Ruprecht-Karls-Universität Heidelberg

Explainable Machine Learning

Dr. Ullrich Koethe

Summer Term 2018

Interpreting Machine Learning Models via Local Linear Approximations

Seminar Report

by

Philip-William Grassal (3535063)

Contents

1 Introduction

Machine learning algorithms have received attention from a wide range of research fields and businesses that deploy learned models in a huge variety of contexts. For example, text and sentiment analysis in social media [Nak+16] or medical image classification for disease detection and decision support [CLW17], [Zha+15]. However, assigning the responsibility for such tasks to models is only meaningful if they can be trusted. The most common approaches to guarantee this include a variety of summary statistics, e. g., accuracy and precision, calculated on validation and test data sets to evaluate the reliability of a model along the training process.

Contrarily, by training an image classifier to separate images of huskies and wolves, Ribeiro et al. [RSG16] demonstrate that summary statistics are not appropriate indicators for faithfulness. In their training process, the training, validation and test sets only contain images of huskies with snowy backgrounds leading to a classifier that distinguishes wolves and huskies based on white background pixels. Yet, it achieves perfect summary statistics during the validation making it an allegedly trustworthy model. Considering this in a real scenario -happening by accident- exemplifies that accuracy scores do not sufficiently indicate fidelity.

From an end-user's perspective, a machine learning model embraces two notions of trust. First, trusting a single prediction of a model, for example, when using it in the context of decision making. Second, trusting the model as it is and all of its predictions which is important when one decides for deploying a model. As solution, [RSG16] proposes the LIME and SP-LIME methods to either evaluate the trustworthiness locally w. r. t. to a model's prediction or globally w. r. t. to the model itself. These approaches complement common summary statistics to validate a model. This report aims to elucidate the foundation behind LIME/SP-LIME and to reflect on the experimental studies in [RSG16] where both methods are compared to other recent approaches from literature.

The report continues in three chapters. First, Chapter 2 provides formal explanations of LIME and SP-LIME. In addition, we present a custom example implemented with the LIME framework. Second, Chapter 3 reviews user experiments with LIME conducted by the authors. At last, Chapter 4 summarizes this report and provides an outlook.

2 Theoretical Foundations

Evaluating the fidelity of a machine learning model is a purely human-based decision. Given the explanation of a model's prediction, a user can decide whether the model acts in a reasonable, intelligent manner which finally leads to trust in the model. Herein, an explanation is represented by visual or textual artifacts that give a comprehensible illustration of how the model arrives at a prediction given certain inputs. In [RSG16], the authors formulate desired characteristics for such explanation methods which they focused on when designing LIME/SP-LIME. So, we summarize them in the following.

First, an explanation must be easily **interpretable** taking into account user's limitations. Thus, even non-experts should be able to understand how inputs lead to a prediction. For example, a small set of features assigned with weights, that indicate their importance for a certain prediction, is considered more interpretable than gradient vectors [Bae+10] or a confusingly long list of features used in a simplified linear model. Second, we expect an explainer to be **model-agnostic**, i. e. to work with any model. Third, an explainer is supposed to be **locally faithful**, i. e. it must precisely illustrate the model's behavior in the vicinity of a single prediction. Hence, an explanation should describe a correct local approximation of the model being analyzed. Last, beyond explaining predictions, an explainer should provide a **global perspective** which enables users to evaluate their trust not only towards single predictions but also towards the complete model. This is done by selecting few local explanations that are representatives of the model. If they are trusted, the model can be trusted, as well.

In Section 2.1 and Section 2.2, we explain how these characteristics are realized in LIME and SP-LIME, respectively.

2.1 LIME

For locally faithful explanations, i. e. predictions with respect to a single instance and other instances in a close proximity, [RSG16] propose *Local Interpretable Model-agnostic Explanations (LIME)*. The core idea is to approximate the behavior of a complex model f with regard to one instance x by an interpretable model $g \in G$. We

let G be the set of possible interpretable models, such as linear models or decision trees, that can immediately be transformed into visual or textual artifacts, making them good explainers. In addition, g uses an interpretable representation $x' \in \{0,1\}^{D'}$ instead of x as input. The reason for this is to avoid inherent complexity that usually comes with original features, such as multiple color channels for each pixel in image classification. Typically, x' is supposed to give a comprehensible binary representation of properties that instance x has or not, e. g., is there a head in the image or a certain word in the text. Nonetheless, this step is not required if x is already interpretable.

In LIME, the model being explained is defined as $f : \mathbb{R}^D \to \mathbb{R}$. For classification, this is a binary classifier with $f(x)$ being the posterior probability of one class. If the classifier predicts multiple classes, it needs to be interpreted as binary classifier for each class of interest. Similarly, we define $g : \{0,1\}^{D'} \to \mathbb{R}$ having the same posterior output space. Further, let $\pi_x(z)$ denote a locality measure between x and another instance z. Using this definition, we can express the approximation loss with $\mathcal{L}(f, g, \pi_x)$ that measures how well g approximates f in the proximity of x. To enforce this local approximation via the loss function, [RSG16] suggests creating a synthetic local data set $\mathcal{Z}'$ by sampling around the interpretable features x'. This is done by randomly flipping each binary feature in x' for a desired number of times and adding the result to $\mathcal{Z}'$. As we expect the relation to be bidirectional between original and interpretable features, a corresponding z can be found for each $z' \in \mathcal{Z}'$. If g behaves very similar on all z' as f does with all z, then g is locally faithful, resulting in a small value for $\mathcal{L}$. The discrepancy between their behaviors is expressed as loss $\mathcal{L}$.

We derive the final objective function in LIME

$$\hat{g} = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{2.1}$$

with $\Omega(\cdot)$ being an arbitrary complexity measure that enforces interpretability in the objective function. For example, $\Omega(\cdot)$ could describe the depth of a decision tree or the number of non-zero weights in a linear model. With this objective, we obtain a model $\hat{g}$ that is both interpretable and locally faithful with respect to x. Additionally, no assumptions about f have been made. Hence, LIME is model-agnostic.

Sparse Linear Models We demonstrate LIME by taking the example of an image classification task and sparse linear models as our class G of explainers. Consider a convolutional neural network (CNN) f and an image x which gets classified correctly

by f. Yet, explaining how f arrived at the decision *dog* is a complex process involving convolutional filter masks, network weights, etc. However, if we use a sparse linear model $g(x) = w \cdot x$ to approximate f at x, a non-zero weight will be assigned to a few features x_j indicating their importance for the decision. Thus, this gives a simple explanation of f via g. Furthermore, the original x consists of multiple color channels and many pixels which is considered rather complex than interpretable. Hence, we replace them by interpretable representations x' where each x'_j is a binary value indicating if a specific super-pixel (contiguous pixel patch) is contained in the image. In the end, the complex prediction of the CNN $f(x)$ is expressed by the behavior of $g(x')$ and its weight vector.

First, we generate a local data set $\mathcal{Z}'$ based on x' by randomly flipping the bits in x'. Due to this random selection of super-pixels contained in x', new instances z' are generated which correspond to perforated images z where all super-pixels with 0 in z' are removed in z. The locality of z regarding x is measured by $\pi_x(z) = \exp(-D(x,z)^2/\sigma^2)$ where $D(\cdot,\cdot)$ is a distance function for images, such as the L_2-distance. Given this, we define the approximation loss over all pairs of z, z'

$$\mathcal{L}(f, g, \pi_x) = \sum_{z,z'} \pi_x(z) * (f(z) - g(z'))^2 \tag{2.2}$$

As we desire to minimize $\mathcal{L}$, we obtain a weighted least squares problem which has an analytical solution and efficient solvers, e. g., Cholesky, singular value or QR decomposition. Yet, the objective function of LIME contains an additive complexity term that, in this example, is defined according to [RSG16]

$$\Omega(g) = \infty * \mathbb{I}[\#\text{non-zero components in } w > K] \tag{2.3}$$

$\Omega(g)$ becomes infinity for all interpretable models that have more than K non-zero weights and 0 otherwise. Hence, by choosing this parameter properly, one gets the top K most important pixel patches for the decision of f.

Both functions are inserted into the objective function, Equation 2.1. To derive the optimal solution, [RSG16] utilize lasso regularization for feature selection until the sparsity is high enough, i. e. #non-zero components in $w \leq K$ and thus $\Omega(g) = 0$. The remaining weighted least squares problem is solved with the selected subset of features and all local instances in $\mathcal{Z}'$. The result is a sparse linear model $\hat{g}$ which approximates the decision f at x. The non-zero weights of $\hat{g}$ representing the importance of each super-pixel for this decision.

We demonstrate the effectiveness of their approach by explaining the decision of Google's Inception neural network using the image in Figure 1. Note the Inception

(a) Original Image (b) Bell Pepper (c) Orange (d) Lemon

Figure 1: Explanations of top 3 classes predicted for original image a). First, bell pepper with posterior $p = 0.49$ in b). Second, orange ($p = 0.06$) in c) and, third, lemon($p = 0.02$) in d)

net has multiple output classes. In consequence, an explanation for each class must be given separately. In Figure 1, the explanations of the classes *bell pepper*, *orange* and *lemon* are depicted. In our experiment, K is set to 10, i. e. in each explaining image the 10 most influential super-pixels for the respective decision are displayed. If the depicted super-pixels were not what we expected, for example, pixels of the bell peppers were highly important when classifying oranges, we could easily conclude that the model's prediction is untrustworthy. We implemented this experiment using the LIME framework [Lim]

2.2 SP-LIME

With LIME, single predictions of a complex model f are transformed into simple explanations. If the fidelity of a complete model f is of interest, however, a global explanation of f is required which can not be provided by a single explanation with LIME. The solution is to form a representative set of explanations that give a global perspective on the model. Therefore, one has to pick the most representative local approximations $\hat{g}_i$ of f with regard to instances x_i. If all of these are trustworthy, the model can be trusted, as well. In contrast, providing a close global approximation rather than a global perspective is difficult as it requires to map the complex model to a simple, interpretable one which is equally powerful.

Naturally, the larger the set of local explanations is the more precise the global perspective is. Considering that checking every decision for its fidelity is not automatable, it can not be infinitely large. Let B denote the number of explanations a human is willing to evaluate in order to accredit fidelity and X be the set of instances x_i with corresponding explanations $\hat{g}_i$. The goal is to select B instances such that their explanations provide the most faithful representation of the model's behavior. Ribeiro et al. [RSG16] propose an extension of LIME for this purpose

where they define a pick strategy, thus calling it **S**ubmodular **P**ick-LIME. Continuing the example of sparse linear models, we elucidate how this selection strategy works. First, the weights of each $\hat{g}_i$ are grouped in an $N \times D'$ matrix W with $N =\mid X \mid$ and D' being the dimension of interpretable feature vector x'_i. For each element $W_{ij} =\mid w_{\hat{g}_{ij}} \mid$. Consequently, a column j represents the weights for a specific interpretable feature among different explanations. Let $\mathcal{I}_j$ denote the importance score for column j which measures how often the feature is considered important across different explanations. For instance in text classification, the authors propose $\mathcal{I}_j = \sqrt{\sum_{i=1}^{N} W_{ij}}$. Thus, if $\mathcal{I}_1 > \mathcal{I}_2$, the interpretable feature in column 1 is more influential for decisions of model f than the feature in column 2. The goal is to select B rows that have high coverage of different interpretable features and columns, respectively. This means that the B corresponding explainers use different sets of features for their explanations, thus approximating the f at different positions. We denote the subset of rows (explainers) select from W as V. Formally, the coverage being achieved with V is defined as

$$c(V, W, \mathcal{I}) = \sum_{j=1}^{D'} \mathbb{I}[\exists i \in V : W_{ij} > 0] * \mathcal{I}_j \tag{2.4}$$

If the weight vectors of each $\hat{g}_i$ are not sparse, the indicator term $\mathbb{I}[\exists i \in V : W_{ij} > \tau]$ may be changed to use some threshold τ instead of 0. Our initial selection problem is expressed as

$$\hat{V} = \underset{V, |V| \leq B}{\arg\max}\, c(V, W, \mathcal{I}) \tag{2.5}$$

The objective is a weighted coverage problem proven NP-hard in [Fei98]. For this reason, [RSG16] present an iterative, greedy algorithm which chooses the row with maximum coverage gain in each iteration and adds it to V. This strategy is always close to the true optimum by a constant factor, see [KG14]. The complete calculation scheme of SP-LIME is presented in Algorithm 1.

Algorithm 1 Submodular Pick (SP) see [RSG16]

Input: Instances X, Budget B
for $x_i \in X$ **do**
  $W_i \leftarrow$ **LIME**(x_i)
end for
for $j \in \{1, ..., D'\}$ **do**
  $\mathcal{I}_j \leftarrow \sqrt{\sum_{i=1}^{N} W_{ij}}$
end for
$V \leftarrow \{\}$
while do $\mid V \mid < B$
  $\hat{i} \leftarrow \arg\max_i c(V \cup \{i\}, W, \mathcal{I})$
  $V \leftarrow V \cup \{\hat{i}\}$
end while
return V

3 Evaluating LIME

Complementing their contribution of LIME and SP-LIME, [RSG16] conduct empirical studies with simulated and real human subjects demonstrating the effectiveness of their approach. In Section 3.1, we reflect on these studies and highlight major points. Afterwards, Section 3.2 discusses advantages and open issues encountered with LIME.

3.1 Empirical Studies

To evaluate LIME, other explanation approaches from literature (*parzen* by [Bae+10], *greedy*[1] similar to [MP14]) and a *random* explanation approach are compared. For SP-LIME, each of the previous approaches is extended to give a global perspective via submodular picks. To put the focus on evaluating the effectiveness of the pick strategy, a random pick (RP) strategy is used as counter part. As the name indicates, it selects local explanations randomly. Eventually, SP-LIME is compared with SP-greedy, SP-parzen, SP-random, RP-LIME, RP-parzen, RP-greedy and RP-random. All prediction tasks in their study are based on text data sets and text classification.

One part of the studies comprises simulated user experiments. The first experiment evaluates how faithful each approach is when highlighting the K most important features to explain a model's decision. Therefore, a sparse linear regression model and a decision tree are trained artificially such that they utilize only a predefined *gold* set of features for their decisions. Afterwards, LIME, parzen, the greedy and random approach are used to explain a decision. The recall of gold features in their explanation is used to compare all explainers, see Figure 2. It illustrates that more truly important (gold) features of the sparse linear model f are identified via LIME than by all other approaches.

In a second experiment, the most generalized classifier should be identified based on picked explanations. The set of classifiers to choose from contains random forest models with different generalization (train-test gap). For each method and classifier, B explanations are generated being automatically checked for their trustworthiness

[1] We refer to [RSG16] for more information.

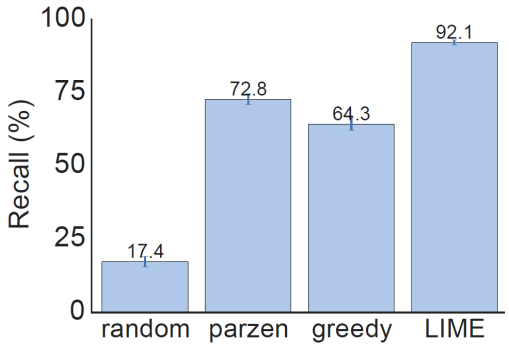

Figure 2: Gold features rediscovered by an explanation

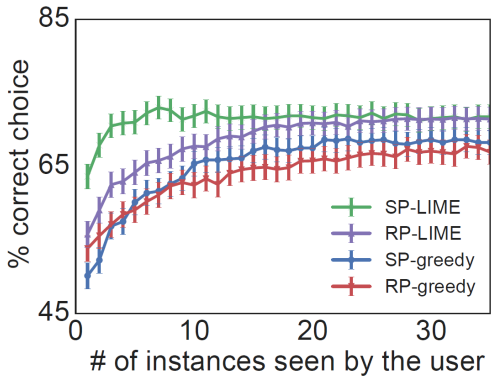

Figure 3: Precision when deciding for the best model

based on predefined criteria. For each explanation approach, the model with the highest number of trustworthy explanations is selected. In Figure 3, we see how often an approach led to a correct choice - with regard to different values of B. Again, SP-LIME outperforms other methods. In addition, the graphs show that the submodular pick strategy is effective, especially for small $B < 20$. For larger B, also the random pick strategy works satisfyingly. The results of parzen and random are completely off. So, they are not displayed at all.

The other part of the studies targets real human interaction involving 100 participants on Amazon Mechanical Turk. In one experiment, machine learning non-experts are asked to identify the better model out of 2 just by seeing $B = 6$ explanations of the model's decisions. The results in Figure 4 clearly show that with SP-LIME 90% of the participants selected the true optimal model which is more than any of the other explanation methods achieved.

The second experiment focuses on feature engineering via SP-LIME and RP-LIME. Non-experts are iteratively asked to pick features which are marked as important by multiple explanations but are semantically incorrectly used by model f. In the next iteration f is trained without the incorrectly used features and gets explained again. Already after three iterations, participants are able to improve f by 20% test accuracy on average, see Figure 5. As expected, RP-LIME performs slightly worse than SP-LIME.

3.2 Discussion

As demonstrated in the previous chapters, LIME and SP-LIME fulfill all desired characteristics of explainers: interpretable, model agnostic, locally trustworthy and

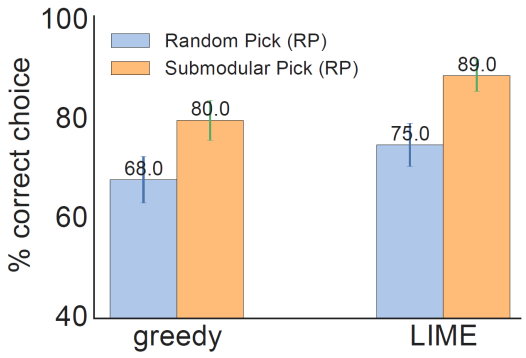

Figure 4: Average precision of participants that have to choose the best of 2 models

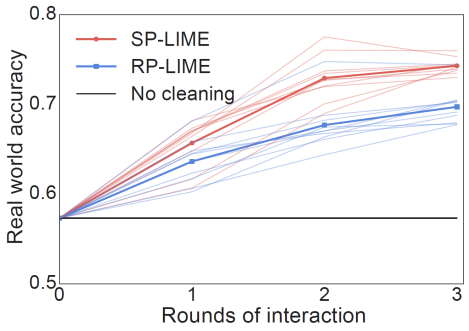

Figure 5: Improvements in test accuracy achieved by ML laymen that used LIME for feature engineering

globally representative. Moreover, the experiments show that this approach outperforms other black box methods from literature and it helps even layman to understand and improve a model.

However, we discover difficulties when using SP-LIME in the context of image classification which has not been identified by the experiments as they focus entirely on text analysis. The main issue concerning images is the mapping from multi-channel pixels to interpretable features, i.e. super-pixels. For a single image, this can easily be achieved by combining matching neighboring pixels as illustrated in Section 2.1. Yet, for SP-LIME multiple instances with the same set of interpretable features are required. Due to the tremendously high diversity of images, it is very unlikely to find another image with the same super-pixels. An intuitive solution is to handcraft semantic information of an image and represent them as interpretable features. Consider, for example, two completely different images x_1, x_2 each depicting a dog at different positions. To explain the classification for each image, LIME expects us to define super-pixel which, of course, are very different for x_1 and x_2. Hence, we cannot simply create a matrix W with SP-LIME. To circumvent this, we suggest representing each super-pixel by its semantic meaning, i. e. by the content it displays. For instance, both, x_1, x_2, contain dog heads. Even though the respective super-pixels of dog heads might be at different position and do not look the same, they both represent the same semantic information. Such a uniform representation among different instances enables using SP-LIME.

Nonetheless, this requires a significant manual effort as each super-pixel and its meaning are manually determined.

4 Conclusion

In this report, we reviewed the theory of LIME and SP-LIME by giving examples of sparse linear models. Also, empirical studies evaluating LIME in practice were explained and discussed. In conclusion, the paper [RSG16] presents a straightforward black-box approach to interpret any machine learning model. It stands out due to its simplicity and effectiveness compared to other approaches from literature in 2016.

For future work, the authors aim to evaluate decision trees as another family of interpretable models. In their paper, they solely focus on linear explainers which is why they motivate a comparison among different families of interpretable approximations. Furthermore, they would like design a pick strategy for images that addresses the issues discussed in Section 3.2. At last, they desire to explore different practical fields that could possibly benefit from explanation by LIME.

Bibliography

[Bae+10] David Baehrens et al. “How to Explain Individual Classification Decisions”. In: *Journal of Machine Learning Research* 11 (2010).

[CLW17] Y. Cho, S. Lee, and S. Woo. “The Kirsch-Laplacian edge detection algorithm for predicting iris-based desease”. In: *2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design*. CSCWD ’17. 2017, pp. 551–555.

[Fei98] Uriel Feige. “A Threshold of Ln N for Approximating Set Cover”. In: *J. ACM* 45.4 (1998), pp. 634–652.

[KG14] Andreas Krause and Daniel Golovin. “Submodular Function Maximization”. In: *Tractability: Practical Approaches to Hard Problems*. Cambridge University Press, 2014, 71–104.

[Lim] *LIME*. 2018. URL: `https://github.com/marcotcr/lime` (visited on 07/02/2018).

[MP14] David Martens and Foster Provost. “Explaining Data-driven Document Classifications”. In: *MIS Q.* 38.1 (Mar. 2014), pp. 73–100. ISSN: 0276-7783.

[Nak+16] Preslav Nakov et al. “Developing a successful SemEval task in sentiment analysis of Twitter and other social media texts”. In: *Language Resources and Evaluation* 50.1 (2016), pp. 35–65.

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “"Why Should I Trust You?": Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’16. San Francisco, California, USA: ACM, 2016, pp. 1135–1144.

[Zha+15] Yudong Zhang et al. “Detection of subjects and brain regions related to Alzheimer’s disease using 3D MRI scans based on eigenbrain and machine learning”. In: *Frontiers in Computational Neuroscience* 9 (2015), p. 66.