Learning to Explain: An Information-Theoretic Perspective on Model Interpretation

Jianbo Chen [1 2] **Le Song** [3 4] **Martin J. Wainwright** [1 5] **Michael I. Jordan** [1]

Abstract

We introduce *instancewise feature selection* as a methodology for model interpretation. Our method is based on learning a function to extract a subset of features that are most informative for each given example. This feature selector is trained to maximize the mutual information between selected features and the response variable, where the conditional distribution of the response variable given the input is the model to be explained. We develop an efficient variational approximation to the mutual information, and show the effectiveness of our method on a variety of synthetic and real data sets using both quantitative metrics and human evaluation.

1. Introduction

Interpretability is an extremely important criterion when a machine learning model is applied in areas such as medicine, financial markets, and criminal justice (e.g., see the discussion paper by Lipton ((Lipton, 2016)), as well as references therein). Many complex models, such as random forests, kernel methods, and deep neural networks, have been developed and employed to optimize prediction accuracy, which can compromise their ease of interpretation.

In this paper, we focus on *instancewise feature selection* as a specific approach for model interpretation. Given a machine learning model, instancewise feature selection asks for the importance score of each feature on the prediction of a given instance, and the relative importance of each feature is allowed to vary across instances. Thus, the importance scores can act as an explanation for the specific instance, indicating which features are the key for the model to make its prediction on that instance. A related concept in machine learning is feature selection, which selects a subset of features that are useful to build a good predictor for a specified response variable (Guyon & Elisseeff, 2003). While feature selection produces a global importance of features with respect to the entire labeled data set, instancewise feature selection measures feature importance locally for each instance labeled by the model.

Existing work on interpreting models approach the problem from two directions. The first line of work computes the gradient of the output of the correct class with respect to the input vector for the given model, and uses it as a saliency map for masking the input (Simonyan et al., 2013; Springenberg et al., 2014). The gradient is computed using a Parzen window approximation of the original classifier if the original one is not available (Baehrens et al., 2010). Another line of research approximates the model to be interpreted via a locally additive model in order to explain the difference between the model output and some "reference" output in terms of the difference between the input and some "reference" input (Bach et al., 2015; Kindermans et al., 2016; Ribeiro et al., 2016; Lundberg & Lee, 2017; Shrikumar et al., 2017; Sundararajan et al., 2017). Ribeiro et al. (2016) proposed the LIME, methods which randomly draws instances from a density centered at the sample to be explained, and fits a sparse linear model to predict the model outputs for these instances. Shrikumar et al. (2017) presented DeepLIFT, a method designed specifically for neural networks, which decomposes the output of a neural network on a specific input by backpropagating the contribution back to every feature of the input. Lundberg & Lee (2017) used Shapley values to quantify the importance of features of a given input, and proposed a sampling based method "kernel SHAP" for approximating Shapley values. (Sundararajan et al., 2017) proposed Integrated Gradients (IG), which constructs the additive model by cumulating the gradients along the line between the input and the reference point. Essentially, the two directions both approximate the model locally via an additive model, with different definitions of locality. While the first one considers infinitesimal regions on the decision surface and takes the first-order term in the Taylor expansion as the additive model, the second one considers the finite difference between an input vector and a reference vector.

[1]University of California, Berkeley [2]Work done partially during an internship at Ant Financial [3]Georgia Institute of Technology [4]Ant Financial [5]The Voleon Group. Correspondence to: Jianbo Chen <jianbochen@berkeley.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

	Training	Efficiency	Additive	Model-agnostic
Parzen (Baehrens et al., 2010)	Yes	High	Yes	Yes
Salient map (Simonyan et al., 2013)	No	High	Yes	No
LRP (Bach et al., 2015)	No	High	Yes	No
LIME (Ribeiro et al., 2016)	No	Low	Yes	Yes
Kernel SHAP (Lundberg & Lee, 2017)	No	Low	Yes	Yes
DeepLIFT (Shrikumar et al., 2017)	No	High	Yes	No
IG (Sundararajan et al., 2017)	No	Medium	Yes	No
L2X	Yes	High	No	Yes

Table 1. Summary of the properties of different methods. "Training" indicates whether a method requires training on an unlabeled data set. "Efficiency" qualitatively evaluates the computational time during single interpretation. "Additive" indicates whether a method is locally additive. "Model-agnostic" indicates whether a method is generic to black-box models.

In this paper, our approach to instancewise feature selection is via mutual information, a conceptually different perspective from existing approaches. We define an "explainer," or instancewise feature selector, as a model which returns a distribution over the subset of features given the input vector. For a given instance, an ideal explainer should assign the highest probability to the subset of features that are most informative for the associated model response. This motivates us to maximize the mutual information between the selected subset of features and the response variable with respect to the instancewise feature selector. Direct estimation of mutual information and discrete feature subset sampling are intractable; accordingly, we derive a tractable method by first applying a variational lower bound for mutual information, and then developing a continuous reparametrization of the sampling distribution.

At a high level, the primary differences between our approach and past work are the following. First, our framework *globally* learns a *local* explainer, and therefore takes the distribution of inputs into consideration. Second, our framework removes the constraint of local feature additivity on an explainer. These distinctions enable our framework to yield a more efficient, flexible, and natural approach for instancewise feature selection. In summary, our contributions in this work are as follows (see also Table 1 for systematic comparisons):

- We propose an information-based framework for instancewise feature selection.
- We introduce a learning-based method for instancewise feature selection, which is both efficient and model-agnostic.

Furthermore, we show that the effectiveness of our method on a variety of synthetic and real data sets using both quantitative metric and human evaluation on Amazon Mechanical Turk.

2. A framework

We now lay out the primary ingredients of our general approach. While our framework is generic and can be applied to both classification and regression models, the current discussion is restricted to classification models. We assume one has access to the output of a model as a conditional distribution, $\mathbb{P}_m(\cdot \mid x)$, of the response variable Y given the realization of the input random variable $X = x \in \mathbb{R}^d$.

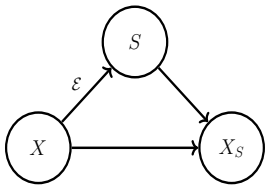

Figure 1. The graphical model of obtaining X_S from X.

2.1. Mutual information

Our method is derived from considering the mutual information between a particular pair of random vectors, so we begin by providing some basic background. Given two random vectors X and Y, the *mutual information* $I(X;Y)$ is a measure of dependence between them; intuitively, it corresponds to how much knowledge of one random vector reduces the uncertainty about the other. More precisely, the mutual information is given by the Kullback-Leibler divergence of the product of marginal distributions of X and Y from the joint distribution of X and Y (Cover & Thomas, 2012); it takes the form

$$I(X;Y) = \mathbb{E}_{X,Y}\left[\log \frac{p_{XY}(X,Y)}{p_X(X)p_Y(Y)}\right],$$

where p_{XY} and p_X, p_Y are the joint and marginal probability densities if X, Y are continuous, or the joint and marginal probability mass functions if they are discrete. The expectation is taken with respect to the joint distribution of X and Y. One can show the mutual information is nonnegative and symmetric in two random variables. The mutual information has been a popular criteria in feature selection, where one selects the subset of features that approximately maximizes the mutual information between the response variable and the selected features (Gao et al., 2016; Peng et al., 2005). Here we propose to use mutual information as a criteria for instancewise feature selection.

2.2. How to construct explanations

We now describe how to construct explanations using mutual information. In our specific setting, the pair (X, Y) are characterized by the marginal distribution $X \sim \mathbb{P}_X(\cdot)$, and a family of conditional distributions of the form $(Y \mid x) \sim \mathbb{P}_m(\cdot \mid x)$. For a given positive integer k, let $\wp_k = \{S \subset 2^d \mid |S| = k\}$ be the set of all subsets of size k. An *explainer* $\mathcal{E}$ of size k is a mapping from the feature space $\mathbb{R}^d$ to the power set $\wp_k$; we allow the mapping to be randomized, meaning that we can also think of $\mathcal{E}$ as mapping x to a conditional distribution $\mathbb{P}(S \mid x)$ over $S \in \wp_k$. Given the chosen subset $S = \mathcal{E}(x)$, we use x_S to denote the sub-vector formed by the chosen features. We view the choice of the number of explaining features k as

best left in the hands of the user, but it can also be tuned as a hyper-parameter.

We have thus defined a new random vector $X_S \in \mathbb{R}^k$; see Figure 1 for a probabilistic graphical model representing its construction. We formulate instancewise feature selection as seeking explainer that optimizes the criterion

$$\max_{\mathcal{E}} I(X_S; Y) \quad \text{subject to} \quad S \sim \mathcal{E}(X). \tag{1}$$

In words, we aim to maximize the mutual information between the response variable from the model and the selected features, as a function of the choice of selection rule.

It turns out that a global optimum of Problem (1) has a natural information-theoretic interpretation: it corresponds to the minimization of the expected length of encoded message for the model $\mathbb{P}_m(Y \mid x)$ using $\mathbb{P}_m(Y|x_S)$, where the latter corresponds to the conditional distribution of Y upon observing the selected sub-vector. More concretely, we have the following:

Theorem 1. *Letting $\mathbb{E}_m[\cdot \mid x]$ denote the expectation over $\mathbb{P}_m(\cdot \mid x)$, define*

$$\mathcal{E}^*(x) := \arg\min_S \; \mathbb{E}_m \left[\log \frac{1}{\mathbb{P}_m(Y \mid x_S)} \;\middle|\; x \right]. \tag{2}$$

Then $\mathcal{E}^*$ is a global optimum of Problem (1). Conversely, any global optimum of Problem (1) degenerates to $\mathcal{E}^*$ almost surely over the marginal distribution $\mathbb{P}_X$.

The proof of Theorem 1 is left to the supplementary materials. In practice, the above global optimum is obtained only if the explanation family $\mathcal{E}$ is sufficiently large. In the case when $\mathbb{P}_m(Y|x_S)$ is unknown or computationally expensive to estimate accurately, we can choose to restrict $\mathcal{E}$ to suitably controlled families so as to prevent overfitting.

3. Proposed method

A direct solution to Problem (1) is not possible, so that we need to approach it by a variational approximation. In particular, we derive a lower bound on the mutual information, and we approximate the model conditional distribution $\mathbb{P}_m$ by a suitably rich family of functions.

3.1. Obtaining a tractable variational formulation

We now describe the steps taken to obtain a tractable variational formulation.

A variational lower bound: Mutual information between X_S and Y can be expressed in terms of the conditional distribution of Y given X_S:

$$\begin{aligned} I(X_S; Y) &= \mathbb{E}\Big[\log \frac{\mathbb{P}_m(X_S, Y)}{\mathbb{P}(X_S)\mathbb{P}_m(Y)} \Big] = \mathbb{E}\Big[\log \frac{\mathbb{P}_m(Y|X_S)}{\mathbb{P}_m(Y)} \Big] \\ &= \mathbb{E}\Big[\log \mathbb{P}_m(Y|X_S) \Big] + \text{Const.} \\ &= \mathbb{E}_X \mathbb{E}_{S|X} \mathbb{E}_{Y|X_S} \Big[\log \mathbb{P}_m(Y|X_S) \Big] + \text{Const.} \end{aligned}$$

For a generic model, it is impossible to compute expectations under the conditional distribution $\mathbb{P}_m(\cdot \mid x_S)$. Hence we introduce a variational family for approximation:

$$\mathcal{Q} := \Big\{ \mathbb{Q} \mid \mathbb{Q} = \{x_S \to \mathbb{Q}_S(Y|x_S), S \in \wp_k\} \Big\}. \tag{3}$$

Note each member $\mathbb{Q}$ of the family $\mathcal{Q}$ is a collection of conditional distributions $\mathbb{Q}_S(Y|x_S)$, one for each choice of k-sized feature subset S. For any $\mathbb{Q}$, an application of Jensen's inequality yields the lower bound

$$\begin{aligned} \mathbb{E}_{Y|X_S}[\log \mathbb{P}_m(Y|X_S)] &\geq \int \mathbb{P}_m(Y|X_S) \log \mathbb{Q}_S(Y|X_S) \\ &= \mathbb{E}_{Y|X_S}[\log \mathbb{Q}_S(Y|X_S)], \end{aligned}$$

where equality holds if and only if $\mathbb{P}_m(Y|X_S)$ and $\mathbb{Q}_S(Y|X_S)$ are equal in distribution. We have thus obtained a variational lower bound of the mutual information $I(X_S; Y)$. Problem (1) can thus be relaxed as maximizing the variational lower bound, over both the explanation $\mathcal{E}$ and the conditional distribution $\mathbb{Q}$:

$$\max_{\mathcal{E}, \mathbb{Q}} \mathbb{E}\Big[\log \mathbb{Q}_S(Y \mid X_S) \Big] \qquad \text{such that } S \sim \mathcal{E}(X). \tag{4}$$

For generic choices $\mathbb{Q}$ and $\mathcal{E}$, it is still difficult to solve the variational approximation (4). In order to obtain a tractable method, we need to restrict both $\mathbb{Q}$ and $\mathcal{E}$ to suitable families over which it is efficient to perform optimization.

A single neural network for parametrizing $\mathbb{Q}$: Recall that $\mathbb{Q} = \{\mathbb{Q}_S(\cdot \mid x_S),\ S \in \wp_k\}$ is a collection of conditional distributions with cardinality $|\mathbb{Q}| = \binom{d}{k}$. We assume X is a continuous random vector, and $\mathbb{P}_m(Y \mid x)$ is continuous with respect to x. Then we introduce a single neural network function $g_\alpha : \mathbb{R}^d \times [c] \to [0, 1]$ for parametrizing $\mathbb{Q}$, where $[c] = \{0, 1, \ldots, c-1\}$ denotes the set of possible classes, and α denotes the learnable parameters. We define $\mathbb{Q}_S(Y|x_S) := g_\alpha(\tilde{x}_S, Y)$, where $\tilde{x}_S \in \mathbb{R}^d$ is transformed from x by replacing entries not in S with zeros:

$$(\tilde{x}_S)_i = \begin{cases} x_i, i \in S, \\ 0, i \notin S. \end{cases}$$

When X contains discrete features, we embed each discrete feature with a vector, and the vector representing a specific feature is set to zero simultaneously when the corresponding feature is not in S.

3.2. Continuous relaxation of subset sampling

Direct estimation of the objective function in equation (4) requires summing over $\binom{d}{k}$ combinations of feature subsets after the variational approximation. Several tricks exist for tackling this issue, like REINFORCE-type Algorithms (Williams, 1992), or weighted sum of features parametrized by deterministic functions of X. (A similar concept to the second trick is the "soft attention" structure in vision (Ba et al., 2014) and NLP (Bahdanau et al., 2014) where the weight of each feature is parametrized by

a function of the respective feature itself.) We employ an alternative approach generalized from Concrete Relaxation (Gumbel-softmax trick) (Jang et al., 2017; Maddison et al., 2014; 2016), which empirically has a lower variance than REINFORCE and encourages discreteness (Raffel et al., 2017).

The Gumbel-softmax trick uses the concrete distribution as a continuous differentiable approximation to a categorical distribution. In particular, suppose we want to approximate a categorical random variable represented as a one-hot vector in $\mathbb{R}^d$ with category probability $p_1, p_2, \ldots, p_d$. The random perturbation for each category is independently generated from a Gumbel$(0, 1)$ distribution:

$$G_i = -\log(-\log u_i), u_i \sim \text{Uniform}(0, 1).$$

We add the random perturbation to the log probability of each category and take a temperature-dependent softmax over the d-dimensional vector:

$$C_i = \frac{\exp\{(\log p_i + G_i)/\tau\}}{\sum_{j=1}^{d} \exp\{(\log p_j + G_j)/\tau\}}.$$

The resulting random vector $C = (C_1, \ldots, C_d)$ is called a Concrete random vector, which we denote by

$$C \sim \text{Concrete}(\log p_1, \ldots, \log p_d).$$

We apply the Gumbel-softmax trick to approximate weighted subset sampling. We would like to sample a subset S of k distinct features out of the d dimensions. The sampling scheme for S can be equivalently viewed as sampling a k-hot random vector Z from $D_k^d := \{z \in \{0, 1\}^d \mid \sum z_i = k\}$, with each entry of z being one if it is in the selected subset S and being zero otherwise. An importance score which depends on the input vector is assigned for each feature. Concretely, we define $w_\theta \colon \mathbb{R}^d \to \mathbb{R}^d$ that maps the input to a d-dimensional vector, with the ith entry of $w_\theta(X)$ representing the importance score of the ith feature.

We start with approximating sampling k distinct features out of d features by the sampling scheme below: Sample a single feature out of d features independently for k times. Discard the overlapping features and keep the rest. Such a scheme samples at most k features, and is easier to approximate by a continuous relaxation. We further approximate the above scheme by independently sampling k independent Concrete random vectors, and then we define a d-dimensional random vector V that is the elementwise maximum of $C^1, C^2, \ldots, C^k$:

$$C^j \sim \text{Concrete}(w_\theta(X)) \text{ i.i.d. for } j = 1, 2, \ldots, k,$$
$$V = (V_1, V_2, \ldots, V_d), \quad V_i = \max_j C_i^j.$$

The random vector V is then used to approximate the k-hot random vector Z during training.

We write $V = V(\theta, \zeta)$ as V is a function of θ and a collection of auxiliary random variables ζ sampled independently from the Gumbel distribution. Then we use the elementwise product $V(\theta, \zeta) \odot X$ between V and X as an approximation of $\tilde{X}_S$.

3.3. The final objective and its optimization

After having applied the continuous approximation of feature subset sampling, we have reduced Problem (4) to the following:

$$\max_{\theta, \alpha} \mathbb{E}_{X,Y,\zeta}\Big[\log g_\alpha(V(\theta, \zeta) \odot X, Y)\Big], \tag{5}$$

where g_α denotes the neural network used to approximate the model conditional distribution, and the quantity θ is used to parametrize the explainer. In the case of classification with c classes, we can write

$$\mathbb{E}_{X,\zeta}\Big[\sum_{y=1}^{c} [\mathbb{P}_m(y \mid X) \log g_\alpha(V(\theta, \zeta) \odot X, y)\Big]. \tag{6}$$

Note that the expectation operator $\mathbb{E}_{X,\zeta}$ does not depend on the parameters (α, θ), so that during the training stage, we can apply stochastic gradient methods to jointly optimize the pair (α, θ). In each update, we sample a mini-batch of unlabeled data with their class distributions from the model to be explained, and the auxiliary random variables ζ, and we then compute a Monte Carlo estimate of the gradient of the objective function (6).

3.4. The explaining stage

During the explaining stage, the learned explainer maps each sample X to a weight vector $w_\theta(X)$ of dimension d, each entry representing the importance of the corresponding feature for the specific sample X. In order to provide a deterministic explanation for a given sample, we rank features according to the weight vector, and the k features with the largest weights are picked as the explaining features.

For each sample, only a single forward pass through the neural network parametrizing the explainer is required to yield explanation. Thus our algorithm is much more efficient in the explaining stage compared to other model-agnostic explainers like LIME or Kernel SHAP which require thousands of evaluations of the original model per sample.

4. Experiments

We carry out experiments on both synthetic and real data sets. For all experiments, we use RMSprop (Maddison et al., 2016) with the default hyperparameters for optimization. We also fix the step size to be 0.001 across experiments. The temperature for Gumbel-softmax approximation is fixed to be 0.1. Codes for reproducing the key results are available online at https://github.com/Jianbo-Lab/L2X.

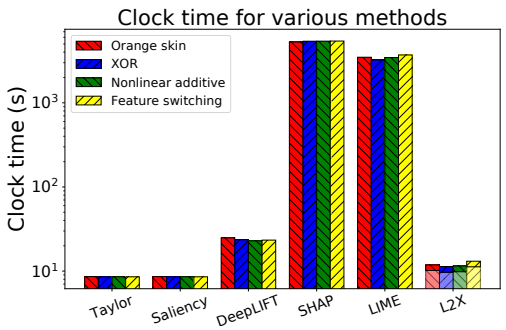

Figure 2. The clock time (in log scale) of explaining $10,000$ samples for each method. The training time of L2X is shown in translucent bars.

4.1. Synthetic Data

We begin with experiments on four synthetic data sets:

- 2-dimensional XOR as binary classification. The input vector X is generated from a 10-dimensional standard Gaussian. The response variable Y is generated from $P(Y = 1|X) \propto \exp\{X_1 X_2\}$.
- Orange Skin. The input vector X is generated from a 10-dimensional standard Gaussian. The response variable Y is generated from $P(Y = 1|X) \propto \exp\{\sum_{i=1}^{4} X_i^2 - 4\}$.
- Nonlinear additive model. Generate X from a 10-dimensional standard Gaussian. The response variable Y is generated from $P(Y = 1|X) \propto \exp\{-100\sin(2X_1) + 2|X_2| + X_3 + \exp\{-X_4\}\}$.
- Switch feature. Generate X_1 from a mixture of two Gaussians centered at ± 3 respectively with equal probability. If X_1 is generated from the Gaussian centered at 3, the $2-5$th dimensions are used to generate Y like the orange skin model. Otherwise, the $6-9th$ dimensions are used to generate Y from the nonlinear additive model.

The first three data sets are modified from commonly used data sets in the feature selection literature (Chen et al., 2017). The fourth data set is designed specifically for instancewise feature selection. Every sample in the first data set has the first two dimensions as true features, where each dimension itself is independent of the response variable Y but the combination of them has a joint effect on Y. In the second data set, the samples with positive labels centered around a sphere in a four-dimensional space. The sufficient statistic is formed by an additive model of the first four features. The response variable in the third data set is generated from a nonlinear additive model using the first four features. The last data set switches important features (roughly) based on the sign of the first feature. The $1-5$ features are true for samples with X_1 generated from the Gaussian centered at -3, and the $1, 6-9$ features are true otherwise.

We compare our method L2X (for "Learning to Explain") with several strong existing algorithms for instancewise feature selection, including Saliency (Simonyan et al., 2013), DeepLIFT (Shrikumar et al., 2017), SHAP (Lundberg & Lee, 2017), LIME (Ribeiro et al., 2016). Saliency refers to the method that computes the gradient of the selected class with respect to the input feature and uses the absolute values as importance scores. SHAP refers to Kernel SHAP. The number of samples used for explaining each instance for LIME and SHAP is set as default for all experiments. We also compare with a method that ranks features by the input feature times the gradient of the selected class with respect to the input feature. Shrikumar et al. (2017) showed it is equivalent to LRP (Bach et al., 2015) when activations are piecewise linear, and used it in Shrikumar et al. (2017) as a strong baseline. We call it "Taylor" as it is the first-order Taylor approximation of the model.

Our experimental setup is as follows. For each data set, we train a neural network model with three hidden dense layers. We can safely assume the neural network has successfully captured the important features, and ignored noise features, based on its error rate. Then we use Taylor, Saliency, DeepLIFT, SHAP, LIME, and L2X for instancewise feature selection on the trained neural network models. For L2X, the explainer is a neural network composed of two hidden layers. The variational family is composed of three hidden layers. All layers are linear with dimension 200. The number of desired features k is set to the number of true features.

The underlying true features are known for each sample, and hence the median ranks of selected features for each sample in a validation data set are reported as a performance metric, the box plots of which have been plotted in Figure 3. We observe that L2X outperforms all other methods on nonlinear additive and feature switching data sets. On the XOR model, DeepLIFT, SHAP and L2X achieve the best performance. On the orange skin model, all algorithms have near optimal performance, with L2X and LIME achieving the most stable performance across samples.

We also report the clock time of each method in Figure 2, where all experiments were performed on a single NVidia Tesla k80 GPU, coded in TensorFlow. Across all the four data sets, SHAP and LIME are the least efficient as they require multiple evaluations of the model. DeepLIFT, Taylor and Saliency requires a backward pass of the model. DeepLIFT is the slowest among the three, probably due to the fact that backpropagation of gradients for Taylor and Saliency are built-in operations of TensorFlow, while backpropagation in DeepLIFT is implemented with high-level operations in TensorFlow. Our method L2X is the most efficient in the explanation stage as it only requires a forward pass of the subset sampler. It is much more efficient compared to SHAP and LIME even after the training time has been taken into consideration, when a moderate number

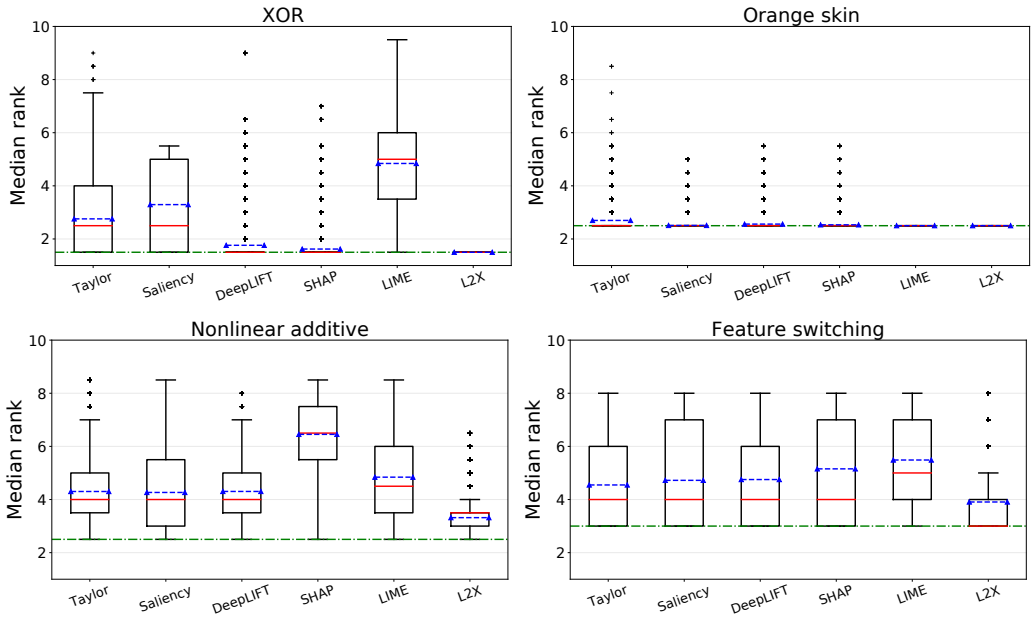

Figure 3. The box plots for the median ranks of the influential features by each sample, over $10,000$ samples for each data set. The red line and the dotted blue line on each box is the median and the mean respectively. Lower median ranks are better. The dotted green lines indicate the optimal median rank.

Truth	Model	Key words
positive	positive	Ray Liotta and Tom Hulce shine in this sterling example of brotherly love and commitment. Hulce plays Dominick, (nicky) a mildly mentally handicapped young man who is putting his 12 minutes younger, twin brother, Liotta, who plays Eugene, through medical school. It is set in Baltimore and deals with the issues of sibling rivalry, the unbreakable bond of twins, child abuse and good always winning out over evil. It is captivating, and filled with laughter and tears. If you have not yet seen this film, please rent it, I promise, you'll be amazed at how such a wonderful film could go unnoticed.
negative	negative	Sorry to go against the flow but I thought this film was unrealistic, boring and way too long. I got tired of watching Gena Rowlands long arduous battle with herself and the crisis she was experiencing. Maybe the film has some cinematic value or represented an important step for the director but for pure entertainment value. I wish I would have skipped it.
negative	positive	This movie is chilling reminder of Bollywood being just a parasite of Hollywood. Bollywood also tends to feed on past blockbusters for furthering its industry. Vidhu Vinod Chopra made this movie with the reasoning that a cocktail mix of deewar and on the waterfront will bring home an oscar. It turned out to be rookie mistake. Even the idea of the title is inspired from the Elia Kazan classic. In the original, Brando is shown as raising doves as symbolism of peace. Bollywood must move out of Hollywoods shadow if it needs to be taken seriously.
positive	negative	When a small town is threatened by a child killer, a lady police officer goes after him by pretending to be his friend. As she becomes more and more emotionally involved with the murderer her psyche begins to take a beating causing her to lose focus on the job of catching the criminal. Not a film of high voltage excitement, but solid police work and a good depiction of the faulty mind of a psychotic loser.

Table 2. True labels and labels predicted by the model are in the first two columns. Key words picked by L2X are highlighted in yellow.

of samples (10,000) need to be explained. As the scale of the data to be explained increases, the training of L2X accounts for a smaller proportion of the over-all time. Thus the relative efficiency of L2X to other algorithms increases with the size of a data set.

4.2. IMDB

The Large Movie Review Dataset (IMDB) is a dataset of movie reviews for sentiment classification (Maas et al., 2011). It contains $50,000$ labeled movie reviews, with a split of $25,000$ for training and $25,000$ for testing. The average document length is 231 words, and 10.7 sentences. We use L2X to study two popular classes of models for sentiment analysis on the IMDB data set.

4.2.1. Explaining a CNN model with key words

Convolutional neural networks (CNN) have shown excellent performance for sentiment analysis (Kim, 2014; Zhang & Wallace, 2015). We use a simple CNN model on Keras (Chollet et al., 2015) for the IMDB data set, which

Truth	Predicted	Key sentence
positive	positive	There are few really hilarious films about science fiction but this one will knock your sox off. The lead Martians Jack Nicholson take-off is side-splitting. The plot has a very clever twist that has be seen to be enjoyed. This is a movie with heart and excellent acting by all. Make some popcorn and have a great evening.
negative	negative	You get 5 writers together, have each write a different story with a different genre, and then you try to make one movie out of it. Its action, its adventure, its sci-fi, its western, its a mess. Sorry, but this movie absolutely stinks. 4.5 is giving it an awefully high rating. That said, its movies like this that make me think I could write movies, and I can barely write.
negative	positive	This movie is not the same as the 1954 version with Judy garland and James mason, and that is a shame because the 1954 version is, in my opinion, much better. I am not denying Barbra Streisand's talent at all. She is a good actress and brilliant singer. I am not acquainted with Kris Kristofferson's other work and therefore I can't pass judgment on it. However, this movie leaves much to be desired. It is paced slowly, it has gratuitous nudity and foul language, and can be very difficult to sit through. However, I am not a big fan of rock music, so its only natural that I would like the judy garland version better. See the 1976 film with Barbra and Kris, and judge for yourself.
positive	negative	The first time you see the second renaissance it may look boring. Look at it at least twice and definitely watch part 2. it will change your view of the matrix. Are the human people the ones who started the war? Is ai a bad thing?

Table 3. True labels and labels from the model are shown in the first two columns. Key sentences picked by L2X highlighted in yellow.

is composed of a word embedding of dimension 50, a 1-D convolutional layer of kernel size 3 with 250 filters, a max-pooling layer and a dense layer of dimension 250 as hidden layers. Both the convolutional and the dense layers are followed by ReLU as nonlinearity, and Dropout (Srivastava et al., 2014) as regularization. Each review is padded/cut to 400 words. The CNN model achieves 90% accuracy on the test data, close to the state-of-the-art performance (around 94%). We would like to find out which k words make the most influence on the decision of the model in a specific review. The number of key words is fixed to be $k = 10$ for all the experiments.

The explainer of L2X is composed of a global component and a local component (See Figure 2 in Yang et al. (2018)). The input is initially fed into a common embedding layer followed by a convolutional layer with 100 filters. Then the local component processes the common output using two convolutional layers with 50 filters, and the global component processes the common output using a max-pooling layer followed by a 100-dimensional dense layer. Then we concatenate the global and local outputs corresponding to each feature, and process them through one convolutional layer with 50 filters, followed by a Dropout layer (Srivastava et al., 2014). Finally a convolutional network with kernel size 1 is used to yield the output. All previous convolutional layers are of kernel size 3, and ReLU is used as nonlinearity. The variational family is composed of an word embedding layer of the same size, followed by an average pooling and a 250-dimensional dense layer. Each entry of the output vector V from the explainer is multiplied with the embedding of the respective word in the variational family. We use both automatic metrics and human annotators to validate the effectiveness of L2X.

Post-hoc accuracy. We introduce *post-hoc accuracy* for quantitatively validating the effectiveness of our method. Each model explainer outputs a subset of features X_S for each specific sample X. We use $\mathbb{P}_m(y \mid \tilde{X}_S)$ to approximate $\mathbb{P}_m(y \mid X_S)$. That is, we feed in the sample X to the model with unselected words masked by zero paddings. Then we compute the accuracy of using $\mathbb{P}_m(y \mid \tilde{X}_S)$ to predict samples in the test data set labeled by $\mathbb{P}_m(y \mid X)$, which we call *post-hoc accuracy* as it is computed after instancewise feature selection.

Human accuracy. When designing human experiments, we assume that the key words convey an attitude toward a movie, and can thus be used by a human to infer the review sentiment. This assumption has been partially validated given the aligned outcomes provided by post-hoc accuracy and by human judges, because the alignment implies the consistency between the sentiment judgement based on selected words from the original model and that from humans. Based on this assumption, we ask humans on Amazon Mechanical Turk (AMT) to infer the sentiment of a review given the ten key words selected by each explainer. The words adjacent to each other, like "not good at all," keep their adjacency on the AMT interface if they are selected simultaneously. The reviews from different explainers have been mixed randomly, and the final sentiment of each review is averaged over the results of multiple human annotators. We measure whether the labels from human based on selected words align with the labels provided by the model, in terms of the average accuracy over 500 reviews in the test data set. Some reviews are labeled as "neutral" based on selected words, which is because the selected key words do not contain sentiment, or the selected key words contain comparable numbers of positive and negative words. Thus these reviews are neither put in the positive nor in the negative class when we compute accuracy. We call this metric *human accuracy*.

The result is reported in Table 4. We observe that the model

prediction based on only ten words selected by L2X align with the original prediction for over 90% of the data. The human judgement given ten words also aligns with the model prediction for 84.4% of the data. The human accuracy is even higher than that based on the original review, which is 83.3% (Yang et al., 2018). This indicates the selected words by L2X can serve as key words for human to understand the model behavior. Table 2 shows the results of our model on four examples.

4.2.2. Explaining hierarchical LSTM

Another competitive class of models in sentiment analysis uses hierarchical LSTM (Hochreiter & Schmidhuber, 1997; Li et al., 2015). We build a simple hierarchical LSTM by putting one layer of LSTM on top of word embeddings, which yields a representation vector for each sentence, and then using another LSTM to encoder all sentence vectors. The output representation vector by the second LSTM is passed to the class distribution via a linear layer. Both the two LSTMs and the word embedding are of dimension 100. The word embedding is pretrained on a large corpus (Mikolov et al., 2013). Each review is padded to contain 15 sentences. The hierarchical LSTM model gets around 90% accuracy on the test data. We take each sentence as a single feature group, and study which sentence is the most important in each review for the model.

The explainer of L2X is composed of a 100-dimensional word embedding followed by a convolutional layer and a max pooling layer to encode each sentence. The encoded sentence vectors are fed through three convolutional layers and a dense layer to get sampling weights for each sentence. The variational family also encodes each sentence with a convolutional layer and a max pooling layer. The encoding vectors are weighted by the output of the subset sampler, and passed through an average pooling layer and a dense layer to the class probability. All convolutional layers are of filter size 150 and kernel size 3. In this setting, L2X can be interpreted as a hard attention model (Xu et al., 2015) that employs the Gumbel-softmax trick.

Comparison is carried out with the same metrics. For *human accuracy*, one selected sentence for each review is shown to human annotators. The other experimental setups are kept the same as above. We observe that post-hoc accuracy reaches 84.4% with one sentence selected by L2X, and human judgements using one sentence align with the original model prediction for 77.4% of data. Table 3 shows the explanations from our model on four examples.

4.3. MNIST

The MNIST data set contains 28×28 images of handwritten digits (LeCun et al., 1998). We form a subset of the MNIST data set by choosing images of digits 3 and 8, with $11,982$ images for training and $1,984$ images for testing. Then we train a simple neural network for binary classification over the subset, which achieves accuracy 99.7% on the test data set. The neural network is composed of two convolutional layers of kernel size 5 and a dense linear layer at last. The two convolutional layers contains 8 and 16 filters respectively, and both are followed by a max pooling layer of pool size 2. We try to explain each sample image with $k = 4$ image patches on the neural network model, where each patch contains 4×4 pixels, obtained by dividing each 28×28 image into 7×7 patches. We use patches instead of raw pixels as features for better visualization.

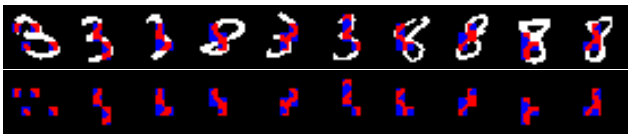

Figure 4. The above figure shows ten randomly selected figures of 3 and 8 in the validation set. The first line include the original digits while the second line does not. The selected patches are colored with red if the pixel is activated (white) and blue otherwise.

	IMDB-Word	IMDB-Sent	MNIST
Post-hoc accuracy	0.90.8	0.849	0.958
Human accuracy	0.844	0.774	NA

Table 4. Post-hoc accuracy and human accuracy of L2X on three models: a word-based CNN model on IMDB, a hierarchical LSTM model on IMDB, and a CNN model on MNIST.

We parametrize the explainer and the variational family with three-layer and two-layer convolutional networks respectively, with max pooling added after each hidden layer. The 7×7 vector sampled from the explainer is upsampled (with repetition) to size 28×28 and multiplied with the input raw pixels.

We use only the post-hoc accuracy for experiment, with results shown in Table 4. The predictions based on 4 patches selected by L2X out of 49 align with those from original images for 95.8% of data. Randomly selected examples with explanations are shown in Figure 4. We observe that L2X captures most of the informative patches, in particular those containing patterns that can distinguish 3 and 8.

5. Conclusion

We have proposed a framework for instancewise feature selection via mutual information, and a method L2X which seeks a variational approximation of the mutual information, and makes use of a Gumbel-softmax relaxation of discrete subset sampling during training. To our best knowledge, L2X is the first method to realize real-time interpretation of a black-box model. We have shown the efficiency and the capacity of L2X for instancewise feature selection on both synthetic and real data sets.

Acknowledgements

L.S. was also supported in part by NSF IIS-1218749, NIH BIGDATA 1R01GM108341, NSF CAREER IIS-1350983, NSF IIS-1639792 EAGER, NSF CNS-1704701, ONR N00014-15-1-2340, Intel ISTC, NVIDIA and Amazon AWS. We thank Nilesh Tripuraneni for comments about the Gumbel trick.

References

Ba, J., Mnih, V., and Kavukcuoglu, K. Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755*, 2014.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and MÃžller, K.-R. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv e-prints*, abs/1409.0473, September 2014.

Chen, J., Stern, M., Wainwright, M. J., and Jordan, M. I. Kernel feature selection via conditional covariance minimization. In *Advances in Neural Information Processing Systems 30*, pp. 6949–6958. 2017.

Chollet, F. et al. Keras. https://github.com/keras-team/keras, 2015.

Cover, T. M. and Thomas, J. A. *Elements of information theory*. John Wiley & Sons, 2012.

Gao, S., Ver Steeg, G., and Galstyan, A. Variational information maximization for feature selection. In *Advances in Neural Information Processing Systems*, pp. 487–495, 2016.

Guyon, I. and Elisseeff, A. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *stat*, 1050:1, 2017.

Kim, Y. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

Kindermans, P.-J., Schütt, K., Müller, K.-R., and Dähne, S. Investigating the influence of noise and distractors on the interpretation of neural networks. *arXiv preprint arXiv:1611.07270*, 2016.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, J., Luong, M.-T., and Jurafsky, D. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057*, 2015.

Lipton, Z. C. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. pp. 4768–4777, 2017.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 142–150. Association for Computational Linguistics, 2011.

Maddison, C. J., Tarlow, D., and Minka, T. A* sampling. In *Advances in Neural Information Processing Systems*, pp. 3086–3094, 2014.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013.

Peng, H., Long, F., and Ding, C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

Raffel, C., Luong, T., Liu, P. J., Weiss, R. J., and Eck, D. Online and linear-time attention by enforcing monotonic alignments. *arXiv preprint arXiv:1704.00784*, 2017.

Ribeiro, M. T., Singh, S., and Guestrin, C. 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*, pp. 1135–1144. ACM, 2016.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. In *ICML*, volume 70 of *Proceedings of Machine*

Learning Research, pp. 3145–3153. PMLR, 06–11 Aug 2017.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pp. 3319–3328, 2017.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pp. 2048–2057, 2015.

Yang, P., Chen, J., Hsieh, C.-J., Wang, J.-L., and Jordan, M. I. Greedy attack and gumbel attack: Generating adversarial examples for discrete data. *arXiv preprint arXiv:1805.12316*, 2018.

Zhang, Y. and Wallace, B. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*, 2015.