Dependable AI Assignment-1 Report

LIME

(Local Interpretable Model-Agnostic Explanations)

Aditya Rathor
B22AI044

Vishesh Sachdeva
B22AI050

Table of Contents

Table of Contents 1
Pipeline for LIME 2
Implementation 3
Model explainability using LIME 4
References 6

Pipeline for LIME

---

1. **Feature Extraction** - The English annotation descriptions of both train and test images are converted into 384-dimensional vectors. This transformation is performed using the "all-MiniLM-L6-v2" sentence embedding model.
2. **Train the AnnexML model** - The AnnexML model is trained on these extracted features.
3. **Interpretable Data Representation** - This involves a binary mapping that determines which words from the original annotation descriptions are retained.

$$x \in \mathbb{R}^d \rightarrow x' \in \{0,1\}^{d'}$$

Original Representation — Interpretable Representation

4. **Perturbation Generation** - Randomly select a subset of words from the original description.

$$x \rightarrow x' \rightarrow z' \rightarrow z \rightarrow f(z)$$

interpretable representation — sampling local area — inverse (interpretable representation) — obtain label

5. **Weighting of perturbed samples** - To ensure that the explainable model focuses on relevant samples, higher weights are given to perturbed samples that closely resemble the original instance.

$$\pi_x(z) = exp\left(-\frac{D(x,z)^2}{\sigma^2}\right)$$

6. **Inverse Interpretable Representation** - This involves constructing a sentence using only the selected words and then converting it into a 384-dimensional vector using the "all-MiniLM-L6-v2" sentence embedding model for each perturbed sample.
7. **Training the explainable model** - Fit an interpretable model (e.g., linear regression) on the perturbed samples to approximate the decision boundary of the black-box model locally separately for each predicted label for an instance. The selected features (binary word presence) serve as inputs, while the label's prediction scores guide the fitting process.

measure of unfaithfulness — interpretable model — complexity measure

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \; \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

an instance — model to be explained — proximity measure to define locality

$$g(z') = w_g \cdot z'$$

$$\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in Z} \pi_x(z)\left(f(z) - g(z')\right)^2$$

$$\Omega(g) = \infty \, \mathbb{I}[\| w_g \|_0 > K]$$

8. **Feature Attributions** - The explainable model's coefficients are used to derive feature attribution scores. These scores indicate the importance of individual words in influencing the black-box model's prediction for the selected instance.

Implementation

---

1. **Preparing Test data**
   - get_desc() returns the image description present in the English annotations file.
   - get_test_data() returns the true labels and the text descriptions for the first explainrows test samples.
   - get_sentence_embedding() returns the 384-dimensional sentence embedding using the "all-MiniLM-L6-v2" pretrained model, given a sentence as input.
   - get_test_files() generates test file and text test file, given the labels and text description for each instance to be explained.

2. **Perturbation Generation**
   - A total of samples_per_row perturbed samples are generated randomly for each instance to be explained.
   - This is done by randomly selecting a subset of words. Each word has a probability of pick_probab to be chosen for the perturbed sample.
   - generate_perturbed_files() generates perturbed interpret file (stores the binary mapping of each perturbed sample) and perturbed test file (stores the sentence embedding of inverse interpretable representation of each perturbed sample)

3. **Explainable model (g) - Linear Regression model**
   - w0 - bias term for linear regression model
     w1 - weights corresponding to each word/token for linear regression model
   - get_distances() method returns the array of weights of perturbed samples ($\pi_x(z) = \exp(-D(x,z)/\sigma^2)$), where D(x,z) is the number of words removed in the perturbed sample and σ=4.
   - MSE() method returns the mean square error loss ($L(f,g,\pi_x)$).
   - fit() method trains the model using gradient descent and returns the bias and weights.
   - K-support Lasso regularisation - Only top K significant weights(i.e., those with the largest absolute weights) are retained during each iteration of gradient descent.

Model explainability using LIME

Explaining 3 predicted labels for row 1:

- Explaining for label **building**:
  Iterations: 7109 , Loss: 589.7362235129539
  [('building', 160.01518347262345), ('palm', 50.12883917206696), ('cars', 21.423414619216587), ('large', 20.910471451147508), ('white', 18.5440195886508), ('red', 5.994898574998625), ('centre', 1.54881194304699), ('park', 1.048492921505037), ('a', 0.0), ('on', 0.0), ('the', 0.0), ('left', 0.0), ('tree', 0.0), ('in', 0.0), ('of', 0.0), ('picture', 0.0), ('mostly', 0.0), ('street', 0.0), ('at', 0.0), ('junction', 0.0), ('some', 0.0), ('them', 0.0), ('turning', 0.0), ('others', 0.0), ('going', 0.0), ('straight', 0.0), ('there', 0.0), ('are', 0.0), ('umbrellas', 0.0), ('right', 0.0), ('people', 0.0), ('walking', 0.0), ('through', 0.0), ('crossing', 0.0), ('road', 0.0), ('foreground', 0.0)]

  a large building on the left a palm tree in centre of picture mostly white cars in the street at a junction some of them turning left others going straight there are red umbrellas in a park on the right people are walking through the park others are crossing the road in the foreground

- Explaining for label **car**:
  Iterations: 10000 , Loss: 419.6898434730393
  [('cars', 162.73276606086714), ('street', 49.00956227722788), ('building', 24.066511908688614), ('turning', 13.450673512974435), ('road', 3.082599433811451), ('at', 2.618981593150313), ('a', 0.0), ('large', 0.0), ('on', 0.0), ('the', 0.0), ('left', 0.0), ('palm', 0.0), ('in', 0.0), ('centre', 0.0), ('of', 0.0), ('picture', 0.0), ('white', 0.0), ('junction', 0.0), ('some', 0.0), ('them', 0.0), ('others', 0.0), ('going', 0.0), ('straight', 0.0), ('there', 0.0), ('are', 0.0), ('red', 0.0), ('umbrellas', 0.0), ('park', 0.0), ('right', 0.0), ('people', 0.0), ('walking', 0.0), ('through', 0.0), ('crossing', 0.0), ('foreground', 0.0), ('mostly', -38.33141124698154), ('tree', -88.94749571934851)]

  a large building on the left a palm tree in centre of picture mostly white cars in the street at a junction some of them turning left others going straight there are red umbrellas in a park on the right people are walking through the park others are crossing the road in the foreground

- Explaining for label **centre**:
  Iterations: 3347 , Loss: 19.266030258045443
  [('a', 0.0), ('large', 0.0), ('on', 0.0), ('the', 0.0), ('left', 0.0), ('palm', 0.0), ('centre', 0.0), ('of', 0.0), ('picture', 0.0), ('mostly', 0.0), ('white', 0.0), ('cars', 0.0), ('street', 0.0), ('at', 0.0), ('junction', 0.0), ('some', 0.0), ('them', 0.0), ('turning', 0.0), ('straight', 0.0), ('there', 0.0), ('are', 0.0), ('red', 0.0), ('umbrellas', 0.0), ('park', 0.0), ('right', 0.0), ('people', 0.0), ('crossing', 0.0), ('road', 0.0), ('foreground', 0.0), ('walking', -2.522736883213293), ('in', -2.6335884414676514), ('through', -2.7075890532401146), ('others', -2.7910182245823583), ('going', -2.8612651533410447), ('building', -4.791817813318154), ('tree', -6.024978891568655)]

  a large building on the left a palm tree in centre of picture mostly white cars in the street at a junction some of them turning left others going straight there are red umbrellas in a park on the right people are walking through the park others are crossing the road in the foreground

Observations & Analysis

- **Strong feature association for certain labels** - The attributions/Linear regression coefficients for the tokens same as the predicted labels is clearly high. These words can be seen with dark blue color.

  E.g. - “building:160.0” dominates “Building”, “car:162.7” dominates “car”

- **Contextual Cues Matter** - We can see significant positive attribution for words (in light blue colour) that share some contextual relationship with the predicted label. This indicates that the predictions does not rely solely on a single keyword but also integrates supporting terms.

  E.g. - “street:49.0”, “building:24.0”, “turning:13.45”, “road:3.08” for label “car”

- **Neutral or Irrelevant words are ignored** - Many words have zero attribution, meaning they do not influence the prediction. These are likely common words (e.g., "the," "a",”on”) or words that do not add relevant information for the label. The model effectively filters out noise.
- **Negative Attributions Reveal Conflicting Features** - Some words get slight negative attributions(in light red color) which shows that presence of some words makes presence of predicted labels in the test image less likely.

  E.g. - “tree:-88.94” for label “car”

- **Weak or Unclear Concepts Struggle to Emerge** - There exists some predicted labels which do not have any significant attributions, meaning the model struggles to find defining features for it.

  E.g. - The "centre" label has low attribution across all words, suggesting that the model might not have a strong understanding of what constitutes a "centre" in the dataset. This might be due to potential AnnexML model biases or Sentence embedding model biases.

References

- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). ***"Why Should I Trust You?": Explaining the Predictions of Any Classifier***. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, 1135–1144. https://doi.org/10.1145/2939672.2939778

- https://github.com/emanuel-metzenthin/Lime-For-Time - LIME implementation for time-series predictor model.

- https://drive.google.com/file/d/1jgT2N0TlBWmuoze-thvtQh_URBYdFAB-/view?usp=sharing - Model explainability using LIME lecture slides - CSL 7370 Dependable AI course (Jan 2025) - Dr. Yashaswi Verma, IIT Jodhpur

- https://www.geeksforgeeks.org/linear-regression-implementation-from-scratch-using-python - Linear Regression from scratch

- https://stackoverflow.com/questions/23271575/printing-bold-colored-etc-text-in-ipython-qtconsole - printing text with colored background

- Also discused with our course instructor Dr. Yashaswi Verma and course TA Rudra Dutt