AIPI 590 - Human-AI Interaction

Neha L Senthil

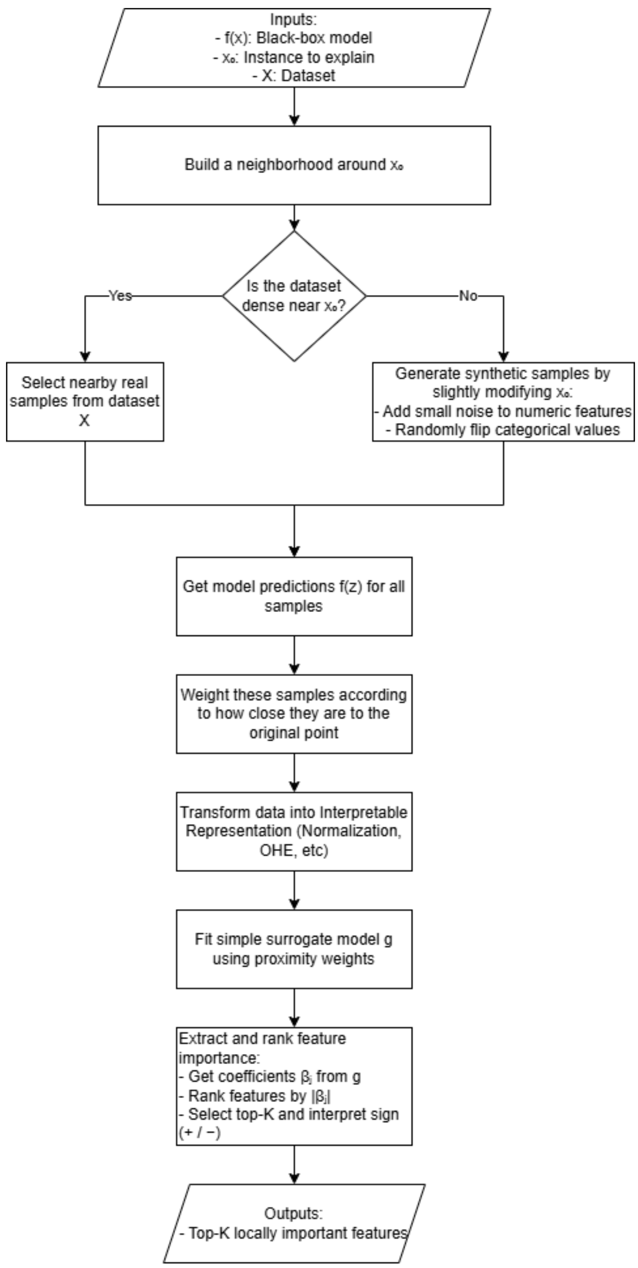
October 20, 2025

LIME: Local Interpretable Model-Agnostic Explanations (Summary)

LIME is a technique to explain the predictions of complex, black-box machine learning models. Instead of trying to explain the entire model, LIME focuses on explaining a specific prediction for a single data point. The key idea is to approximate the black-box model locally (around the chosen instance) with a simple, interpretable model.

This process provides human-understandable insights into which features played the biggest role in the model's decision for that instance. LIME does not depend on the underlying model, making it a model-agnostic explanation approach.

Flowchart



Algorithm

Goal: Explain how a complex (black-box) model makes a specific prediction for one instance x_0 by learning a simple, interpretable model around it. This is more in depth compared to the flow chart, aimed more for developers and people looking at this algorithm from a technical perspective.

Inputs:

- $f(x)$: trained black-box model
- x_0: instance to explain
- X: background dataset
- N: number of neighborhood samples to generate (e.g., 5000)
- σ: kernel width for measuring locality
- g: simple surrogate model (e.g., linear or decision tree)
- K: number of top features to display

Outputs:

- Top-K features that most strongly influenced the model near x_0
- A local fidelity score showing how well the explanation matches the original model

Algorithm Steps:

1. **Select an instance.** Choose the particular data point x_0 whose prediction you want to explain.
2. **Build a neighborhood around x_0.** Construct a small region around the chosen instance x_0 to study how the model behaves for similar data points.
   - **(2a) If sufficient nearby data points already exist:** Select those samples from the original dataset X that lie close to x_0 based on a distance measure (e.g., Euclidean distance). These points form the local neighborhood.
   - **(2b) If the dataset is sparse around x_0:** Generate additional *synthetic samples* by making small, controlled changes to x_0. For example:
     - Add slight Gaussian noise to numeric features to create realistic variations.
     - Randomly flip categorical values with a small probability while preserving valid feature combinations.

   This process creates a smooth 'cloud' of data around x_0, ensuring enough samples to approximate the model's local behavior.
3. **Get model predictions.** Pass each sample (real or synthetic based on output from 2) through the black-box model $f(x)$ and record the predicted value or class probability.
4. **Compute proximity weights.** Assign a weight w_i to each sample z_i based on its distance from x_0:

$$w_i = \exp\left(-\frac{d(z_i, x_0)^2}{\sigma^2}\right)$$

   Nearby samples have higher weights, ensuring the explanation focuses on the local region around x_0.
5. **Transform into an interpretable representation.** Prepare the data so that it can be understood both by the surrogate model and by humans:
   - **Machine side:** Normalize numeric features and one-hot encode categorical features, producing $Z^{(int)}$ for model training.
   - **Human side:** Keep a mapping so each encoded feature can later be converted back to a readable concept (e.g., *Gender: Male*, *Income: $75,000*).
6. **Fit the interpretable surrogate model.** Train a simple model g (e.g., sparse linear regression or small decision tree) on the interpretable data $Z^{(int)}$ using the proximity weights w_i. This surrogate mimics the behavior of $f(x)$ locally around x_0.
7. **Extract and rank feature importance.**
   - **(7a)** Obtain feature coefficients β_j from g.

- **(7b)** Rank features by their absolute weights $|\beta_j|$ to determine their influence strength.
- **(7c)** Select the top-K most important features and translate their encoded form back into human labels.
- **(7d)** Interpret each feature's sign:
  - Positive β_j: increasing this feature raises the model's predicted outcome.
  - Negative β_j: increasing this feature lowers the model's predicted outcome.

8. **Output the top features.** Present the top-K features in a readable table showing their names, values, weights, and qualitative interpretation. Also report a local fidelity measure (e.g., weighted R^2) to indicate how well g matches f near x_0.

Example Output Table:

Feature	**Value**	**Weight**	**Interpretation**
Credit Utilization	0.82	$+0.70$	Higher utilization increases risk
Income	\$75,000	-0.60	Higher income decreases risk
Debt Ratio	0.55	$+0.40$	Higher debt increases risk