% of the features as untrustworthy.
- Simulated users deem a prediction untrustworthy if:
  - Lime & Parzen: the linear approximation changes, when all untrustworthy features are removed from the explanation.
  - Greedy & Random: they contain any untrustworthy features.

Table 1: Average F1 of *trustworthiness* for different explainers on a collection of classifiers and datasets.

	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**



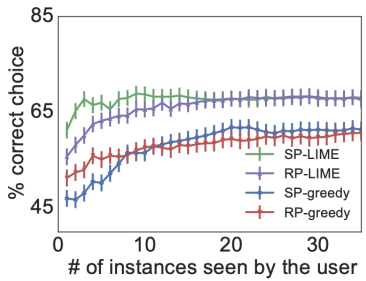
- 2nd experiment: test trust in individual predicitions.
- Test-set predictions are deemed (oracle,truely) untrustworthy if the prediction from the black-box classifier changes when these features are removed.
- Simulated user knows which features to discount.
- If the line is different when untrustworthy features are removed, something is wrong!
- F-measure = a measure of a test's accuracy, i.e. if the user correctly distrusts a prediction based on the explanation given by fx LIME.
- The table show that the other methods achieve lower recall = mistrust too many predictions, or lower precision = trust too many predictions.

Can I trust this model?

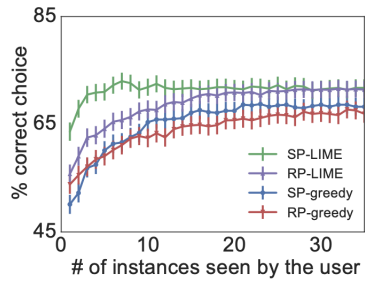
- Evaluate if explanations can be used for model selection
- They add 10 artificially “noisy” features s.t.
  - Each artificial feature appears in 10% of the examples in one class, and 20% of the other in the training/validation data.
  - While on the test instances, each artificial feature appears in 10% of the examples in each class.
- Results in models both using actual informative features, but also ones creating random correlations.
- Pairs of competing classifiers are computed by repeatedly training pairs of random forests with 30 trees until their validation accuracy is within 0.1% of each other, but their test accuracy differs by at least 5%.

- 3rd experiment: two models, user should select the best based on validation accuracy.
-

Can I trust this model?



(a) Books dataset



(b) DVDs dataset

- They evaluate whether the explanations can be used for model selection, simulating the case where a human has to decide between two competing models with similar accuracy on validation data.
- Accomplished by "marking" the artificial features found within the B instances seen, as unstrustworthy. We then evaluate how many total predictions in the validation set should be trusted (as in the previous section, treating only marked features as untrustworthy).
- As B, the number of explanations seen, increases, the simulated human is better at selecting the best model.

Can humans pick the best classifier?

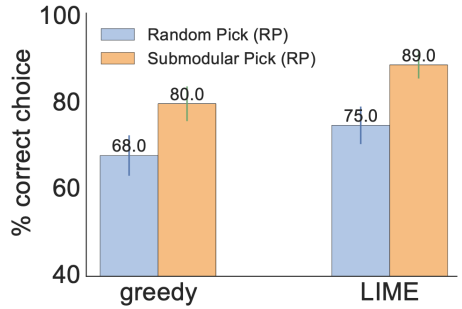
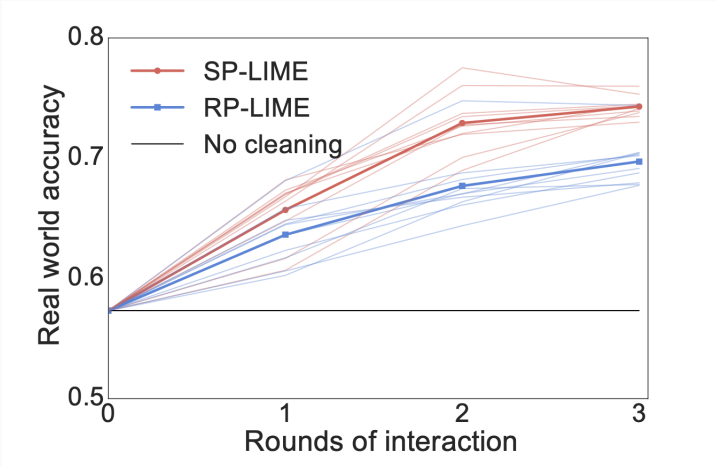


Figure 9: Average accuracy of human subject (with standard errors) in choosing between two classifiers.

- Non-expert humans, without any knowledge of machine learning
- Train two classifiers, one on standard data set and one on a cleaned version of the same data set
- Use the newsgroup dataset for training, which is the one with the atheism/christianity emails
- Run the classifiers on a “religion” dataset, that the authors create themselves, to question if the classifiers generalizes well
- Standard one achieves higher validation accuracy - but it’s not correct!
- Humans are asked to pick the best classifier when seeing explanations from the two classifiers for B and K = 6 (They see 6 explanations with 6 features)
- Repeated 100 times
- Clearly SP LIME outperforms other options

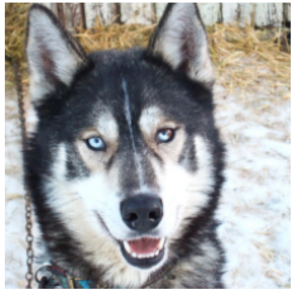
Can non-experts improve a classifier?



- 200 words were removed with SP, 157 with RP
- Out of the 200 words removed, 174 were selected by at least half the users, 68 by all

- Non-expert humans, without any knowledge of machine learning
- Use newsgroup dataset
- Ask mechanical turk users to select features to be removed (email headers), before the classifier is retrained
- B = K = 10
- Accuracy shown in graph, is on the homebrewed religion dataset
- Without cleaning, the classifiers achieve roughly 58%, so it helps a lot!
- It only took on average 11 minutes to remove all the words over all 3 iterations, so little time investment, but much better accuracy
- SP-LIME outperforms RP-LIME, suggesting that selection of the instances to show the users is crucial for efficient feature engineering.

Can we learn something from the explanations?



(a) Husky classified as wolf

(b) Explanation

- Images picked to create fake correlation between wolf and snow
- Use Logistic Regression classifier
- Features come from Google's pre-trained *Inception* **NN**

- Use graduate students who has taken at least one course in machine learning.
- Intentionally train bad classifier by having snow on all wolf-images during training.

Can we learn something from the explanations?

- Present 10 predictions without explanations
  - 2 are miss-predictions with a husky in snow and a wolf without snow, the rest are correct
- Ask three questions:
  1. Do you trust this algorithm to generalize?
  2. Why?
  3. How do you think the algorithm distinguishes?
- Results shown in table, before and after having seen the explanations.

	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

- Clearly shows that seeing the explanations leads to insight, changing their answers consistently.

Conclusion

Conclusion

- They argue that trust is crucial for effective human interaction with machine learning systems
- Explaining individual predictions is important in assessing trust
- They proposed LIME, a modular and extensible approach to faithfully explain the predictions of any model in an interpretable manner
- They introduced SP-LIME, a method to select representative and non-redundant predictions, providing a global view of the model to users.
- Experiments demonstrated that explanations are useful for a variety of models in trust-related tasks in the text and image domains

- Establishing trust in machine learning models, requires that the system can explain its behaviour.
  - Both Individual predictions.
  - As well as the entire model.
- To this end, they introduce (submodular-pick) SP-LIME, which select a small number of explanations, which together (hopefully) explain the entire model.
- Experiments show that this is indeed the case.

Future work

- Explanation families beyond spare linear models.
- One issue that they do not mention in this work was how to perform the pick step for images.
- They would like to investigate potential uses in speech, video, and medical domains, as well as recommendation systems.
- They would like to explore theoretical properties (such as the appropriate number of samples) and computational optimizations (such as using parallelization and GPU processing)

- The paper only describes sparse linear models as explanations, but the framework supports other explanation families, such as decision trees.
- They envision adapting the explanation family based on the dataset and classifier.
- Extend framework to support images(better), speech, video, etc.
- LIME framework ready for production and available on GitHub.
- Therefore would like to optimise computation using parallelisation and GPU processing.

Recap

Recap

- LIME is a framework for explaining predictions made by machine learning algorithms.
- It explains models by intelligently picking a limited number of individual explanations.
- Only uses linear models at the moment.
- Is shown to make it significantly easier for people to better the classifiers, even non-experts.

- LIME is able to explain entire ML models by presenting the user with a limited number of individual, non-redundant explanations, that describe the model well enough without overwhelming them.

Discussion

- Is it fair that the authors create their data in such a way that *Parzen* becomes unusable in their tests?
- What do you expect to happen if the data is very non-linear even in the local predicitions?
- The *K-Lasso* algorithm used in *Algorithm 1* is explicitly used for regression analysis and as such it should only work when they use linear models for their explanations. Is this okay?