TP Explicabilité
Counterfactual Explanation
Jean-François Bonastre- 2022

Introduction

Comme vu en cours, l'explication contrefactuelle consiste à rechercher l'élément (significatif) indispensable à une décision : l'absence de cet élément implique un changement majeur (classe ou saut de valeur) dans la décision.

La définition suivante résume bien l'idée générale et le principe d'application à l'IA : *" A counterfactual explanation of an outcome or a situation Y takes the form "If X had not occured, Y would not have occured. In the context of a machine learning classifier X would be an instance of interest and Y would be the label predicted by the model. The task of finding a counterfactual explanation is then to find some X' that is in some way related to the original instance X but leading to a different prediction Y'. "*

Toute la complexité du « counterfactual explanation » réside dans le « « that in some way related to the original instance ».

L'approche contrefactuelle partage avec LIME et SHAP le fait d'être locale : elle peut expliquer une décision donnée. Elle partage également la caractéristique d'être « Model Agnostic », soit de pouvoir s'adapter à tout modèle/méthode. Par rapport à ces méthodes, l'approche contrefactuelle a l'avantage de proposer des explications « matérialisées » dans l'espace des données et potentiellement facilement interprétables. Il n'est pas nécessaire de manipuler des notions complexes, comme les valeurs de Shapley ou les modèles (localement) interprétables de LIME, les explications sont directement dans le domaine des données, qui doit être le domaine d'expertise des opérateurs.

L'approche contrefactuelle offre aussi l'avantage potentiel de tenir compte de toutes les interactions entre les données : il n'y a pas d'apriori de « marginalité » dans l'étude d'un cas précis (attention cependant, les méthodes sous-jacentes, pour générer les analyses, peuvent elles intégrer ce type d'apriori)

Nous allons dans ce TP mettre en œuvre en pratique une solution de « Counterfactual Explanation » dans le cadre d'une application de comparaison de voix. La comparaison de voix consiste à déterminer si 2 extraits vocaux ont été prononcés par la même personne ou non.

Le système de comparaison de voix que vous devez « expliquer » suit la structure qui vous a été présentée dans le TD « Counterfactual », sous forme simplifiée :

- Un système qui prend en entrée deux enregistrements sonores, A et B, sous forme de fichiers wav
- Et qui retourne en sortie un score, S, entre -1 et 1, plus S est grand plus la probabilité que A et B proviennent du même locuteur est forte.
- Un module de décision qui compare S à un seuil T (à fixer) et répond 1 si S>=T, 0 sinon

Vous disposez également de deux listes de tests de comparaison de voix :

- « tar » contient des couples de fichiers (A,B) tels que A et B proviennent du même locuteur

- « non » contient des couples de fichiers (A,B) tels que A et B proviennent de locuteurs différents

Le seuil, T, sera à fixer manuellement en fonction de la distribution des scores de tar et non.

Evaluer la « performance globale » du système, consiste à calculer 4 grandeurs :

- True Positive (TP) : le nombre de test de tar pour lesquels la décision a été positive (score> seuil)
- True Negative (TN) : le nombre de test de non pour lesquels la décision a été négative (score<= seuil)
- False Negative (FN) : le nombre de test de tar pour lesquels la décision a été négative (score<= seuil)
- False Positive (FP) : le nombre de test de non pour lesquels la décision a été positive (score> seuil)

Et trois taux d'erreur :

- False Reject (FR) : FR=FN/(TP+FN)
- False Acceptance (FA) : FA= FP/(FP+TN)
- La moitié du taux total : HTER=(FR+FA)/2

Le travail à effectuer

Mettre en place une solution de CA basée sur les fichiers audio. L'idée est de déterminer quelle est la partie minimale du signal d'entrée dont l'absence permet d'inverser la décision positive (nous nous intéresserons qu'aux tests positifs) ou d'accroitre significativement le score correspondant à une décision négative.

A. Mise en place

Prendre en main le système proposé et savoir calculer FR, FA et HTER pour différentes valeurs du seuil

B. Principe du « knock-out »

L'idée principale est de procéder par « knock-out ». Il s'agit d'enlever alternativement une petite portion du signal d'entrée (le processus sera réalisé alternativement sur A et sur B), Fx, et de recalculer le test avec le signal modifié, pour obtenir le score correspond, Sx. La partie de signal Fx dont le retrait a mené à une modification maximale du score dans le sens souhaité est la partie la plus importante.

Ce processus peut être répété pour trouver, parmi le signal diminuer de Sx, une nouvelle zone, etc.

C. Implémentation

Réaliser la mise en œuvre du processus de calcul d'une CA par knock-out, où CA est l'ensemble des Sx de taille nécessaire et suffisante pour obtenir l'effet souhaité sur le score.

D. Variante

Recherchez la zone CONTINUE du signal, telle qu'elle soit nécessaire et suffisante pour obtenir l'effet recherché sur le score.

E. Essayez de choisir une taille de Sx qui soit un multiple de 80 ms.

F. Note IMPORTANTE sur le rendu

Comme pour la partie TD, l'analyse et la présentation des résultats est privilégié à la quantité. Votre recul individuel sur les solutions proposées et sur vos implémentations sera apprécié.

Il est conseillé d'analyser les résultats en fonction de :

- Le type de test parmi TP, FP, TN et FN
- La valeur du score proche/loin du seuil

G. Accès au package logiciel/data

Le système est une implémentation de speechbrain (https://speechbrain.github.io/) réalisée sous forme de docker, avec un client en python.

L'extracteur a été appris sur la base voxceleb2 (https://www.robots.ox.ac.uk/~vgg/data/voxceleb/vox2.html).

La petite base d'exemple pour réaliser des tests tar et non est issue de voxceleb1 (https://www.robots.ox.ac.uk/~vgg/data/voxceleb/vox1.html)

Solutions existantes

Le lien ci-dessous montre l'exemple d'une méthode simple d'application de ce principe :

https://docs.seldon.io/projects/alibi/en/latest/methods/CF.html

https://github.com/interpretml/DiCE est également un toolkit renommé.

Références

Verma, S., Dickerson, J., & Hines, K. (2020). Counterfactual explanations for machine learning: A review. *arXiv preprint arXiv:2010.10596*.

Delaney, E., Greene, D., & Keane, M. T. (2021, September). Instance-based counterfactual explanations for time series classification. In *International Conference on Case-Based Reasoning* (pp. 32-47). Springer, Cham.

Byrne, R. M. (2019, August). Counterfactuals in Explainable Artificial Intelligence (XAI): Evidence from Human Reasoning. In *IJCAI* (pp. 6276-6282).

Keane, M. T., & Smyth, B. (2020, June). Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable AI (XAI). In *International Conference on Case-Based Reasoning* (pp. 163-178). Springer, Cham.

Keane, M. T., Kenny, E. M., Delaney, E., & Smyth, B. (2021). If only we had better counterfactual explanations: Five key deficits to rectify in the evaluation of counterfactual xai techniques. *arXiv preprint arXiv:2103.01035*.