The Intriguing Properties of Model Explanations

Maruan Al-Shedivat
Carnegie Mellon University
alshedivat@cs.cmu.edu

Avinava Dubey
Carnegie Mellon University
akdubey@cs.cmu.edu

Eric P. Xing
Carnegie Mellon University
epxing@cs.cmu.edu

Abstract

Linear approximations to the decision boundary of a complex model have become one of the most popular tools for interpreting predictions. In this paper, we study such linear explanations produced either *post-hoc* by a few recent methods or generated along with predictions with *contextual explanation networks* (CENs). We focus on two questions: (i) whether linear explanations are always consistent or can be misleading, and (ii) when integrated into the prediction process, whether and how explanations affect performance of the model. Our analysis sheds more light on certain properties of explanations produced by different methods and suggests that learning models that explain and predict jointly is often advantageous.

1 Introduction

Model interpretability is a long-standing problem in machine learning that has become quite acute with the accelerating pace of widespread adoption of complex predictive algorithms. There are multiple approaches to interpreting models and their predictions ranging from a variety of visualization techniques [1–3] to explanations by example [4, 5]. The approach that we consider in this paper thinks of explanations as models themselves that approximate the decision boundary of the original predictor but belong to a class that is significantly simpler (e.g., local linear approximations).

Explanations can be generated either *post-hoc* or alongside predictions. A popular method, called LIME [6], takes the first approach and attempts to explain predictions of an arbitrary model by searching for linear local approximations of the decision boundary. On the other hand, recently proposed contextual explanation networks (CENs) [7] incorporate a similar mechanism directly into deep neural networks of arbitrary architecture and learn to predict and to explain jointly. Here, we focus on analyzing a few properties of the explanations generated by LIME, its variations, and CEN. In particular, we seek answers to the following questions:

1. Explanations are as good as the features they use to explain predictions. We ask whether and how feature selection and feature noise affect consistency of explanations.
2. When explanation is a part of the learning and prediction process, how does that affect performance of the predictive model?
3. Finally, what kind of insight we can gain by visualizing and inspecting explanations?

2 Methods

We start with a brief overview of the methods compared in this paper: LIME [6] and CENs [7]. Given a dataset of inputs, $\mathbf{x} \in \mathcal{X}$, and targets, $y \in \mathcal{Y}$, our goal is to learn a predictive model, $f : \mathcal{X} \mapsto \mathcal{Y}$. To explain each prediction, we have access to another set of features, $\mathbf{z} \in \mathcal{Z}$, and construct explanations, $g_{\mathbf{x}} : \mathcal{Z} \mapsto \mathcal{Y}$, such that they are consistent with the original model, $g_{\mathbf{x}}(\mathbf{z}) = f(\mathbf{x})$. These additional features, $\mathbf{z}$, are assumed to be more interpretable than $\mathbf{x}$, and are called *interpretable representation* in [6] and *attributes* in [7].

Interpretable ML Symposium, 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

2.1 LIME and Variations

Given a trained model, f, and an instance with features $(\mathbf{x}, \mathbf{z})$, LIME constructs an explanation, $g_{\mathbf{x}}$, as follows:

$$g_{\mathbf{x}} = \operatorname*{argmin}_{g \in \mathcal{G}} \mathcal{L}(f, g, \pi_{\mathbf{x}}) + \Omega(g) \tag{1}$$

where $\mathcal{L}(f, g, \pi_{\mathbf{x}})$ is the loss that measures how well g approximates f in the neighborhood defined by the similarity kernel, $\pi_{\mathbf{x}} : \mathcal{Z} \mapsto \mathbb{R}_+$, in the space of additional features, $\mathcal{Z}$, and $\Omega(g)$ is the penalty on the complexity of explanation. Now more specifically, Ribeiro et al. [6] assume that $\mathcal{G}$ is the class of linear models:

$$g_{\mathbf{x}}(\mathbf{z}) := b_{\mathbf{x}} + \mathbf{w}_{\mathbf{x}} \cdot \mathbf{z} \tag{2}$$

and define the loss and the similarity kernel as follows:

$$\mathcal{L}(f, g, \pi_{\mathbf{x}}) := \sum_{\mathbf{z}' \in \mathcal{Z}} \pi_{\mathbf{x}}(\mathbf{z}') \left(f(\mathbf{x}') - g(\mathbf{z}')\right)^2, \quad \pi_{\mathbf{x}}(\mathbf{z}') := \exp\left\{-D(\mathbf{z}, \mathbf{z}')^2/\sigma^2\right\} \tag{3}$$

where the data instance is represented by $(\mathbf{x}, \mathbf{z})$, $\mathbf{z}'$ and the corresponding $\mathbf{x}'$ are the perturbed features, $D(\mathbf{z}, \mathbf{z}')$ is some distance function, and σ is the scale parameter of the kernel. $\Omega(g)$ is further chosen to favor sparsity of explanations.

2.2 Contextual Explanation Networks

LIME is a *post-hoc* model explanation method. This means that it justifies model predictions by producing explanations which, while locally correct, are never used to make the predictions in the first place. Contrary to that, CENs use explanations as the integral part of the learning process and make predictions by *applying* generated explanations. Now more formally, CENs construct the predictive model $f : \mathcal{X} \times \mathcal{Z} \mapsto \mathcal{Y}$ via a composition: given $\mathbf{x}$, an encoder, $e_\theta : \mathcal{X} \mapsto \mathcal{G}$, produces an explanation g which is further applied to $\mathbf{z}$ to make a prediction. In other words:

$$f(\mathbf{x}, \mathbf{z}) := g_{\mathbf{x}}(\mathbf{z}), \text{ where } g_{\mathbf{x}} := e_\theta(\mathbf{x}) \tag{4}$$

In [7] we introduced a more general probabilistic framework that allows to combine different deterministic and probabilistic encoders with explanations represented by arbitrary graphical models. To keep our discussion simple and concrete, here we assume that explanations take the same linear form (2) as for LIME and the encoder maps $\mathbf{x}$ to $(b_{\mathbf{x}}, \mathbf{w}_{\mathbf{x}})$ as follows:

$$b_{\mathbf{x}} := \boldsymbol{\alpha}_\theta(\mathbf{x})^\top B, \quad \mathbf{w}_{\mathbf{x}} := \boldsymbol{\alpha}_\theta(\mathbf{x})^\top W, \quad \text{where } \sum_{k=1}^{K} \boldsymbol{\alpha}_\theta^{(k)}(\mathbf{x}) = 1, \forall k : \boldsymbol{\alpha}_\theta^{(k)}(\mathbf{x}) \geq 0 \tag{5}$$

In other words, explanation $(b_{\mathbf{x}}, \mathbf{w}_{\mathbf{x}})$ is constrained to be a convex combination of K components from a global learnable dictionary, $D := (B, W)$, where the combination weights, $\boldsymbol{\alpha}_\theta(\mathbf{x})$, also called *attention*, are produced by a deep network. Encoder of such form is called *constrained deterministic map* in [7] and the model is trained jointly w.r.t. (θ, B, W) to minimize the prediction error.

3 Analysis

Both LIME and CEN produce explanations in the form of linear models that can be further used for prediction diagnostics. Our goal is to understand how different conditions affect explanations generated by both methods, see whether this may lead to erroneous conclusions, and finally understand how jointly learning to predict and to explain affects performance.

We use the following 3 tasks in our analysis: MNIST image classification[1], sentiment classification of the IMDB reviews [8], and poverty prediction for households in Uganda from satellite imagery and survey data [9]. The details of the setup are omitted in the interest of space but can be found in [7], as we follow exactly the same setup.

3.1 Consistency of Explanations

Linear explanation assign weights to the interpretable features, $\mathbf{z}$, and hence strongly depend their quality and the way we select them. We consider two cases where (a) the features are corrupted with additive noise, and (b) selected features are incomplete. For analysis, we use MNIST and IMDB data.

[1] `http://yann.lecun.com/exdb/mnist/`

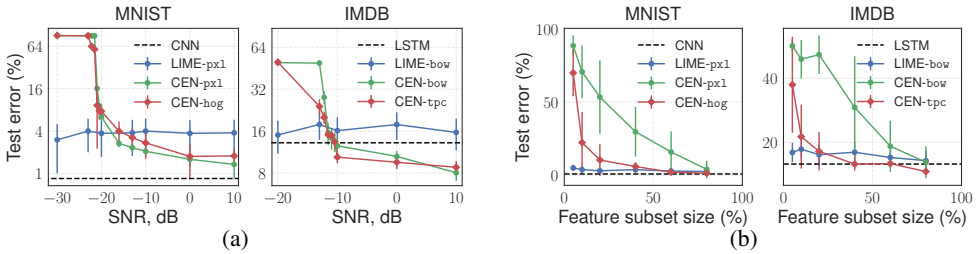


Fig. 1: The effect of feature quality on explanations. (a) Explanation test error vs. the level of the noise added to the interpretable features. (b) Explanation test error vs. the total number of interpretable features.

We train baseline deep architectures (CNN on MNIST and LSTM on IMDB) and their CEN variants. For MNIST, $\mathbf{z}$ is either pixels of a scaled down image (`pxl`) or HOG features (`hog`). For IMDB, $\mathbf{z}$ is either a bag of words (`bow`) or a topic vector (`tpc`) produced by a pre-trained topic model.

The effect of noisy features. In this experiment, we inject noise[2] into the features $\mathbf{z}$ and ask LIME and CEN to fit explanations to the noisy features. The predictive performance of the produced explanations on noisy features is given on Fig. 1a. Note that after injecting noise, each data point has a noiseless representation $\mathbf{x}$ and noisy $\tilde{\mathbf{z}}$. Since baselines take only $\mathbf{x}$ as inputs, their performance stays the same and, regardless of the noise level, LIME "successfully" overfits explanations—it is able to almost perfectly approximate the decision boundary of the baselines using very noisy features. On the other hand, performance of CEN gets worse with the increasing noise level indicating that model fails to learn when the selected interpretable representation is low quality.

The effect of feature selection. Here, we use the same setup, but instead of injecting noise into $\mathbf{z}$, we construct $\tilde{\mathbf{z}}$ by randomly subsampling a set of dimensions. Fig. 1b demonstrates the result. While performance of CENs degrades proportionally to the size of $\tilde{\mathbf{z}}$, we see that, again, LIME is able to fit explanations to the decision boundary of the original models despite the loss of information.

These two experiments indicate a major drawback of explaining predictions *post-hoc*: when constructed on poor, noisy, or incomplete features, such explanations can overfit the decision boundary of a predictor and are likely to be misleading. For example, predictions of a perfectly valid model might end up getting absurd explanations which is unacceptable from the decision support point of view.

3.2 Explanations as a Regularizer

In this part, we compare CENs with baselines in terms of performance. In each task, CENs are trained to simultaneously generate predictions and construct explanations. Overall, CENs show very competitive performance and are able to approach or surpass baselines in a number of cases, especially on the IMDB data (see Table 1). This suggests that forcing the model to produce explanations along with predictions does not limit its capacity.

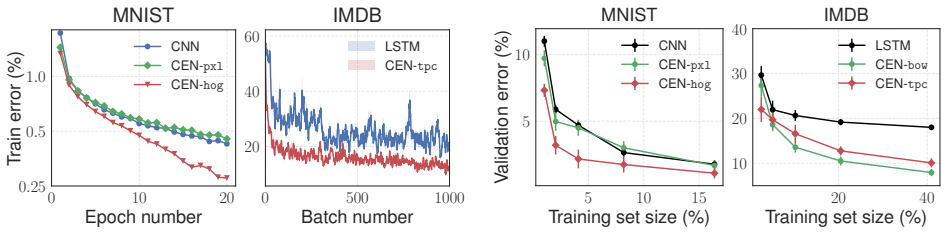


Fig. 2: (a) Training error vs. iteration (epoch or batch) for baselines and CENs. (b) Validation error for models trained on random subsets of data of different sizes.

Additionally, the "explanation layer" in CENs affects the geometry of the optimization problem and causes faster and better convergence (Fig. 2a). Finally, we train the models on subsets of data (the size varied from 1% to 20% for MNIST and from 2% to 40% for IMDB) and notice that explanations play the role of a regularizer which strongly improves the sample complexity (Fig. 2b).

[2]We use Gaussian noise with zero mean and select variance for each signal-to-noise ratio level appropriately.

Table 1: Performance of the models on classification tasks (averaged over 5 runs; the std. are on the order of the least significant digit). The subscripts denote the features on which the linear models are built: pixels (`pxl`), HOG (`hog`), bag-or-words (`bow`), topics (`tpc`), embeddings (`emb`), discrete attributes (`att`).

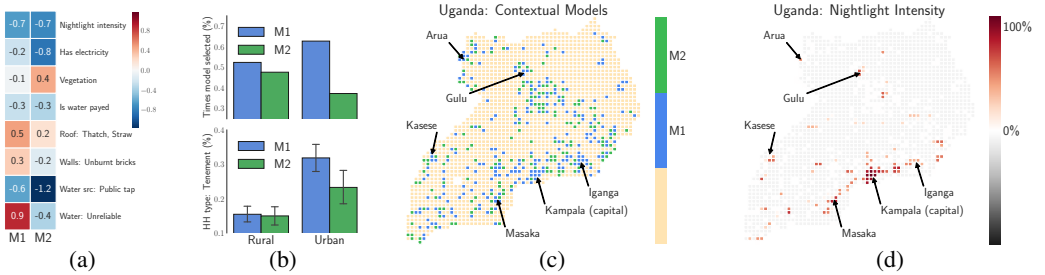
MNIST		IMDB		Satellite		
Model	**Err (%)**	**Model**	**Err (%)**	**Model**	**Acc (%)**	**AUC (%)**
$\text{LR}_{\texttt{pxl}}$	8.00	$\text{LR}_{\texttt{bow}}$	13.3	$\text{LR}_{\texttt{emb}}$	62.5	68.1
$\text{LR}_{\texttt{hog}}$	2.98	$\text{LR}_{\texttt{tpc}}$	17.1	$\text{LR}_{\texttt{att}}$	75.7	82.2
CNN	**0.75**	LSTM	13.2	MLP	77.4	78.7
$\text{MoE}_{\texttt{pxl}}$	1.23	$\text{MoE}_{\texttt{bow}}$	13.9	MoE	77.9	**85.4**
$\text{MoE}_{\texttt{hog}}$	1.10	$\text{MoE}_{\texttt{tpc}}$	12.2	CEN	81.5	84.2
$\text{CEN}_{\texttt{pxl}}$	**0.76**	$\text{CEN}_{\texttt{bow}}$	***6.9**	VCEN	**83.4**	84.6
$\text{CEN}_{\texttt{hog}}$	**0.73**	$\text{CEN}_{\texttt{tpc}}$	*7.8			

* Best previous results for similar LSTMs: 8.1% (supervised) and 6.6% (semi-supervised) [10].

3.3 Visualizing Explanations

Finally, we showcase the insights one can get from explanations produced along with predictions. Particularly, we consider the problem of poverty prediction for household clusters in a Uganda from satellite imagery and survey data. The $\mathbf{x}$ representation of each household cluster is a collection of 400×400 satellite images; $\mathbf{z}$ is represented by a vector of 65 categorical features from living standards measurement survey (LSMS). The goal is binary classification of households in Uganda into poor and not poor. In our methodology, we closely follow the original study of Jean et al. [9] and use a pretrained `VGG-F` network for embedding the images into a 4096-dimensional space on top of which we build our contextual models. Note that this datasets is fairly small (642 points), and hence we keep the `VGG-F` frozen to avoid overfitting. We note that quantitatively, by conditioning on the VGG features of the satellite imagery, CENs are able to significantly improve upon the sparse linear models on the survey features only (known as the gold standard in remote sensing techniques).

After training CEN with a dictionary of size 32, we discover that the encoder tends to sharply select one of the two explanations (M1 and M2) for different household clusters in Uganda (see Fig. 3a and also Fig. 4a in appendix). In the survey data, each household cluster is marked as either urban or rural; we notice that, conditional on a satellite image, CEN tends to pick M1 for urban areas and M2 for rural (Fig. 3b). Notice that explanations weigh different categorical features, such as reliability of the water source or the proportion of houses with walls made of unburnt brick, quite differently. When visualized on the map, we see that CEN selects M1 more frequently around the major city areas, which also correlates with high nightlight intensity in those areas (Fig. 3c,3d). High performance of the model makes us confident in the produced explanations (contrary to LIME as discussed in Sec. 3.1) and allows us to draw conclusions about what causes the model to classify certain households in different neighborhoods as poor.



(a) (b) (c) (d)

Fig. 3: Qualitative results for the Satellite dataset: (a) Weights given to a subset of features by the two models (M1 and M2) discovered by CEN. (b) How frequently M1 and M2 are selected for areas marked rural or urban (top) and the average proportion of Tenement-type households in an urban/rural area for which M1 or M2 was selected. (c) M1 and M2 models selected for different areas on the Uganda map. M1 tends to be selected for more urbanized areas while M2 is picked for the rest. (d) Nightlight intensity of different areas of Uganda.

References

[1] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[2] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

[3] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.

[4] Rich Caruana, Hooshang Kangarloo, JD Dionisio, Usha Sinha, and David Johnson. Case-based explanation of non-case-based learning methods. In *Proceedings of the AMIA Symposium*, page 212, 1999.

[5] Been Kim, Cynthia Rudin, and Julie A Shah. The bayesian case model: A generative approach for case-based reasoning and prototype classification. In *Advances in Neural Information Processing Systems*, pages 1952–1960, 2014.

[6] 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*, pages 1135–1144. ACM, 2016.

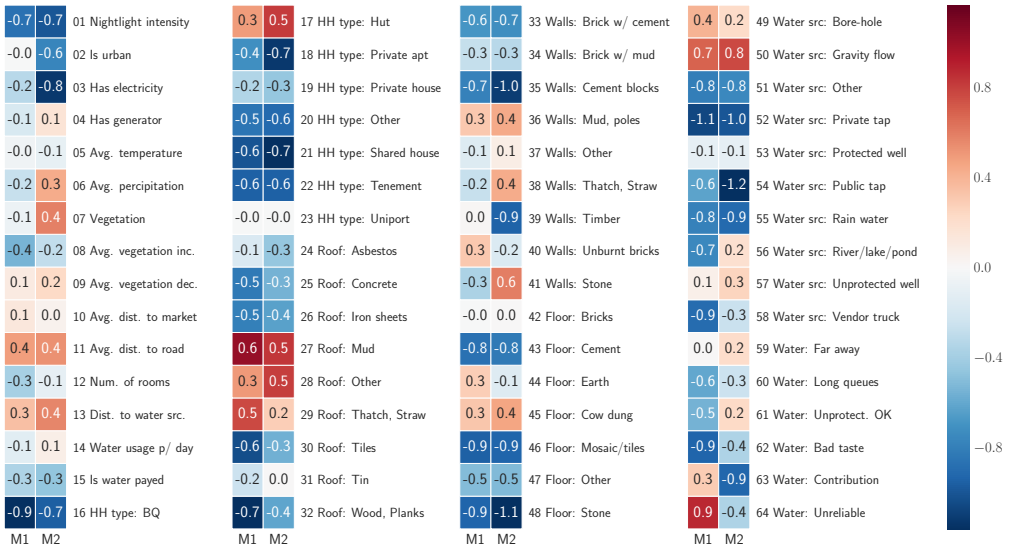
[7] Maruan Al-Shedivat, Avinava Dubey, and Eric P Xing. Contextual explanation networks. *arXiv preprint arXiv:1705.10301*, 2017.

[8] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 142–150. Association for Computational Linguistics, 2011.

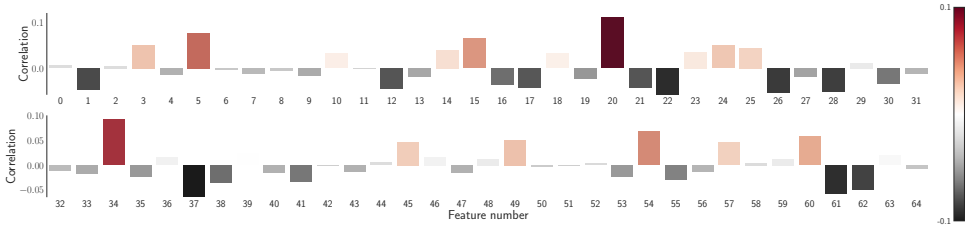
[9] Neal Jean, Marshall Burke, Michael Xie, W Matthew Davis, David B Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 353 (6301):790–794, 2016.

[10] Rie Johnson and Tong Zhang. Supervised and semi-supervised text categorization using lstm for region embeddings. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 526–534, 2016.

A Appendix



(a) Full visualization of explanations M1 and M2 learned by CEN on the poverty prediction task.



(b) Correlation between the selected explanation and the value of a particular survey variable.

Fig. 4: Additional visualizations for the poverty prediction task.

B Details on Consistency of Explanations

We provide a detailed description of the experimental setup used for our analysis in Section 3.1.