Classifier Explanation

Introduction to the Algorithms LIME and SP-LIME

Leonhard Applis

TH Nürnberg

21.1.2019

Table of Contents

Me: Hey Siri, order me a Pizza

Siri: *(After a short break that nearly drains your whole battery)* Ok, I'm calling your mother...

Me: Wait! Why would you do this!?

Siri: This is the 5th time you ordered Pizza this week.

What do we want from our model?

1. Why did failed predictions fail?
2. Why did correct predictions succeed?
3. Why is my model uncertain about a prediction?

special importance:
setting a model *live*, where it's not *prelabeled*

Trusting a Prediction
Requirements

Interpretations must be ...

- *human-readable*
- reproducable (same input + same model $\rightarrow$ same output)
- **model agnostic**, meaning they can work with any (black-box) model

Difficulties:

- Models can be huge (millions of weights)
- Inputvectors can be huge (e.g. images)
- Some models are to complex by it's structure to be readable, (e.g. neural networks)

Example
Desired output of a "Atheism"-Classifier

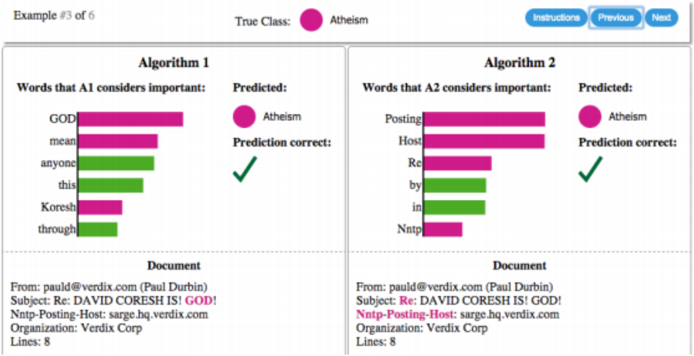

Figure: LIME-Text: predicting "Atheism" for given text

Both algorithms predict correct - yet Algorithm 2 has strange reasons.

Trusting a Model

trusting predictions $\neq$ trusting a model

What do we want?

1. get an *overview* of our Model
2. compare models in reasonable time
3. proove correctness & flaws of a model
4. improve our models

Prooving a Model

Several topics which benefit from machine learning, but need special care:

1. Terrorism-detection
2. Medical diagnosis & prescriptions
3. Fraud-detection

Noone will buy a model, if you can't prove that it's performing reasonable predictions.

Improving a Model

There are several issues, at which explanations can help you improve your models:

1. Filtering of Features
2. Find overfitted weighting of features
3. Find Links in Classification (Similiar Classes and Features)

Gaining insights from explanations can help you improve your model!

Table of Contents

What do we want:

- Human Readable Model Explanation
- For Every Classifier
- For Every Input

features $\neq$ human readable

To gain *readability*:

- show influence relative to each other, not as numbers
- only show most important features
- use *superpixels* instead of pixels

Definitions

Let:

1. G be any possible explanation model
2. g be our explanation Model
3. $\Omega(g)$ the complexity of our Model
   - Weights in a regressions model
   - Depth of an decisiontree
   - Number of trees in a random forest
4. $f : Features- > Class$ be the real classification
5. $\Pi_x(z)$ as proximity-measure from x to z
6. $\mathcal{L}(f, g, \Pi_x)$ measure of un-faithfullness of g compared to f given the proxmity Π_x

Wanted:

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

Read:

- We want for every input x
- an explanation(-model)
- where complexity of g and the failure of g are minimal
- given a set of possible explanations G

We do so by picking samples x' as subsets from an input x and **optimizing** our model g [1]

[1] We do not really check different models, we train one

Additional Requirements:
LASSO[2] - *Least Absolute Shrinkage and Selection Operator*
Machine Learning algorithm to select most important features relative to each other.
G are only *sparse linear regression models* (e.g. Decision Trees or simple logistic regression)

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

[2] Further Reading: https://beta.vu.nl/nl/Images/werkstuk-fonti_tcm235-836234.pdf

Table of Contents

Setup Problem, Show Code, Plot Examples, nice This could be left out from the presentation, and just be a live demo
Do both: LIME and ANCHOR and sample with SPLIME

Table of Contents

Problem with Sampling

Explain that we have to little time to inspect everything Looking for a new way to pick samples

Here is the Pseudocode

SPLIME Example

I guess this needs more than 2 Pages, we should add an example