# LIME METHOD TUTORIAL

**Paloma Almodovar and Tin Dizdarevic**

## 1. INTRODUCTION

A concept that is at first counterintuitive, but essential in the machine learning environment, is that a prediction being correct doesn't mean that you have found a good model.
If the "train of thought" of your model is incorrect, even if its predictions happen to be accurate, you won't consider it reliable. This lack of trust will prevent you from using it in the future.

A good model needs to perform well on real-world data not only as a whole but also in its individual predictions (e.g. medical diagnosis, terrorism detection…). This is why, understanding the reasons behind each decision being made is essential.

Because of this, an efficient method to convince someone that a model is good is the following. Start by providing explanations for individual predictions (in order to build trust in them). After we are confident in a large enough sample of predictions (and their explanations) we can say we trust the model as a whole and feel comfortable using it.

But how can we effectively do the above? This is where explainers like the Local Interpretable Model-Agnostic Explanations (LIME) come into play. LIME is "an algorithm that can explain the predictions of any classifier or regressor in a faithful way, by approximating it locally with an interpretable model". In the same vein we find SP-LIME (where SP stands for Submodular Pick) "a method that selects a set of representative instances with explanations to address the "trusting the model" problem, via submodular optimization".

## 2. EXPLANATIONS AND EXPLAINERS

It is important to specify that when we said "explaining a prediction" in the introduction, we referred to presenting textual or visual aids that provide qualitative understanding of the relationship between the instance's components (e.g. words in text, patches in an image) and the model's prediction.

For example, if a good model predicts that someone has coronavirus, an explainer highlights the symptoms in the patient's record that led to the prediction: "coughing" and "loss of taste" are portrayed as contributing to the "coronavirus" prediction, while "no fatigue" is evidence against it. This intelligible reasoning helps the health care professionals make an informed decision about whether to trust the model or not. It also enables them to compare different models in order to choose the best one.

Explainers are also extremely useful to identify data leakage (ex. patient ID contributing to the above prediction) and the need to change the training data to avoid dataset shift (where training data is different than test data).

Among the desired characteristics for explainers we find the following:
- They must be interpretable (easy to understand, take into account the user's limitations), regardless of the features used by the model.
- They must provide qualitative understanding between the input variables and the response.
- They should be able to explain any model (model-agnostic).
- They should "practice" local fidelity. For an explanation to be meaningful, it must correspond to how the model behaves in the vicinity of the instance being predicted.

## 3. LOCAL INTERPRETABLE MODEL-AGNOSTIC EXPLANATIONS (LIME)

The overall goal of LIME is to identify an interpretable model over the interpretable representation that is locally faithful to the classifier.

### 3.1 INTERPRETABLE DATA REPRESENTATIONS:

We denote $x \in \mathbb{R}^d$ be the original representation of an instance being explained.
We then use $x' \in \{0, 1\}^{d'}$ to denote a binary vector for its interpretable representation, the one that is easy to understand and takes into account the user's limitations.

### 3.2 FIDELITY-INTERPRETABILITY TRADE-OFF

We define an explanation as a model $g \in G$, where G is a class of potentially interpretable models (such as linear models, decision trees, falling rule lists…). Therefore, a model $g \in G$ can be presented to the user with visual or textual artifacts that make it easier to understand.

The domain of g is $\{0, 1\}^{d'}$, which means that g acts over the presence or absence of the interpretable components.

However, in real life cases, we can't expect every $g \in G$ to be simple enough to be interpretable. Because of this, we use a measure of complexity of the explanation $g \in G$, which is denoted by $\Omega(g)$. For example, for linear models $\Omega(g)$ can be the number of non-zero weights.

Let the model being explained be denoted $f : \mathbb{R}^d \rightarrow \mathbb{R}$.
- In classification, f(x) is the probability (or a binary indicator) that x belongs to a certain class (category) .
- In order to define locality around x, we need a proximity measure between an instance z and x. We denote this measure using $\pi_x(z)$.
- We then let $L(f, g, \pi_x)$ be a measure of how unfaithful g is in approximating f in the locality defined by $\pi_x$.

Now that we have defined everything, our goal is to minimize $L(f, g, \pi_x)$ as much as possible while having $\Omega(g)$ be low enough for the user to interpret. By doing this we achieve interpretability and local fidelity.

This is why the explanation produced by LIME is $\xi(x) = \underset{g \in G}{\text{argmin}}\ L(f, g, \pi_x) + \Omega(g)$

This equation is very versatile since it can be used with different explanation families G, fidelity functions L, and complexity measures Ω. In this tutorial we will focus on sparse linear models as explanations and on performing the search using perturbations.

### 3.3 SAMPLING FOR LOCAL EXPLORATION:

As mentioned in the previous section, one of our goals is to minimize the locality-aware loss $L(f, g, \pi_x)$ . We have to make sure that in the process we don't make any assumptions about f, since we want the explainer to be model agnostic.

Because of this, in order to learn the local behavior of f as the interpretable inputs vary, we approximate $L(f, g, \pi_x)$ by drawing samples, weighted by $\pi_x$.
- We sample instances around x' by drawing nonzero elements of x' uniformly at random.
- Given a perturbed sample $z' \in \{0, 1\}^{d'}$ (which contains a fraction of the nonzero elements of x'), we recover the sample in the original representation $z \in \mathrm{R}^d$ and obtain f(z), which is used as a label for the explanation model.
- Given this dataset Z of perturbed samples with the associated labels, we optimize the LIME equation to get an explanation ξ(x).

The benefit of doing this is that, even though the original model may be too complex to explain globally, LIME presents an explanation that is locally faithful, where the locality is captured by $\pi_x$. An additional advantage is that this method is quite robust to sampling noise the samples are weighted by $\pi_x$.

### 3.4 SPARSE LINEAR EXPLANATIONS

For the rest of the tutorial, we let G be the class of linear models, such that $g(z') = w_g \cdot z'$ and we use the locally weighted square loss as $L(f, g, \pi_x) = \sum_{z, z' \in Z} \pi_x(z)\,(f(z) - g(z'))^2$ , where we let $\pi_x(z) = \exp(-D(x, z)^2/\sigma^2)$ be an exponential kernel defined on some distance function D with width σ.

In regard to text classification, we make sure that the explanation is understandable by letting the interpretable representation be a set of up to K words ( $\Omega(g) = \infty 1[||w_g||_0 > K]$ ). K can be changed depending on the instance or remain constant.

We use the same Ω for image classification, using "super-pixels" instead of words: the interpretable representation of an image is a binary vector where 1 indicates the original super-pixel and 0 indicates a grayed out super-pixel.

However, this particular choice of Ω makes directly solving the LIME equation intractable. Because of this, we approximate it by selecting K features using LASSO and then learning the weights via least squares.

## 4. SUBMODULAR PICK FOR EXPLAINING MODELS

Like we mentioned at the beginning of this tutorial, although an explanation of a single prediction provides some understanding into the reliability of a classifier, it is not enough to determine and establish trust in the model as a whole. Because of this, we instead look to give a global understanding of the model by explaining a set of individual instances.

In order for the explanations of multiple instances to be truly insightful, they need to be selected meticulously, because users probably won't have patience or time to examine a large number of explanations. We establish a budget B that denotes the maximum number of explanations that someone can or is willing to look at in order to understand a model.

Therefore, given a set of instances X, the pick step is the task of selecting B instances for the user to inspect. This step should also take into account the explanations that accompany each prediction and make sure to pick a diverse and representative set of explanations to show the user how the model behaves globally.

Given the explanations for a set of instances X (|X| = n), we construct an n × d' explanation matrix W that represents the local importance of the interpretable components for each instance. When using linear models as explanations, for an instance $x_i$ and explanation $g_i = \xi(x_i)$, we set $W_{ij} = |w_{gij}|$. Further, for each component (column) j in W, we let $I_j$ denote the global importance of that component in the explanation space. Intuitively, we want I such that features that explain many different instances have higher importance scores.
For the text applications, we set $I_j = \sqrt{\sum_{i=1}^{n} Wij}$.
For images, I must measure something that is comparable across the super-pixels in different images (ex. color histograms).

To make sure that the set of explanations showed to the users is non redundant we define coverage as the set function c(V ,W, I) that, given W and I, computes the total importance of the features that appear in at least one instance in a set V: $c(V, W, I) = \sum_{j=1}^{d'} 1[\exists\, i \in V : Wij > 0]\ Ij$

To find that the ideal set V (the one that achieves the highest coverage we perform
$\text{Pick}(W, I) = \underset{V, |V| <= B}{\text{argmax}}\ c(V, W, I)$
The problem with the above equation is the difficulty of maximizing a weighted coverage function. Because of this we let $c(V \cup \{i\}, W, I) - c(V, W, I)$ be the marginal coverage gain of adding an instance i to a set V, since due to submodularity, a greedy algorithm that iteratively adds the instance with the highest marginal coverage gain to the solution offers a constant-factor approximation guarantee of $1-1/e$ to the optimum.

After doing the steps described above, we will obtain the ideal set V which we call submodular pick.

Now we have all that is necessary to determine if we should trust a model according to the explanations it gives as to why it made a specific set of predictions.