Revue de FACE : Feasible and Actionable Counterfactual Explanations

Principle of Deep Learning

Louise Gabion, Adrien Tarquinio

I. Introduction

Dans ce papier nous nous intéressons à un article de *Poyadzi and al.* [1] nommé "FACE : Feasible and Actionable Counterfactual Explanations". L'article propose une approche novatrice pour générer des explications contrefactuelles qui soient à la fois réalisables et exploitables dans des contextes réels.
Le but ici sera d'éclaircir le principe et de le comparer à d'autres proposition de explicabilité contrefactuelles.

II. Principe

Les explications contrefactuelles (*Contrastive Explanations*) visent à chercher : « Quels changements minimaux (*Counterfactuals*) apporter pour passer d'un premier résultat (*The Fact*) un résultat différent (*The Foil*) ? » [2]. Là où le cerveau humain arrive assez bien à répondre inconsiemment à cette question, dans différents domaines, comme le Machine Learning (ML) ou dans le Depp Learning (DL), la réponse à cette question n'est pas automatique. Ces dernières années, dans une volontée *Explanable AI* (EAI), différentes méthodes ont été proposées pour répondre à cette question. Cependant, ces méthodes présentent deux limites majeures.
Premièrement, les solutions proposées peuvent être situées dans des zones de faible densité de données, les rendant peu représentatives et difficilement atteignables, c'est ce qu'on appelle l'irréalité des recommandations.
Deuxièmement, les changements suggérés ne tiennent pas compte de la progression réaliste entre l'état actuel et l'état souhaité, rendant les recommandations peu exploitables, c'est ce qu'on appelle manque de cheminement faisable.
Contrairement aux méthodes traditionnelles qui privilégisent le chemin le plus court entre deux états, mais parfois trop peu faisable en réalité, FACE propose de générer des explications qui s'inscrivent dans des régions de haute densité de l'espace des données, c'est-à-dire des zones où les exemples sont couramment observés. Pour cela, la méthode utilise des graphes construits à partir des k plus proches voisins (k-NN) dans l'espace des données, pondérés par des distances tenant compte de la densité locale. Cela permet de trouver des trajectoires de changement qui suivent la structure naturelle des données et évitent les régions irréalistes ou aberrantes.
L'approche repose ensuite sur la recherche d'un chemin le plus court dans ce graphe entre un point donné (par exemple, un individu refusé pour un prêt) et un point cible (par exemple, un individu similaire ayant obtenu un prêt), tout en respectant certaines contraintes. Ces contraintes peuvent être définies par l'utilisateur, comme le gel de certaines caractéristiques (ex. : l'âge ou le sexe), ou par des critères métier (ex. : ne pas suggérer un doublement irréaliste du revenu). Le contrefactuel proposé n'est donc pas simplement un point cible arbitraire, mais le résultat d'un enchaînement de modifications réalisables à chaque étape du chemin. Cette construction progressive garantit que l'explication est non seulement plausible, mais qu'elle fournit également une feuille de route concrète que l'utilisateur peut suivre pour changer sa situation. Ainsi, FACE rend les explications contrefactuelles non seulement interprétables, mais aussi concrètement utiles dans la prise de décision individuelle.
La Figure 1 illustre donc ces propos. En partant donc de l'état

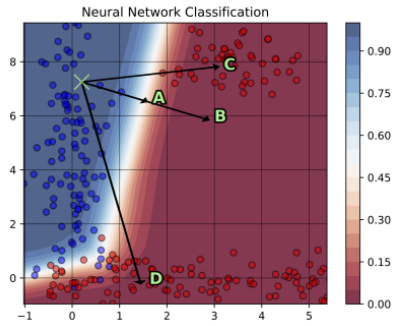


Figure 1 – Présentation de 4 chemins logiques possibles pour passer d'un état à un autre [1]

représenté par $\times$, les chemins A, B, C et D sont logiques et possibles. A correspond au chemin minimisant la longueur de ce dernier. Correspondant à un emplacement limitrophe entre les deux classes, B semble une meilleure option. Cependant, A et B se trouve dans une zone avec une faible densité de résultats. On va donc préférer C et D, se trouvant dans une zone de plus forte densité. D correspond au chemin choisi par l'algorithme FACE car ce chemin présente plus d'états intermédiaires pour atteindre le changement de classe souhaité.
Une analogie serait alors de considérer notre point $\times$ comme représentant un employé sous-qualifié cherchant de quoi se payer une maison. La classe représenté en bleue serait le résultat "l'employé n'a pas assez d'argent" et la classe en rouge "l'employé peut se payer la maison". Dans cette exemple, le chemin A serait que l'employé obtient juste assez pour se payer la maison, au détriment d'avoir de quoi se nourir pendants les prochains mois : un résultats, rapide mais instable et ne garantissant pas de rester dans l'état rouge. Le chemin B serait que l'employé décide de voler cet argent : un résultat rapide et plus certain, mais surtout très instable. Le chemin C serait que l'on propose à cet employé peu qualifié de changer d'emploi et de négocier un salaire deux fois supérieur au sien. Cette fois le résultat est certain et stable mais le chemin est fastidieux et difficilement réalisable (peu d'étapes intermédiaires). Le résultat D reviendrait à dire que l'employé choisi de faire un petit prêt, choisi de complété sa formation dans l'idée d'évoluer dans son métier et donc d'évoluer dans son salaire et d'attendre quelques années de plus : un chemin plus long mais avec des états intermédiaires accéssibles et un résultat stable.

III. Mise en oeuvre

L'algorithme FACE utilise alors la méthode de l'estimateur à noyau de densité (*Kernel Density Estimator*) (KDE) pour déterminer si les résultats possibles se trouvent dans une zone à faible densité ou non (Cf chemin A et B, Figure 1). La méthode des k plus proches voisin (*k-Nearest Neighbour*) (k-NN) ainsi que la méthode des ϵ-graphs sont aussi utilisés pour la notion de "voisin" d'une donnée.
Soit x_i l'état de départ et x_j l'état lié à x_i. On nomme f la fonction de passage de x_i à x_j. Suivant la méthode utilisée, les poids sont calculées de différentes manières [1] :

$$w_{ij} = f(p(\frac{x_i + x_j}{2})).\|x_i - x_j\| \tag{1}$$

pour la méthode KDE.

$$w_{ij} = f(\frac{\frac{k}{N}}{\|x_i - x_j\|}).\|x_i - x_j\| \tag{2}$$

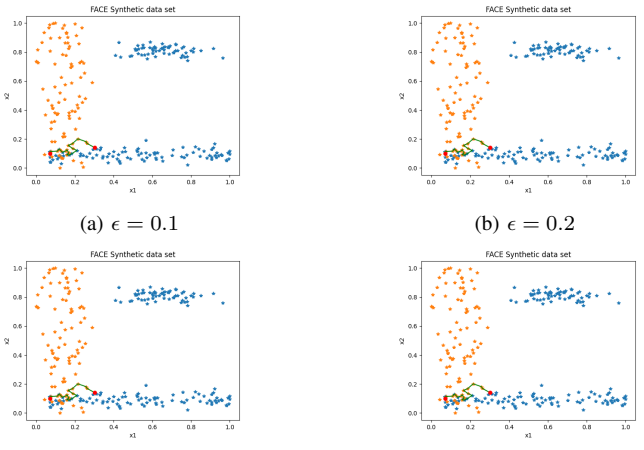
pour le graph k-NN.

$$w_{ij} = f(\frac{\epsilon^d}{\|x_i - x_j\|}).\|x_i - x_j\| \tag{3}$$

pour le graph epsilon. Si x_i et x_j ne sont pas liés alors $w_{ij} = 0$ qu'importe la méthode.

A. Résultats

Nous avons récupéré l'agorithme FACE et l'avons testé avec son propre set de donnée (*synthetic_face_datatset.pk*).
Nous obtenons les chemins suivants pour différentes valeurs d'epsilon.



(a) $\epsilon = 0.1$ (b) $\epsilon = 0.2$

(c) $\epsilon = 0.5$ (d) $\epsilon = 0.8$

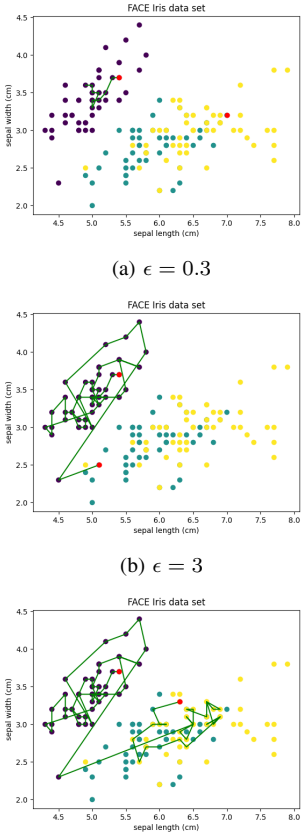
FIGURE 2 – Résultats FACE sur le dataset *synthetic_face_datatset.pk* pour différentes valeurs d'epsilon

On n'observe pas de différence notable suivant ces valeurs de epsilon, simplement l'algorithme a mis plus de temps pour les epsilons les plus grands.

IV. Comparaison avec DiCE

Les méthodes FACE et DiCE, bien que toutes deux dédiées à la génération de contrefactuels dans des systèmes d'intelligence artificielle, se distinguent par leurs objectifs, leurs méthodologies et leurs applications. Par contraste avec FACE, DiCE présenté par Mothilal et al. (2020) [3], vise à explorer une variété de contrefactuels, en couvrant plusieurs scénarios de changement de classe via une optimisation multi-objectifs. L'approche de DiCE prend en compte la proximité, la validité et la diversité des exemples générés, ce qui permet de proposer plusieurs alternatives de manière plus diversifiée, mais au prix d'une contrainte moins stricte en termes de réalisme. En outre, DiCE ne dépend pas d'une structure de graphe explicite, mais s'appuie plutôt sur un espace d'entrée et une fonction de perte imposant des contraintes supplémentaires. Cela rend l'interprétabilité de DiCE moins évidente, nécessitant un filtrage plus poussé des suggestions pertinentes. Alors que FACE trouve des applications dans des domaines nécessitant des explications causales et justifiées (par exemple, la finance, la santé, l'embauche), DiCE est plus adapté à des contextes exploratoires, notamment dans des situations éthiques ou critiques, où la diversité des contrefactuels peut offrir une meilleure compréhension des alternatives possibles.

Nous avons alors souhaité comparer les deux méthodes avec le même dataset IRIS. Concernant DiCE, le modèle est disponible sous forme d'une librairie. De ce fait, son utilisation est plus iné avec un dataset de donnée différente. Ce n'est pour le coup pas le cas de FACE, pour lequel nous avons dû lourdement modifier le code afin de lui faire utiliser ces données.De ce fait, nos résultats ne sont pas pertinents.



(a) $\epsilon = 0.3$

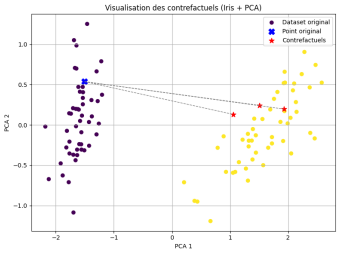
(b) $\epsilon = 3$

(c) $\epsilon = 8$

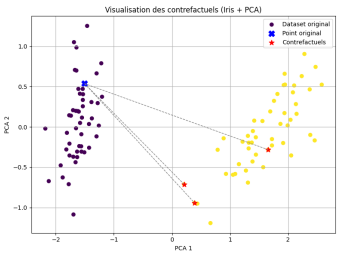
FIGURE 3 – Résultats FACE sur le dataset *IRIS* pour différentes valeurs d'epsilon

On observe sur la Figure 3 le besoin d'utiliser une epsilon

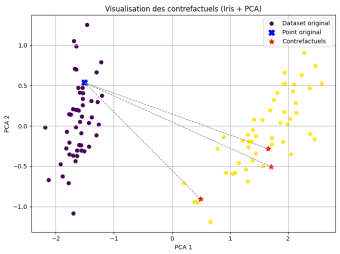
beaucoup plus élevé avec ce dataset, de façon à ce qu'il y ait bien un changement de classe.



(a) method = random



(b) method = kdtree



(c) method = genetic

FIGURE 4 – Résultats DiCE sur le dataset *IRIS* pour différentes méthodes

De manière générale, les résultats sont plus pertinant avec la méthode *genetic*. Ils tombent dans une zone de forte densité plus souvent.

V. CONCLUSION

Nous avons pu découvrir le fonctionnement de FACE et expérimenter ce dernier. Nous avons décidé de le comparer à un autre algorithme, DiCE, pour cerner les différentes approches actuelles dans le domaine. Un problème de pertinence nous empêche de tirer des conclusions précises mais nous avons pu déterminer que FACE semble plus adapté dans des domaines ayant des explications causales et justifiés. Quant à lui, DiCE est plus adapté pour des problèmes impliquant des contextes plus exploratoires, moins terre à terre.

RÉFÉRENCES

[1] R. POYIADZI, K. SOKOL, R. SANTOS-RODRIGUEZ, T. D. BIE et P. FLACH, "FACE: Feasible and Actionable Counterfactual Explanations", in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 7 fév. 2020, p. 344-350. DOI : 10.1145/3375627.3375850. arXiv : 1909.09369[cs]. adresse : http://arxiv.org/abs/1909.09369 (visité le 17/04/2025).

[2] M. J. ROBEER, "Contrastive explanation for machine learning", mém. de mast., 2018.

[3] R. K. MOTHILAL, A. SHARMA et C. TAN, "Explaining machine learning classifiers through diverse counterfactual explanations", in *Proceedings of the 2020 conference on fairness, accountability, and transparency*, 2020, p. 607-617.

VI. ANNEXE

L'ensemble des codes sont disponibles sur GitHub : https://github.com/Hysa0/Projet_PDL.git