Evaluating Contrastive Explanations

Project Plan

52093275

Hariss Ali Gills

h.gills.20@abdn.ac.uk

Department of Computing Science,
University of Aberdeen, Aberdeen AB24 3UE, UK

Introduction

Machine learning (ML) models are increasingly utilized to support major decisions in various industries [1]. These models may be Black Box Models which cannot be understood by looking at their parameters (e.g. a neural network). However, the field of Explainable Artificial Intelligence (XAI) attempts to address this issue by understanding models and their predictions to promote efficient debugging of models and to achieve a higher degree of user satisfaction and trust [2]. One such method is through Counterfactuals (CF) explanations which suggest what should be different in the input instance to flip the outcome. This in turn, leads to a Contrastive explanation, focusing on the differences in features that led to the different outcome.

CF explanations are fundamental in closing the gap between the users' mental model and the ML prediction [3]. For example, a customer of a bank using a ML-powered loan approval system that has been rejected can take the necessary steps to take action. Likewise, a bank would want to know its model is in line with financial regulation. Miller et al's work also concluded that good explanations, in addition to being contrastive, are selected in a biased manner and that they are socially aligned [3]. A survey on the latest CF methods by Guidotti et al. presented a taxonomy of CF explainers and quantitatively benchmarked these explainers on several metrics [4].

This dissertation aims to use a number of the metrics defined in the survey to statistically compare the following methods:

- Diverse Counterfactual Explanations (DICE) is an optimization-based CF method that ensures feasibility and diversity while finding counterfactuals [5]. It was found to be a top performer on most of the metrics in the survey [4].
- Artificial Immune Diverse Explanations (AIDE) is another optimization-based CF method which uses the Immune System as a metaphor to find the counterfactuals [6]. This method was adapted from the opt-aiNet algorithm [7] inspired by the

clonal selection theory of acquired immunity.

Goals

As mentioned in the introduction, the goal of the paper is to conduct an experiment to statistically compare DICE and AIDE. Specifically, based on the following metrics chosen based on their importance:

- Size: Measures the proportion of counterfactuals generated relative to the maximum requested, emphasizing the ability to produce multiple explanations when applicable. Higher values are better.
- Dissimilarity: Evaluates the proximity between the original instance and the counterfactuals, considering both feature-level sparsity and overall distance. Lower values are better.
- Runtime: measures the efficiency based on the elapsed time needed for the explainer to generate counterfactuals. A shorter runtime indicates better performance.
- Actionability: Measures the proportion of counterfactuals that can be practically implemented based on actionable features; higher values indicate better actionability.
- Diversity: Evaluates the variety among counterfactuals in terms of both feature differences and overall distance, with greater diversity being preferable.

Additional Goals

The experiment can be expanded to consider the following:

1. Measure metrics like Instability, Implausibility, and Discriminative Power.
2. Develop a version of AIDE that is optimized for a certain metric.
3. Compare counterfactuals from different domains to check if counterfactuals perform better in some domains than others.
4. Compare counterfactuals from different Black Box Models to see if counterfactuals perform better in specific models.

Methodology

In addition to iteratively working on the project report, the following steps will be taken to achieve the goals:

- Reading the background work and understanding DICE, AIDE, and the experimental design of the survey.

- Preprocess the `adult`, `compas`, `fico`, and `german` datasets since they are common in XAI literature.
- Train and tune the hyperparameters of Random Forest and Deep Neural Network classifiers for each dataset.
- Statistically evaluate and plot the results of the measured metrics.

Resources Required

To develop the project, the following resources are required:

- A machine with a web browser, an internet connection, and a text editor.
- `Python 3.11+`
- The `dice_ml` package and its dependencies like `sklearn`, `keras`, `tensorflow`.

Risk Assessment

Time constraints represent a significant risk factor that may impede the completion of the project. To address this challenge, the number of metrics, model types, and datasets will be reduced. Such adjustments will allow for a more focused and manageable approach. Since a majority of the datasets cover socially sensitive tasks, consulting with experts from relevant domains valuable insights and ensure that the counterfactuals align with professional and ethical standards.

Timeline

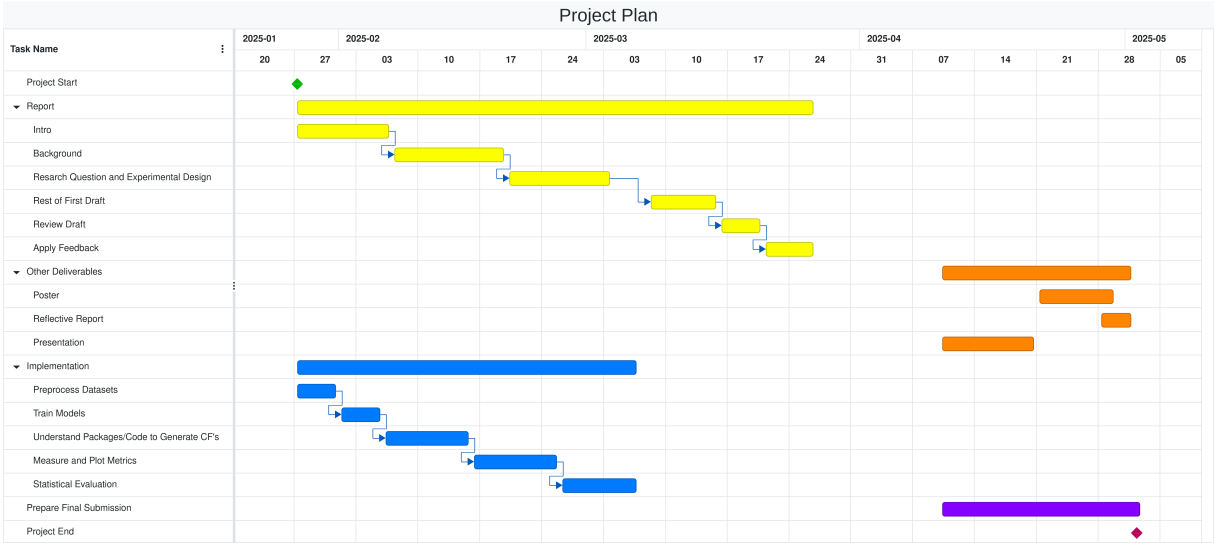

Figure 1: Project Gantt Chart

References

[1] Sheena Angra and Sachin Ahuja. Machine learning and its applications: A review. In *2017 international conference on big data analytics and computational intelligence (ICBDAC)*, pages 57–60. IEEE, 2017.

[2] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[3] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

[4] Riccardo Guidotti. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*, 38(5):2770–2824, 2024.

[5] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 607–617, 2020.

[6] James Forrest, Somayajulu Sripada, Wei Pang, and George M Coghill. Are contrastive explanations useful? 2021.

[7] Jason Brownlee. *Clever algorithms: nature-inspired programming recipes*. Jason Brownlee, 2011.