Counterfactual Explanations for Rankings

Mozhgan Salimiparsa†,*
† Department of Computer Science, Western University

Abstract

Machine learning models have the potential to transform healthcare by enabling the construction of decision support systems. However, a major challenge is the lack of transparency and accountability, as many models do not provide understandable explanations for their recommendations. Explainable Artificial Intelligence (XAI) methods aim to address this challenge by constructing and communicating explanations of how a model works and why it produces a particular output. This can help users evaluate the system and build trust in it, where appropriate.

In this paper, we propose a method for explaining the relative rankings of predictions made by an XGBoost model, which involves understanding and comparing multiple predictions together. Our method uses counterfactual examples to show how changing the feature values of an entity can affect its position within the ranking defined by the model. Unlike traditional counterfactual explanations, which aim to find feature value changes that would result in a different predicted class label by meeting a fixed threshold, the proposed approach is unique in that it aims to identify changes that would bring the predictions in line with a dynamic threshold determined by other data items.

We demonstrate the effectiveness of our approach in a healthcare triage problem. Our framework for counterfactual explanation provides a powerful tool for understanding the relationships between feature values and model rankings and can help promote transparency and accountability in healthcare decision-making and decision support.

Keywords: Explainable AI (XAI), Ranking, Counterfactual explanation

1. Introduction

While much of the Explainable Artificial Intelligence literature for supervised Machine Learning (ML) focuses on explaining classification and regression, explaining ranked outputs of these models has received less attention [1]. The application of ML in ranking has proven to be a useful tool in various industries; for example, in healthcare, ML has been used to predict patient needs such as admission to Intensive Care Units (ICUs) [2]. This is achieved by analyzing various factors such as medical history, current health status, and severity of the illness to learn from past data about how admission is prioritized. The use of ML for ranking patients by need as indicated by past data has the potential to help healthcare providers to make more informed decisions by helping them better understand which features have driven decision-making in the past and by reflecting on their impact when allocating limited resources to future patients.

One of the challenges to the use of Machine Learning (ML) models in the healthcare industry is the perception of ML as a “black box,” where the internal workings of the model are not easily understood or trusted by clinicians [3–6]. To mitigate this issue, the field of Explainable AI (XAI) has emerged as a means of making ML models more transparent and accountable [7, 8]. The term “counterfactual explanation” is used in XAI to describe explanations of predictions of individual instances [9]. Counterfactual examples are based on actual data examples whose feature values are modified to show how much of a change in feature values is needed to produce a change in a model’s output [10]. Such examples are often used in the context of classification to understand the minimum change in feature

*msalimip@uwo.ca Supervisors: Dr. Kamran Sedig, Dr. Daniel Lizotte

This article is © 2023 by author(s) as listed above. The article is licensed under a Creative Commons Attribution (CC BY 4.0) International license (https://creativecommons.org/licenses/by/4.0/legalcode), except where otherwise indicated with respect to particular material included in the article. The article should be attributed to the author(s) identified above.
values that would be required for a specific example to be classified differently. This allows a deeper understanding of the impact of individual feature values on the output of the model and enables users to better understand the reasoning behind the outputs of the model and to make informed decisions themselves.\footnote{Note that the term “counterfactual explanation” is used in the XAI community even if the associated model does not make valid causal inferences. It is an explanation of response of the model to changes in inputs, not of the response of the system the model is intended to describe.}

Despite their utility, counterfactual examples used for classification models are not directly applicable to ranking systems, where we need to explain why the model ranks an example above or below other examples in the output ranking. Furthermore, in ranking systems, it is crucial to understand not only the impact of changes in a single item on its ranking but also how changes in one item affect the ranking of other items in the list.

Our contribution entails modifying existing counterfactual explanations to make them applicable for ranking purposes. In contrast to traditional methods, our proposed approach incorporates the position of an item in the list and how a modification of the item can affect its ranking. The objective of our approach is to determine the minimum change required for an item to be ranked differently in comparison to the other items on the list.

2. **Background**

A counterfactual explanation describes the smallest change to the feature values of an example that results in the model making a meaningful change in output [11, 12]. This is typically defined as a change in the predicted class (for classification), or the prediction reaching a pre-specified threshold (for probability or regression outputs). Counterfactual examples are explored as a way to investigate how tweaking feature values affects the output of the model.

Different ways of generating counterfactual examples have been studied [13–15]. The most popular approaches use optimization. Wachter et al. use an optimization algorithm to generate a new input that is the closest sample to the data instance but makes the model produce a different prediction [16]. Russell et al. used the idea of generative models to synthesize new input instances that are close to the original data but output models with different results [17]. Dandl et al. use a gradient-based optimization algorithm to find counterfactual examples [18]. Another approach to generating counterfactual examples is greedy search. In this approach, feature values are modified iteratively until the model prediction changes. Yang et al. used this approach to find counterfactual examples [19]. Prior works have focused on generating counterfactual examples using different approaches with the aim of explaining the minimum changes required to alter class predictions. In our approach, we modify the traditional greedy algorithm to address the specific needs of ranking problems. To achieve this, we generate counterfactual examples that take into account the relationship between items in a ranked list, providing a better understanding of the changes required to reach a desired rank.

3. **Method**

Our proposed method uses a greedy algorithm to find counterfactual explanations that are applicable to ranked model outputs, which are a collection of outputs for a set of test data that have been sorted according to the output of a probabilistic classification model. The goal of our algorithm is to determine the minimum changes required for a data instance to achieve a different rank. For this purpose, we use a greedy approach given in Algorithm 1. The algorithm takes as input: $M_{model}$ - the machine learning model that produced the ranking; $dataInstance$ - a specific data instance; and $RChange$ - a desired rank change.
The algorithm first finds the most important features $\mathcal{F}^*$ sorted in descending order for the given dataInstance according to Shapley values. Shapley values measure the importance of features by calculating the marginal contribution of each feature in a prediction [20]. The algorithm initializes the set $\mathcal{F}$ with $f^*$ (the first important feature in $\mathcal{F}^*$) and generates InteractionList which is a list of features that have the most interaction with this feature. These are the features whose values have the biggest impact on the relationship between the $f^*$ feature and the outcome; they are available in standard implementations of XGBoost and can be computed for other models as well. To generate counterfactual examples, the approach varies the feature values in the subset $\mathcal{F}_{sub}$, which is obtained by adding one feature at a time from the feature list $\mathcal{F}$). The approach changes the values of these features for dataInstance along a specified range using a grid search, and observing the corresponding model outputs while holding all other features constant. It computes the rank assigned by the machine learning model for each new potential counterfactual example and compares it to the rank of the original data instance.

Given that the proposed approach aims to achieve the desired ranking with the minimum number of changes possible, if a single important feature modification is sufficient to meet the target ranking, then modifying other features can be avoided. However, if changing the value of one of the features in $\mathcal{F}_{sub}$ alone is not able to change the rank of dataInstance, the algorithm considers changing multiple important features together to see if a change in rank can be produced. After each modification to the feature(s), the algorithm replaces the feature value with the modified value and evaluates the MLmodel's output. The premise is that by simultaneously changing important features together with their most strongly interacting features, the algorithm has the best chance of being able to identify a counterfactual example whose rank is at least $R_{\text{change}}$ away from the rank of dataInstance. The algorithm iteratively adds features from the list of most important features and their corresponding interaction list and performs a grid search after each is added. This process is repeated until a change in the feature values results in a counterfactual example whose rank is different from that of dataInstance by the desired amount ($R_{\text{change}}$), at which point the algorithm will stop and return the new counterfactual example.

The objective of our approach is to identify a counterfactual example that involves the minimum possible number of feature modifications. To achieve this, we begin with the most important feature and add its interacting feature one at a time. If this process fails to yield a suitable counterfactual example, it will add another important feature ($f^*$) from $\mathcal{F}^*$ along with its interacting features to $\mathcal{F}$ and repeat the algorithm.

4. Discussion

In order to evaluate the effectiveness of our proposed method, we conducted a study focused on triaging (ranking) patients to admit to the ICU. The triage process involves prioritizing patients based on their level of severity. For this purpose, we utilized a dataset from Sírio-Libanés Hospital [21], which contains patient demographic information, previous disease groupings, blood results, vital signs, and blood gases. The dataset includes labels indicating whether a patient was admitted to the ICU or not. We used this label to train an XGBoost model to predict patient admission probability. We then utilized these prediction probabilities to rank a test set of patients for triage purposes, and applied our proposed method was applied to explain how patients were ranked. Based on our experimental results, we applied our proposed algorithm to various scenarios and discovered that it successfully provided recommendations for achieving desired rankings. To illustrate, in a scenario where we intended to modify a patient’s ranking from 10 to 5, the algorithm advised increasing the temperature by 0.2. This modification succeeded in achieving the desired ranking of 5.
Our method enables users to identify the minimum changes required in an item’s attributes to alter their ranking. With these explanations, users can gain a better understanding of the system’s ranking process, the main contributors to a patient’s rank, and how alternative rankings may be achieved.

5. Conclusion

The modification to the traditional counterfactual analysis approach has the potential to offer more meaningful insights for informed decision-making across a range of applications, particularly in areas where accurate and transparent ranking systems are imperative, such as healthcare. By focusing on the interplay between items in the ranking list, this novel approach has the ability to make machine learning algorithms more transparent and accountable, thereby increasing their reliability and applicability in various domains.

In our ongoing work, we aim to refine the ranking counterfactual algorithm to promote feature sparsity and closeness to the underlying data manifold. Furthermore, we will work towards enhancing the understanding of the ranking system by developing a visual analytics tool that combines the proposed method with other explanation methods. These XAI methods provide different explanations through interactive visual components. By interacting with the tool, users can gain a deeper understanding of the inner working of the ranking model and examine the impact of different feature values on the output. Also, the user can request that an item (patient) be ranked higher or lower by a particular number, and the system will display the required changes to feature values. Ultimately, users can engage with the tool and enhance their comprehension of the ranking model’s internal workings by leveraging a variety of XAI methodologies.
References


