Transferability of counterfactual examples across different models

Project Work - Responsible AI (by Bilal Zafar)

Frederik Hüttemann
Applied Computer Science
Ruhr-Universität Bochum
frederik.huettemann@rub.de

Abstract

An abstract should concisely (less than 300 words) motivate the problem, describe your aims, describe your contribution, and highlight your main finding(s). // In this project, we investigate the transferability of counterfactual examples across different models. Neural Networks are often used to make life critical decisions, such as in healthcare or criminal justice. They also find applications in less critical domains, such as image classification. However, these models are often criticized for being black boxes, which makes it difficult to understand their decisions. Counterfactual examples are a method to explain the decisions of a model by showing how the input would have to change in order to change the output.
It is often assumed that counterfactual examples are model-specific, so their the same counterfacutal example usually cannot be applied on different models. The goal of this project is to look deeper into this assumption and try to find out whether counterfactual examples can be transferred across different models. For this, different types of data and models are used. For the simple cases two numerical datasets are used. For the more complex case, the Imagenet dataset is converted into a binary classification task.
TBD: results and conclusion The results show that counterfactual examples can be transferred across different models, but the transferability depends on the type of data and the model used. In particular, counterfactual examples for numerical data can be transferred to at least some extent, while counterfactual examples for image data are not transferable at all. This suggests that the transferability of counterfactual examples is not a general property, but rather depends on the specific data and model used.

1 Introduction

Counterfactual examples are a method to explain a model's decisions by showing how one or more input features would have to change in order to change the output of the model. This explainability method is often used in life-critical domains, such as healthcare or criminal justice, where a binary classification task is performed. For example, in criminal justice, a model might predict whether a defendant will reoffend within two years.
The generation of counterfactual examples is typically model-specific, as it generally only applies to a particular model. In this project, it is investigated whether counterfactual examples for the base model also work for different models or if they are model-specific. This is done by gradually increasing the difference between the base model and the reference model. In the first experiements both model only differ in the number of trained epochs. In the other experiements differ in their number of parameters namingly layers and hidden units per layer. The goal is to find trends in the transferability of counterfactual examples across different models. The found trends are formulated as hypotheses and tested in the experiments. Using the experiments' results, the hypotheses are either confirmed or rejected.

For the experiements two different types of data is used: tabular data and image data. Using multiple different datasets allows to reliably find trends and proof or disproof the hypotheses. On the tabular data, simpler feed-forward neural networks (FFNN) are used, while on the image data more complex convolutional neural networks (CNN) are used.

2 Related Work

Counterfactual examples are a well-known method to explain the decisions of a model. Machine learning is used in many different domains, such as healthcare, job applications and administrative decisions. In these domains, it is often important to understand the model's decisions and to be able to explain them. Using counterfactual examples, a user can understand how the model's decision would change if one or more input features were changed. This is especially important in life-critical domains, where wrong decisions can have severe consequences. [5]

There are approaches to generate more general, diverse and feasible counterfactual examples. A set of them can be used to understand local model behavior and approximate the decision boundary at a given point.[5]

Additionally, it is stated that the closely linked adversarial examples can be used to attack a model by generating inputs that are misclassified by the model. This is done by changing the input features in a way that the model's decision changes. Adversarial examples can that mislead the base model can also mislead other models which can be different in architecture, training data, and hyperparameters. In the paper [7] an approach is developed which allows attacks on unknown models by using adversarial examples generated on a known model. They managed to generated general adversarial examples that lead to a high number of misclassifications. [7]

3 Approach

In the related work section, it is stated that counterfactual examples are often model-specific and therefore cannot be transferred across different models. However, it is also known that adversarial examples, which are are similar to counterfactual examples, can be used to attack unknown different models.

In this project, an exploratory approach is taken. The goal is to formulate hypotheses on how counterfactual examples can be transferred. For this, some general tests are performed to find trends in the transferability of counterfactual examples across different models. Using these trends, hypotheses are formulated and tested in further experiments using multiple different datasets, different type of models and different types of data. In general, every experiment performed in this project uses two models. The first model is called the *base model* and is used to generate counterfactual examples. The second model is called the *reference model* and is used to evaluate the transferability of the counterfactual examples. Both models are trained on the same training data to ensure that the counterfactual examples are not biased towards a specific model.

4 Experiments

The experiments, which are described in this section, are designed to find trends in the transferability of counterfactual examples. For simplicity, only binary classification is considered. To solve a binary classification task, different types of models can be used. The tested model architectures are FFNN, CNN, Random Forest (RF) and Kernel SVM (K-SVM). For the first two model architectures, the generation of counterfactual examples is straight forward, as these models are differentiable and therefore the counterfactual examples can be generated using gradient descent. RF and K-SVM are not differentiable, so for these models the DiCE library is used [6]. This approach also minimizes a loss function using gradient descent. The loss function here is a combination of the distance between the original input, the distance to the counterfactual class while ensuring diversity among the generated counterfactuals. As I am only generating a single counterfactual example per datapoint, the diversity is not considered.

To evaluate the transferability of counterfactual examples, a metric is needed that quantifies how well the counterfactual examples transfer to the reference model. This metric is called the *transferability rate*. The transferability rate is given by equation 1 and is defined as the ratio of the number of counterfactual examples that are correctly classified in the counterfactual class by the reference model to the total number of counterfactual examples generated by the base model. The transferability rate is a value between 0 and 1, where 0 means that none of the counterfactual examples are correctly classified in the counterfactual class by the reference model and 1 means that all counterfactual examples are correctly classified in the counterfactual class.

$$\text{transferability rate} = \frac{\text{counterfactual examples classified in counterfactual class}}{\text{total number of counterfactual examples}} \tag{1}$$

Because the training and generation of counterfactual examples is computationally expensive, the transferability rate is only calculated for a limited number (250) of counterfactual examples. Eventough this number is low, it is sufficient to find trends in the transferability of counterfactual examples as well as to formulate and test hypotheses. The transferability rate is calculated for each experiment and for each model.

To find trends in the transferability of counterfactual examples, the idea was to start as simple as possible and then gradually increase the complexity of the experiments.

4.1 Experiment 1: Base model vs. reference model with different number of epochs

The first experiements performed are the simplest ones. Here, both the base model and the reference model are the same model architecture. The base model is trained for a fixed number of epochs on the training data. The reference model is a copy of the trained base model, but its training is continued for additional epochs. For additional epoch zero, the base model and the reference model are identical. Here, the transferability rate should be one. As the number of additional epochs increases, the transferability rate should decrease as the reference model's weights are updated and differ more and more from the base model's weights. Therefore the reference model's decision boundary is updated and the counterfactual examples generated by the base model are not necessarily classified in the counterfactual class by the reference model anymore. The hypothesis for this experiment is:

- **Hypothesis 1:** The transferability rate decreases as the number of additional epochs increases.
- **Hypothesis 2:** The transferability rate is one for zero additional epochs.

It is important to note, that this experiment can only be performed on models, which rely on gradient descent to update their weights as they are trained for a fixed number of epochs. This is the case for FFNN and CNN, but not for RF and K-SVM, because these models have a fixed number of iterations they perform to converge. On convergence, the models are not updated anymore and therefore the transferability rate is always one.

4.2 Experiment 2: Base model vs. reference model with different number of parameters

The second experiment is similar to the first one. Again the base model and the reference model are the same model architecture and trained with the same hyperparameters. The only difference is that the models both trained from scratch and therefore differ in their initialization. As the random initialization of the weights is different, the models learn different decision boundaries. The more parameters the model has, the more complex the decision boundary can be. Two models with a high number of parameters probably learn different decision boundaries and therefore have a lower transferability rate than two models with a low number of parameters. The hypothesis for this experiment is:

- **Hypothesis 3:** The transferability rate decreases as the number of parameters increases.

To test this hypothesis, the experiements are performed with different model architectures and different type of data.

4.3 Data

The used data can be divided into two categories: tabular data and image data. For the tabular data, two datasets are used: a *Compas* dataset [8] and an *ACS* dataset [1].

The *Compas* dataset contains information about criminal defendants, such as their age, the prior number of offenses, and the type of offense. The goal is to predict whether a defendant will reoffend within two years.
The *ACS* dataset contains information about the American Community Survey, such as the age, education, and income of individuals. The dataset label is wheather the the data point represents a person who is employed or not.
For image classification data there are many different possible datasets to choose from. As I wanted a binary classification task, I had to convert all image classification datasets into a binary classification task. The used datasets are *CIFAR-10* [3], *Imagenet* [2], *MNIST* [4] and *Fashion MNIST* [9].
MNIST and *Fashion MNIST* are both low resolution (28x28 pixels) grayscale images of handwritten digits and clothing items, respectively. It contains 10 classes, one for each digit or clothing item. To convert it to a binary classification task, every datapoint representing a *digit 0* becomes a positive example, while all other datapoints become negative examples.
For *CIFAR-10*, the images are also low resolution (32x32 pixels) but colored of 10 classes. Similar to *MNIST*, the positive class is *airplane* and the negative class is everything else.
The same approach has been taken for *Imagenet*, where the images are high resolution (224x224 pixels), colored images and representing 1000 classes. The positive class is *orange* and the negative class is everything else. Here, the positive and negative examples are balanced, so that the number of positive and negative examples is equal. The selection of the rest class is done by randomly selecting images from the other classes.
The datasets are split into a training set and a test set. The training set is the same for all experiments, the test set is on the one hand used to evaluate and compare models but also to generate counterfactual examples. This way, the counterfactual examples are not biased towards a specific model because they are generated on the same, new data.

4.4 Evaluation method

The evaluation method is based on the transferability rate, which is defined in equation 1. The transferability rate is calculated for each experiment and for each model. The transferability rate is a value between 0 and 1, where 0 means that none of the counterfactual examples are correctly classified in the counterfactual class by the reference model and 1 means that all counterfactual examples are correctly classified in the counterfactual class.
To identify trends, a linear line is fitted to to the transferability rates.
To verify the stated hypothesis, the Spearman rank correlation coefficient is calculated. The Spearman rank correlation coefficient is a measure of correleation between two variables. It is used to determine whether there is a monotonic relationship between the two variables. The Spearman rank correlation coefficient is a value between -1 and 1, where -1 means that there is a perfect negative correlation, 0 means that there is no correlation and 1 means that there is a perfect positive correlation. The hypotheses can be confirmed or rejected by also calculating the p-value. The p-value is a measure of the statistical significance of the correlation. As a common threshold for statistical significance, a p-value of 0.05 is used. If the p-value is below this threshold, the hypothesis is confirmed, otherwise it is rejected. [10]

5 Results

Both experiments are performed on the tabular data and the image data.

5.1 Experiment 1: Base model vs. reference model with different number of epochs

- If you're a default project team, you should **report the accuracy and Pearson correlation scores you obtained on the test leaderboard** in this section. You can also report dev set results if you'd like.
- Comment on your quantitative results. Are they what you expected? Better than you expected? Worse than you expected? Why do you think that is? What does that tell you about your approach?

6 Analysis

Your report should include *qualitative evaluation*. That is, try to understand your system (e.g., how it works, when it succeeds and when it fails) by inspecting key characteristics or outputs of your model.

7 Conclusion

Summarize the main findings of your project and what you have learned. Highlight your achievements, and note the primary limitations of your work. If you'd like, you can describe avenues for future work.

8 Outlook

Because of limited time, the experiments are not exhaustive and only a limited number of experiments are performed. More experiments could be performed especially on the image data as each experiment takes a long time to run. For the last three weeks my personal GPU did run non stop to generate counterfactuals and train the models. Eventough I started early with the experiments, I was not able to finish all experiments I wanted to perform.

The generation of counterfactual examples does not take into account that the feature values might have different values. For tabular data some features represent binary values while others represent continuous values. The counterfactual algorithm used does not take those value ranges into account, which should lead to lower results as it then only relies on the model's bias.

I also limited my experiments on neural networks while other decision making models could be used as well. Examples could be decision trees or a Kernel SVM. These models are often more interpretable as they generate linear decision boundaries. Here, it might also be possible to draw the decision boundaries as well as the counterfactual examples in a 2D plot. This would allow a visual understanding of why some counterfactuals are transferable and others are not. For this, apporaches like PCA or t-SNE could be used to reduce the dimensionality of the data.

References

[1] U.S. Census Bureau. American community survey (acs) dataset, 2018. Accessed: 2025-07-01.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[3] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[4] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[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*, FAT* '20, page 607–617, New York, NY, USA, 2020. Association for Computing Machinery.

[6] 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*, FAT* '20, page 607–617. ACM, January 2020.

[7] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples, 2016.

[8] ProPublica. Compas dataset, 2016. Accessed: 2025-07-01.

[9] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[10] Jerrold Zar. *Spearman Rank Correlation*, volume 5. 07 2005.

A Usage of AI tools

In this project, AI tools were used. For the coding part mainly GitHub Copilot was used for code completiton as am I used to program with it at work. Therefore, I used it in this project as well. It was mainly used to complete code lines are to complete configs as they are often repetitive. Additionally, I had trouble with the *dice-ml* library which I used to generate counterfactual examples for non differentiable models. The library is not well documented and therefore I asked Copilot to help me with the correct usage of the library. The rest of the code was written by myself or was already given by some excercises from the lecture.
For research work, I used ChatGPT to summarize two papers I found on the topic of counterfactual examples. The papers were not directly related to my project, but they helped me to understand the topic better. As they were quite extensive, I used ChatGPT to summarize them and to get an overview of the main findings.
Additionally, a few sentences sentences were rewritten by ChatGPT as I had trouble with the wording. This only applies to a few sentences in the introduction and the related work section. The rest of the report was written by myself.

B Statutory Declaration

I officially ensure, that this paper has been written solely on my own. I herewith officially ensure, that I have not used any other sources but those stated by me. Any and every parts of the text which constitute quotes in original wording or in its essence have been explicitly referred by me by using official marking and proper quotation. This is also valid for used drafts, pictures and similar formats.

Ort, Datum	Unterschrift (Frederik Hüttemann)