Amélioration des explications contrefactuelles pour les recommandations à l'aide de SHAP

Jinfeng Zhong*, Elsa Negre*

*Paris-Dauphine University, PSL Research University,
CNRS UMR 7243, LAMSADE, 75016 Paris France
jinfeng.zhong@dauphine.eu
elsa.negre@lamsade.dauphine.fr

Résumé. Les explications dans les systèmes de recommandation aident les utilisateurs à mieux comprendre pourquoi de telles recommandations sont générées. Expliquer la recommandation est crucial pour renforcer la confiance et la satisfaction des utilisateurs. Étant donné que les systèmes de recommandation deviennent de plus en plus impénétrables, expliquer directement les recommandations devient parfois impossible. Les méthodes d'explication post-hoc qui n'élucident pas les mécanismes internes des systèmes de recommandation sont des approches populaires. Les méthodes d'explication post-hoc telles que SHAP génèrent des explications en construisant des modèles de substitution plus simples pour se rapprocher des modèles originaux. Cependant, l'application directe de telles méthodes pose plusieurs soucis : (1) Il se peut que les explications post-hoc ne soient pas fidèles aux modèles de recommandation originaux puisque les mécanismes internes ne sont pas élucidés; (2) Les résultats retournés par des méthodes telles que SHAP ne sont pas triviales à comprendre pour la plupart des utilisateurs, car des connaissances mathématiques sont nécessaires. Dans ce travail, nous présentons une méthode d'explication des recommandations à l'aide de SHAP qui peut générer des explications contrefactuelles facilement compréhensibles avec une forte fidélité au modèle original.

1 Introduction

L'article résumé ici, intitulé "Shap-enhanced counterfactual explanations for recommendations" (Zhong et Negre, 2022a), a été accepté et présenté à la conférence internationale *37th ACM/SIGAPP Symposium on Applied Computing (SAC '22)*. Dans Zhong et Negre (2022a), nous nous intéressons à l'explicabilité dans les systèmes de recommandation (SRs). Plus spécifiquement, nous étudions des explications contrefactuelles à l'aide de SHAP (Lundberg et Lee, 2017). Les SRs sont de plus en plus déployés pour aider les utilisateurs à trouver des articles qui les intéressent. Par conséquent, les SRs sont devenus des outils d'aide à la décision importants et omniprésents. Cependant, les techniques actuelles des SRs deviennent de plus en plus impénétrables en raison de la complexité des modèles et de la grande dimensionnalité des données avec lesquelles les modèles fonctionnent. Des modèles "boîte noire" tels que les

réseaux de neurones sont déployés pour générer des recommandations, ce qui soulève des inquiétudes concernant des propriétés indésirables telles que le manque de transparence et les préjugés d'illégalité (Rudin, 2019). Dans certains domaines tels que la finance et la médecine où l'impact de mauvaises décisions ne peut être négligé, les utilisateurs s'attendent à ce que les SRs génèrent non seulement des recommandations de haute qualité, mais également des explications pour justifier de telles recommandations. Selon les modèles utilisés dans les SRs, diverses méthodes pour générer des explications ont été développées (Zhang et Chen, 2018). Cependant, l'application de ces méthodes est limitée à des techniques de recommandation spécifiques, ce qui limite leur adaptabilité, leur réutilisation et leur généricité. Cela signifie que chaque fois qu'une nouvelle technique de recommandation est développée, une méthode d'explication correspondante doit être développée. Ainsi, peut-on trouver une méthode générique pour expliquer pourquoi de telles recommandations sont générées ?

Les avancées récentes de l'Intelligence Artificielle (IA) explicable/interprétable éclairent ce sujet. Afin d'expliquer les sorties d'un système d'IA, deux stratégies sont possibles : (1) Adopter des modèles transparents, ce qui signifie que le fonctionnement des modèles soit compréhensible par les humains, e.g. les modèles linéaires simples et/ou les règles d'association ; (2) Fournir des explications post-hoc sans élucider précisément le fonctionnement des modèles. Les modèles "simples" sont faciles à comprendre, mais leurs applications sont limitées par une précision faible (compromis entre complexité et précision). LIME (Ribeiro et al., 2016) et SHAP (Lundberg et Lee, 2017) sont, selon la littérature, des outils capables de fournir des explications post-hoc. Ils peuvent expliquer pourquoi une instance est proposée en estimant l'importance de chaque caractéristique de cette instance. Il existe des travaux qui expliquent les recommandations en calculant directement, grâce à LIME et/ou SHAP, l'importance de chacune de leurs caractéristiques. Les principales limites et préoccupations liées à l'application directe de telles méthodes incluent : (1) Il se peut que les explications post-hoc ne soient pas fidèles aux modèles de recommandations originaux et parfois ces explications sont contradictoires avec la recommandation ; (2) Pour certains utilisateurs, comprendre les résultats retournés par LIME et SHAP n'est pas trivial puisque des connaissances mathématiques plutôt complexes sont requises. Par exemple, les sorties de SHAP incluent la figure "force_plot" qui nécessite que les utilisateurs aient des connaissances en théorie des jeux. Par conséquent, LIME et SHAP sont de bons outils pour les développeurs de modèles pour visualiser les comportements des modèles et peuvent les aider à déboguer les modèles. Cependant, pour les utilisateurs, les importances/résultats retounés ne sont pas faciles à comprendre.

Comment utiliser ces méthodes d'explication post-hoc pour expliquer n'importe quel modèle de recommandation tout en évitant leurs limites ? Générer des explications contrefactuelles facilement compréhensibles et avec une forte fidélité au modèle de recommandations original est une solution potentielle. Contrairement à LIME et SHAP qui se rapprochent des modèles originaux, les explications contrefactuelles recherchent les changements minimaux nécessaires pour modifier les sorties des modèles originaux (Wachter et al., 2017). Par conséquent, les explications contrefactuelles ont une forte fidélité (Kaffes et al., 2021). Une explication contrefactuelle typique décrit une situation causale : "Si A ne s'est pas produit, alors B ne se serait pas produit" (Molnar, 2020). Une méthode naïve pour rechercher des explications contrefactuelles consiste à énumérer tous les sous-ensembles de caractéristiques d'une instance (Martens et Provost, 2014). Une préoccupation majeure est que la complexité est exponentielle par rapport au nombre de caractéristiques utilisées dans le modèle original. Dans

Zhong et Negre (2022a), nous montrons que SHAP peut être adapté pour générer des explications contrefactuelles sans énumérer tous les sous-ensembles de caractéristiques d'une instance. Plus de détails sur LIME et SHAP seront présentés dans la section 2.

2 LIME et SHAP

Dans cette section, nous comparons les deux méthodes et montrons pourquoi nous avons choisi SHAP.

Les similarités entre LIME et SHAP comprennent : (1) LIME et SHAP peuvent générer des explications post-hoc pour une seule instance à expliquer sans avoir à élucider les mécanismes internes des modèles. Par conséquent, LIME et SHAP peuvent être utilisés pour expliquer tous les modèles ; (2) LIME et SHAP fournissent une visualisation intuitive qui montre la contribution de chaque caractéristique, ce qui aide les développeurs à comprendre le fonctionnement de modèles complexes. Il convient de noter que pour les utilisateurs ordinaires, il n'est pas facile de comprendre les résultats obtenus par LIME et SHAP, car des connaissances mathématiques sont nécessaires.

Cependant, à chaque fois que LIME est appliqué, il sélectionne un nombre limité de caractéristiques et génère de nouveaux points de données. Par conséquent, il se peut que les modèles linéaires (par exemple, la régression de Ridge) générés soient différents (c'est-à-dire les caractéristiques utilisées et les coefficients renvoyés), ce qui soulève des inquiétudes quant à la stabilité de LIME (Visani et al., 2020; Zafar et Khan, 2019). SHAP est fondamentalement fondé sur la théorie des jeux donc SHAP assure un calcul équitable de la contribution de chaque caractéristique. Par conséquent, les propriétés souhaitées telles que la précision et la cohérence locales sont garanties (Lundberg et Lee, 2017). Finalement, nous avons choisi SHAP pour générer des explications post-hoc.

Nous invitons les lecteurs à se référer à notre article Zhong et Negre (2022a) ainsi que LIME (Ribeiro et al., 2016) et SHAP (Lundberg et Lee, 2017) pour plus de détails sur la comparaison entre eux.

3 Notre proposition

Selon SHAP, le "jeu" est la prédiction et les "joueurs" sont les caractéristiques que le modèle de recommandation prend en compte. SHAP calcule la contribution de chaque caractéristique, dans la suite de cet article, nous notons la contribution de la caractéristique j comme ϕ_j. Cependant, interpréter ϕ_j comme une déclaration contrefactuelle n'est pas une manière judicieuse de résumer l'information puisque ϕ_j est la contribution marginale moyenne de la caractéristique j (Kumar et al., 2020). Bien qu'il ne soit pas approprié d'interpréter ϕ_j comme une déclaration contrefactuelle, SHAP permet de guider la recherche d'explications contrefactuelles.

Nous proposons l'algorithme 1 pour générer des explications contrefactuelles. L'idée principale est d'appliquer SHAP pour sélectionner les caractéristiques importantes i.e. celles pour lesquelles ϕ_j est positive. Ensuite, la valeur de la caractéristique qui est classée en tête est modifiée. Dans l'étape 13, x_j est remplacé par x_j^{new}. Une autre liste de recommandations $\{i'_1, i'_2, i'_3, \ldots, i'_{max}\}$ est calculée étant donné $x_{new} = \{x_1, x_2, x_j^{new} \ldots x_p\}$. Si i n'est plus

dans la liste $\{i'_1, i'_2, i'_3, \ldots, i'_{max}\}$ alors x_j est une explication contrefactuelle. Si i est encore dans la liste $\{i'_1, i'_2, i'_3, \ldots, i'_{max}\}$ après avoir essayé toutes les possibilités de x_j^{new}, alors les deux premières caractéristiques classées sont examinées, et ainsi de suite, jusqu'à ce que toutes les caractéristiques de I_p soient explorées. Enfin, si l'ensemble des valeurs des caractéristiques originales E est vide, alors cela signifie que cette recommandation ne peut pas être expliquée par l'algorithme 1. Il est à noter qu'il est également possible d'expliquer une liste de recommandations avec notre algorithme 1. Dans ce cas, i devient une liste de recommandations $\{i_1, i_2, i_3, \ldots, i_{max}\}$; dans l'étape 15, la condition devient : $\{i_1, i_2, i_3, \ldots, i_{max}\} \neq \{i'_1, i'_2, i'_3, \ldots, i'_{max}\}$; la différence entre $\{i_1, i_2, i_3, \ldots, i_{max}\}$ et $\{i'_1, i'_2, i'_3, \ldots, i'_{max}\}$ est quantifiée via le coefficient de chevauchement (*Overlap Coefficient*) (Vijaymeena et Kavitha, 2016) entre eux. Nous appliquons l'algorithme 1 pour expliquer un modèle de recommandation contextuelle que nous avons récemment proposé (Zhong et Negre, 2022b). Nous montrons que notre méthode est capable d'expliquer plus de 95% des recommandations, montrant ainsi une grande fidélité du modèle. Par rapport à d'autres méthodes, notre méthode peut réduire le temps de recherche car elle permet de réduire l'espace de recherche. De plus, les explications générées par notre méthode sont fidèles au modèle de recommandation original en identifiant les conditions contextuelles clés. Un autre avantage est que ces explications sont adaptées aux situations contextuelles des utilisateurs, ce qui pourrait être plus convaincant.

En résumé, les contributions de ce travail comprennent : (1) Nous commençons par résumer les travaux de l'état de l'art, nous montrons que les explications dans les SRs doivent être contrastives et sélectionnées ; (2) Nous comparons également LIME et SHAP, cela nous permet de préciser pourquoi nous avons choisi SHAP plutôt que LIME dans ce travail ; (3) Nous proposons ensuite un algorithme capable de générer des explications contrefactuelles dans les SRs à l'aide de SHAP. À notre connaissance, il s'agit du premier article qui combine la méthode de la valeur de Shapley et la méthode contrefactuelle pour générer des explications dans les SRs ; (4) Parallèlement, nous montrons que les explications contrefactuelles générées par notre algorithme sont contrastives et sélectionnées, deux propriétés privilégiées pour les explications en IA ; (5) Enfin, nous présentons une étude de cas pour montrer l'efficacité de notre algorithme. Nos expériences montrent que SHAP, sous certaines conditions, peut réduire le temps nécessaire à la recherche d'explications contrefactuelles.

Références

Kaffes, V., D. Sacharidis, et G. Giannopoulos (2021). Model-agnostic counterfactual explanations of recommendations. In *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*, pp. 280–285.

Kumar, I. E., S. Venkatasubramanian, C. Scheidegger, et S. Friedler (2020). Problems with shapley-value-based explanations as feature importance measures. In *International Conference on Machine Learning*, pp. 5491–5500. PMLR.

Lundberg, S. M. et S.-I. Lee (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems 30*.

Martens, D. et F. Provost (2014). Explaining data-driven document classifications. *MIS quarterly 38*(1), 73–100.

Molnar, C. (2020). *Interpretable machine learning*. Lulu. com.

Ribeiro, M. T., S. Singh, et C. Guestrin (2016). " why should i trust you ?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence 1*(5), 206–215.

Vijaymeena, M. et K. Kavitha (2016). A survey on similarity measures in text mining. *Machine Learning and Applications : An International Journal 3*(2), 19–28.

Visani, G., E. Bagli, F. Chesani, A. Poluzzi, et D. Capuzzo (2020). Statistical stability indices for lime : obtaining reliable explanations for machine learning models. *Journal of the Operational Research Society*, 1–11.

Wachter, S., B. Mittelstadt, et C. Russell (2017). Counterfactual explanations without opening the black box : Automated decisions and the gdpr. *Harv. JL & Tech. 31*, 841.

Zafar, M. R. et N. M. Khan (2019). Dlime : A deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems. *arXiv preprint arXiv :1906.10263*.

Zhang, Y. et X. Chen (2018). Explainable recommendation : A survey and new perspectives. *arXiv preprint arXiv :1804.11192*.

Zhong, J. et E. Negre (2022a). Shap-enhanced counterfactual explanations for recommendations. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pp. 1365–1372.

Zhong, J. et E. Negre (2022b). Towards better representation of context into recommender systems. *International Journal of Knowledge-Based Organizations (IJKBO) 12*(2), 1–12.

Summary

Explanations in recommender systems help users better understand why recommendations are generated, which is crucial for enhancing users' trust and satisfaction. As recommender systems become ever more inscrutable, directly explaining recommender systems sometimes becomes impossible. Post-hoc explanation methods that do not elucidate internal mechanisms of recommender systems are popular approaches. State-of-art post-hoc explanation methods such as SHAP can generate explanations by building simpler surrogate models to approximate the original models. However, directly applying such methods has several concerns: (1) Post-hoc explanations may not be faithful to the original recommender systems since the internal mechanisms of recommender systems are not elucidated; (2) The outputs returned by methods such as SHAP are not trivial for plain users to understand since background mathematical knowledge is required. In this work, we present an explanation method enhanced by SHAP that can generate easily understandable counterfactual explanations with high fidelity.

Algorithme 1 : Generate counterfactual explanations using SHAP

Input : $X = \{x_1, x_2, x_3, \ldots, x_p\}$, information used to generate a list of recommendations ; f, recommender model ; i, a recommendation to be explained

Output : Set of original features values E to be changed, Set of counterfactual feature values $E\_counter$

$E\_to\_expand = \{\}$ % The indices of candidate features
$E = \{\}$
$E\_counter = \{\}$
Stop = False % Indication of termination of algorithm
Generate recommendations $\{i_1, i_2, i_3 \ldots, i_{max}\}$ for user u
Compute the contribution of each features in X using SHAP : $\hat{r}_{ui} = \phi_0 + \phi_1 + \phi_2 + \cdots + \phi_p$
$I_p \subset X : \{I_1, I_2, \ldots\}$, indices of features whose $\phi_{I_p} >= 0$
Sort I_p according to ϕ_{I_p} in a descending manner
for $l \leftarrow 1$ ***to*** *length(I_p)* **do**
    E_to_expand = $I_p[: l]$ % The l first indices
    for $j \in E\_to\_expand$ **do**
        Change x_j to x_j^{new}
        Generate a new list of recommendations $\{i'_1, i'_2, i'_3, \ldots, i'_{max}\}$ given $x_{new} = \{x_1, x_2, x_j^{new}, \ldots, x_p\}$
        if $i \notin \{i'_1, i'_2, i'_3, \ldots, i'_{max}\}$ **then**
            $E = E \cup x_j$
            $E\_counter = E\_counter \cup x_j^{new}$
            Stop = True
            break
    if *Stop = True* **then**
        break
if $i \in \{i'_1, i'_2, i'_3, \ldots, i'_{max}\}$ **then**
    $E = \{\}$
return E, $E\_counter$