Week 4 - Example-Based Interpretability

Wednesday, February 5, 2025 1:11 PM

Motivation

1. Reasoning by analogy to examples can be powerful. We'd like ML systems that support this type of reasoning.

2. These tools are especially useful when we want to avoid ML jargon. A general audience might not be able to make sense of an explanation of "CNN Filter 1432," but they *can* make sense of how the model behaves on examples.

3. Unlike feature attribution, there doesn't seem to be a consensus around a few central approaches (like SHAP). So in these notes, we'll review a few representative methods (examples!) from the literature.

Synthesizing Examples

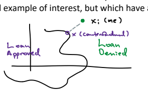
Counterfactuals

4. Surprisingly, some of the simplest approaches to example-based interpretability don't refer to examples in the training data. Instead, they _synthesize_ examples that help make sense of the original predictions. We'll focus on Wachter et al. (2018)'s "counterfactual " and Hendricks et al. (2016)'s "visual" explanations.

5. Their motivation comes from a close reading of GDPR. They realized that describing internal model mechanics wouldn't help average citizens with:

   - **Understanding** why a prediction was made.
   - **Contesting** the factors that led to a prediction.
   - **Adapting** to help achieve their goals ("algorithmic recourse")

   To resolve this, they defined counterfactual explanations as examples that "are as close as possible" to the real example of interest, but which have a different (desired) model prediction.



6. There are many ways to operationalize this definition. They suggest solving the optimization:
   For a target prediction y and a target sample x[i], solve:

$$X^* = \arg\min_{x}\max_{\lambda} \lambda\left(f_\theta(x) - y\right)^2 + d(x, x_i)$$

   For example, with a distance function:

$$d(x, x') = \sum_k \frac{1}{\underbrace{\nu_k}_{\text{Coord. K scaling factor}}} |x_k - x'_k|$$

7. Their illustrations are relatively simple (2-layer networks, few neurons), but their main arguments are conceptual. In this example (table 2 in the paper), they trained a model to predict GPA in the first year of law school based on characteristics from an admissions classifier. The left panel gives original examples, and those in the middle/right are counterfactuals with target scores y = 0. The change in score from the original example to the counterfactual is the far left column "score."

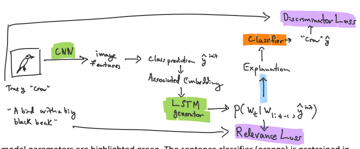
				Normalised L2					
score	GPA	LSAT	Race	GPA	LSAT	Race	GPA	LSAT	Race
0.17	3.1	39.0	0	3.0	37.0	0.2	3.0	34.0	0
0.54	3.7	48.0	0	3.5	39.5	0.4	3.5	33.1	0
-0.77	3.3	28.0	1	3.5	39.8	0.4	3.4	33.4	0
-0.83	2.4	28.5	1	2.7	37.4	0.2	2.6	35.7	0
-0.57	2.7	18.3	0	2.8	28.1	-0.4	2.9	34.1	0

Visual Explanations

8. Counterfactual examples look like the original data. Hendricks et al. (2016) instead synthesize text explanations to accompany the predictions on image data. For being the oldest paper we review this week, it feels the most prescient, both in its use of multimodal data and its focus on using language models to generate explanatory text.

9. The key challenge they emphasize is that description ≠ explanation. A description only needs to accurately summarize what's in an image. An explanation needs to highlight those features that are relevant to the model's class prediction.

10. Their architecture has this form:



   The model parameters are highlighted green. The sentence classifier (orange) is pretrained in advance. The blue edge involves sampling a new sentence. This complicates optimization, but is solved using the REINFORCE algorithm (a low sample-efficiency approach, but it seems to work).

11. The two losses (purple) serve complementary purposes:

   - Relevance Loss: Do the sampled sentences look like the original sentence description from the training data?
   - Discriminator Loss: Do the sampled sentences help predict the correct class?

   The second term is what makes this different from a caption model, say. It forces the generated sentences to mention features that help with classification.

Finding Examples

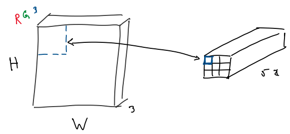
Parts-Based Explanation

12. Chen et al. (2019) try defining an intrinsically interpretable model that:

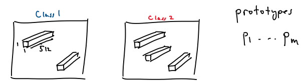
   - Defines representative image patches ("prototypes").
   - Uses similarity to those prototypes to guide classification.

   Since the prototypes can be matched with training data examples, this helps users identify which parts of which images led to the final classification.

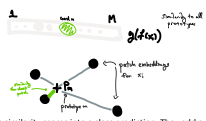
13. CNNs transform images into "long" cubes. The depth dimension can be interpreted as different learned image features. The spatial position in that cube (W x H) corresponds to a spatial patch in the original image. E.g., the top left corner in the cube corresponds represents the top left patch in the original image.



14. The idea is to create class-specific prototypes with the same depth as these representations and which are just 1 x 1 pixel tall/wide. You can imagine comparing each prototype to the features associated with a path.

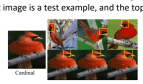


15. Specifically, the "long" summaries of each image patch can be compared to each prototype. Patches that are close to a prototype are more likely to come from the same class as that prototype. For prototype m, each find the most similar patch in x[i] and store that similarity score in coordinate m of the vector g(f(x[i])).



16. A final linear model maps those similarity scores into a class prediction. They add a sparsity penalty so that only a few prototypes are relevant for each class.

17. The prototypes, CNN weights, and final linear model coefficients are optimized to maximize classification accuracy. For any new example, we can look at which of the prototypes had the most influence on the final classification. These prototypes can be matched up with training data patches that give embeddings close to those prototypes. E.g., the left image is a test example, and the top row highlights relevant training patches.



Q: Can you come up with geometric interpretation of the "Clst" or "Sep" terms in objective on page 5? It should look similar to the drawing under point (15) above.

Prototypes and Criticism

18. Kim et al. (2016) try to extract relevant examples without reference to any specific model architecture. They identify prototypes (now meaning "typical examples") and criticism (unusual examples) to help people understand a dataset. In principle they could help clarify differences in model behavior on these selected examples. However, their human subjects study simply considers whether participants can identify the correct class (not necessarily any model's prediction).

19. We could use any clustering method to find prototypes and any anomaly detection method to find criticisms. They discuss how to use Maximum Mean Discrepancy (MMD) for both tasks.

$$MMD(\mathcal{F}, \mathbb{P}, \mathbb{Q}) = \sup_{f \in \mathcal{F}} \mathbb{E}_{\mathbb{P}}[f(x)] - \mathbb{E}_{\mathbb{Q}}[f(y)]$$

Q. If F = {x, x^2}, then for which pairs of P and Q is the MMD 0?

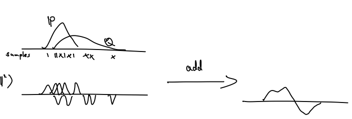
20. To contrast two distributions, we can use a witness function,

$$f(x) = \mathbb{E}_{\mathbb{P}}[k(x, \overset{\text{Random}\sim P}{x'})] - \mathbb{E}_{\mathbb{Q}}[k(x, \overset{\text{Random}\sim Q}{x'})]$$

   Imagine drawing N samples from both P and Q. Then we can approximate the witness function.

$$f(x) = \frac{1}{N}\left[\sum_{x'_i \sim \mathbb{P}} K(x, x'_i) - \sum_{x'_i \sim \mathbb{Q}} K(x, x'_i)\right]$$

$$K(x, x') = \exp(-\gamma \|x - x'\|^2)$$



21. Let P be the empirical distribution over the entire training set. Let Q be the empirical distribution over a selected subset S. To find prototypes, we look for a subset S so that P and Q have small MMD distance from one another. Note that the subset S that minimizes the MMD also maximizes

$$J(S) = \underbrace{\frac{1}{n^2}\sum_{i,j} K(x_i, x_j)}_{\text{constant wrt } S} - MMD^2(\mathcal{F}, \mathbb{P}_{1:N}, \mathbb{Q}_S)$$

   It turns out that this is a submodular function, which means greedy heuristics can be used to solve the problem efficiently.