PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'Université Paris Dauphine

Generality and explainability in recommender systems

Soutenue par
Jinfeng ZHONG
Le Le 30 Novembre 2023

École doctorale n°543
Ecole Doctorale SDOSE

Spécialité
Informatique

Composition du jury :

Max Chevalier Professeur, Université de Toulouse III	*Président*
Armelle Brun Professeur, Université de Lorraine	*Rapporteur*
Vincent Guigue Professeur, AgroParisTech, Université Paris-Saclay	*Rapporteur*
Nicolas Labroche Maître de Conférences, HDR, Université de Tours	*Examinateur*
Florian Yger Maître de Conférences, Université Paris Dauphine-PSL	*Examinateur*
Elsa Negre Maître de Conférences, HDR, Université Paris Dauphine-PSL	*Directrice de thèse*

Dauphine | PSL UNIVERSITÉ PARIS

Acknowledgement

Firstly, I would like to express my profound gratitude to my esteemed supervisor, Dr. Elsa NEGRE. Her unwavering support, nurturing encouragement, and unparalleled expertise have served as the bedrock upon which this thesis stands. Dr. NEGRE's remarkable patience and depth of knowledge have not only facilitated the fruition of this academic endeavor but have also enriched this scholarly journey into an invaluable life experience I will treasure forever.

I extend my heartfelt thanks to the distinguished committee members who have significantly contributed to the refinement of this research. Special acknowledgment is due to Prof. Armelle Brun from Université Lorraine and Prof. Vincent Guigue from AgroParisTech for their astute critiques and insightful recommendations. Your invaluable feedback has immeasurably shaped the trajectory and integrity of this research. I am equally indebted to Prof. Max Chevalier from IRIT, Dr. Nicolas Labroche from Université Tour, and Dr. Florian Yger from Université Paris Dauphine. Your interest in my work and willingness to serve on the jury is highly appreciated.

Further acknowledgments go to Pr. Camille Rosenthal-Sabroux and Dr. Michel Grundstein. Our in-depth discussions on the intricacies of "interpretation" and "explanation" have been instrumental in conceptualizing the framework for this thesis. Similarly, I am grateful to Dr. Pierre-Emmanuel Arduin for generously sharing pivotal documents on knowledge management, thereby enriching the content and direction of this thesis.

I am delighted to express my gratitude to my collaborators, Pr. Marie-Hélène Abel and Dr. LE Ngoc Luyen. Our collaboration has broadened my academic horizons and provided a fertile ground for potential future applications of my research. A particular nod to Pr. Marie-Hélène Abel for her proficiency in ontology engineering and academic mentorship, and to Dr. LE Ngoc Luyen for our stimulating dialogues and brainstorming sessions.

Additionally, I want to thank Dr. Denis Cornaz and Dr. Brice Mayag for entrusting me with teaching responsibilities. This experience has augmented my research perspective, lending it a unique, pedagogical dimension.

A warm acknowledgment goes out to my fellow Ph.D. students at LAMSADE. The camaraderie, intellectual debates, and the nurturing environment you have provided have been invigorating.

On a more personal note, I owe an enormous debt of gratitude to my ever-supportive girlfriend, Xinyi XU. Her unflinching support, boundless love, and enduring patience have been my anchor and emotional pillar throughout this arduous journey.

Lastly, my thanks are reserved for my family, particularly my parents and my grandparents. No words can adequately capture my gratitude for your relentless love, sacrifices, and unwavering belief in me. You have been my foremost teachers, lifelong role models, and the bedrock of my existence. This achievement is as much a testament to your upbringing as it is a milestone in my academic journey.

Résumé

Les systèmes de recommandation (SRs) sont devenus une composante essentielle du web intelligent à l'ère d'Internet, trouvant des applications dans de nombreux systèmes pratiques. Alors que les SRs sont de plus en plus intégrés dans la vie quotidienne des gens, ils sont devenus des outils essentiels d'aide à la décision ayant un impact significatif sur le processus décisionnel des utilisateurs (Lu et al., 2015, Ricci et al., 2015). Par conséquent, une demande croissante des utilisateurs émerge pour comprendre la logique derrière les recommandations, en particulier dans des domaines à haut risque tels que la médecine et la finance, afin de réduire les risques. Les SRs explicables, qui fournissent des recommandations personnalisées et offrent des explications pour justifier ces recommandations, ont attiré une attention significative à la fois de l'industrie et du monde académique (Zhang et al., 2020, Nunes and Jannach, 2017, Daher et al., 2017). L'exemple 0.1 est une explication typique dans un scénario de recommandation de films. Il a été reconnu que les explications sont cruciales pour améliorer l'efficacité, l'efficience, la persuasivité, la transparence, la confiance et la satisfaction des SRs (Tintarev and Masthoff, 2015). Malgré l'émergence de diverses méthodes d'explication, peu de recherches ont été menées sur la généricité des SRs explicables. Dans cette thèse, nous contribuons à ce sujet en proposant un cadre explicatif général et en l'instanciant sous deux scénarios d'explication: les explications intrinsèques au modèle et les explications post-hoc. L'objectif de ce chapitre est de fournir le contexte de recherche, de décrire les principaux problèmes et défis de recherche, leur importance pratique et valeur scientifique, de décrire les principales contributions de cette thèse, et d'en présenter la structure.

Example 0.1 *Nous vous recommandons ce film car c'est un film de Tom Hanks et vous avez auparavant attribué de bonnes notes à ses films.*

Antécédents

D'une perspective plus large, l'Intelligence Artificielle (IA) est définie comme un domaine de l'informatique qui se concentre sur la conception, le développement et la mise en œuvre de machines intelligentes capables d'émuler et d'exécuter des tâches qui nécessitent généralement une intelligence humaine (Russell and Norvig, 2016). Ces tâches comprennent, entre autres, la résolution de problèmes, la prise de décision, l'apprentissage, la perception et le traitement du langage naturel. L'IA a rapidement évolué pour devenir une technologie centrale avec une large gamme d'applications dans divers domaines du monde réel, tels que la santé, la finance, la recommandation et la communication (Russell and Norvig, 2016). Ses avancées ont contribué à des améliorations et innovations significatives, révolutionnant les industries et impactant la société de nombreuses manières. D'une part, l'IA a connu un succès remarquable grâce à l'utilisation de modèles complexes tels que les réseaux neuronaux profonds (Liu et al., 2017) ou les modèles d'ensemble (Sagi and Rokach, 2018), qui ont la capacité d'effectuer des tâches complexes. D'autre part, le déploiement de tels modèles opaques avec des milliers, voire des millions, de paramètres peut être à double tranchant. La complexité des systèmes d'IA rend difficile la justification des résultats qu'ils génèrent, et sans explications convaincantes, il se peut que ces systèmes ne soit pas vraiment utile pour aider les humains à prendre de meilleures décisions (Zhu et al., 2018). De nombreux chercheurs ont souligné le rôle crucial de la recherche dans la formulation d'explications visant à promouvoir une IA responsable. Lipton (2018) a introduit les critères clés pour la recherche centrée sur la génération d'explications, comprenant la confiance, la causalité, la transférabilité, l'informativité, et la prise

de décision équitable et éthique. S'appuyant sur ces principes, Arrieta et al. (2020) a approfondi la logique derrière la recherche sur les explications en IA, mettant l'accent sur la promotion de la confiance, de la causalité, de la transférabilité, de l'informativité, de la confiance, de l'équité, de l'accessibilité, de l'interactivité et de la conscience de la vie privée. D'un point de vue pratique, Hall and Gill (2019) a avancé que l'élan pour la recherche en explication peut être efficacement condensé en deux catégories: intellectuelle et sociale.

Motivé par les raisons susmentionnées, une attention significative a été accordée tant par le monde académique que par l'industrie au développement de systèmes d'IA capables de générer des explications pour justifier leurs résultats (Arrieta et al., 2020). Selon Lipton (2018), deux stratégies peuvent être utilisées pour expliquer les sorties d'un système d'IA : (1) L'utilisation de modèles transparents, dont le fonctionnement interne peut être facilement expliqué (explications intrinsèques au modèle), tels que les modèles linéaires simples et les règles d'association; (2) La fourniture d'explications a posteriori sans élucider précisément comment les modèles fonctionnent. Le terme *eXplainable Artificial Ingelligence* (XAI) a rapidement gagné en popularité après que la DARPA (Defense Advanced Research Projects Agency) ait lancé un appel à propositions de recherche sur l'explicabilité de l'IA. Il se réfère aux technologies de l'IA capables de fournir des justifications pour leur processus de prise de décision. La demande croissante des utilisateurs et des réglementations telles que le Règlement Général sur la Protection des Données (RGPD) (Voigt and Von dem Bussche, 2017) a suscité un intérêt de recherche significatif pour l'XAI ces dernières années. Cependant, il n'y a pas de consensus sur les définitions explicites de l'XAI et de l'intelligence artificielle interprétable, certains travaux les utilisant de manière interchangeable, tandis que d'autres les distinguent. Souvent, ces distinctions sont basées sur les intuitions subjectives des chercheurs sans soutien théorique solide (Miller, 2019). L'utilisation mal définie et ambiguë des termes "interprétabilité" et "explicabilité" a rendu difficile la formulation de problèmes pour définir, concevoir et évaluer les systèmes d'IA pouvant fournir des explications pour leurs résultats. Pour faire progresser le domaine, il est crucial de formaliser et de clarifier ces termes (Doshi-Velez and Kim, 2017).

Les SRs sont fréquemment utilisés comme outils d'aide à la décision, leur rôle étant d'atténuer l'impact de la surcharge d'informations en fournissant des suggestions pertinentes. Dans des situations où la prise de décision a des implications importantes, le besoin d'explications justifiables soutenant ces recommandations s'intensifie. Parallèlement, dans le domaine des SRs, une sous-division notable de l'IA, il y a eu un intérêt croissant pour le développement de SRs dotés de la capacité de fournir des explications pour leurs recommandations. Cependant, à l'instar des défis dans le domaine de l'IA, les termes "recommandations interprétables" et "recommandations explicables" manquent de définitions claires, rendant difficile la comparaison et la généralisation des méthodes existantes pour fournir des explications. En conséquence, il y a un besoin urgent d'un cadre général pour développer des SRs capables de générer des explications pour les recommandations, malgré les nombreuses recherches menées dans ce domaine.

Motivations de recherche

Comme exploré précédemment, il y a eu une forte augmentation des efforts de recherche axés sur le développement de systèmes IA capables de fournir des explications pour leurs résultats. Les SRs, un sous-ensemble essentiel de l'IA, ont connu une croissance correspondante dans les efforts visant à les doter de facultés explicatives. Cependant, malgré l'attention considérable attirée par ce domaine, il subsiste un manque notable de consensus concernant les définitions de termes

fréquemment utilisés tels que “explicabilité” et “interprétabilité”. Cette ambiguïté pose des défis en matière de formalisation, de comparaison et d’évaluation des différentes méthodes d’explication. Par conséquent, il est essentiel d’établir des définitions claires pour ces termes avant de se plonger dans l’étude. En d’autres termes, un cadre général capable de clarifier des termes fréquemment utilisés, mais vaguement définis, en accord avec l’aspect “généricité” mentionné dans le titre de cette thèse. À cet égard, la première question de recherche que nous souhaitons aborder est:

RQ1, “Quelles sont les différences entre ‘interprétabilité’ et ‘explicabilité’ dans le contexte de l’IA, et plus spécifiquement dans le contexte des SRs?”

Pour répondre pleinement à cette question, les deux points suivants doivent être clarifiés:

- **RQ1.1, “Que signifient respectivement ‘explicabilité’ et ‘interprétabilité’?”** Les définitions existantes de ces deux termes sont généralement basées sur des intuitions subjectives des chercheurs sans un solide soutien théorique (Miller, 2019), ce qui rend difficile la formulation de problèmes pour définir, concevoir et évaluer des systèmes IA pouvant fournir des explications pour leurs résultats.

- **RQ1.2, “À qui s’adressent ‘explicabilité’ et ‘interprétabilité’?”** Il est essentiel de déterminer le public cible pour ces explications (Vilone and Longo, 2021, Arrieta et al., 2020). Les explications destinées aux développeurs de modèles, aux experts du domaine et aux entités/agences réglementaires diffèrent considérablement de celles destinées aux utilisateurs non experts.

En résumé, notre premier objectif est de clarifier et de fixer la signification des termes “explicabilité” et “interprétabilité” afin de faciliter la comparaison et l’évaluation des différentes méthodes d’explication dans le contexte de l’IA, en particulier au sein des SRs. Une fois que nous aurons établi l’utilisation des termes “explicabilité” et “interprétabilité” dans les SRs, il serait pertinent de préciser :

RQ2, “Pourquoi ‘explicabilité’ et ‘interprétabilité’ sont-elles particulièrement demandées dans les SRs?”

Nous détaillerons davantage les objectifs d’explication des recommandations formulées par les SRs grâce à une revue approfondie de la littérature dans la Section 3.7: promouvoir l’efficacité, la pertinence, le pouvoir de persuasion, la transparence, la confiance, la satisfaction et la scrutabilité des SRs (Tintarev and Masthoff, 2015). Notez que ces objectifs sont également utilisés pour évaluer les explications dans les SRs (Gedikli et al., 2014). Par la suite, nous arrivons à la question de recherche centrale de cette thèse:

RQ3, “Comment expliquer les recommandations?”

À l’instar du domaine de l’IA, les explications dans les SRs, qui constituent une branche importante de l’IA, sont classées en deux principales catégories (Zhang et al., 2020): (1) adopter des modèles transparents; et (2) adopter des techniques d’explication post-hoc pour générer des explications post-hoc. Les modèles transparents, tels que la régression linéaire et les règles d’association, sont faciles à comprendre. Cependant, ils souffrent souvent d’une moindre précision de prédiction, ce qui est connu comme le compromis entre complexité et précision (Doshi-Velez and Kim, 2017). D’autre part, les techniques d’explication post-hoc, telles que SHAP (Lundberg and Lee, 2017) et LIME (Local Interpretable Model-agnostic explanations) (Ribeiro et al., 2016), peuvent être appliquées pour expliquer n’importe quel modèle, offrant ainsi plus de flexibilité. Cependant, une préoccupation majeure avec l’application de ces techniques est que les explications post-hoc ne

représentent pas toujours fidèlement les SRs originaux (Rudin, 2019). Par conséquent, les deux stratégies ont leurs mérites, et ce n'est pas une situation binaire. Cela nous amène naturellement à nous demander:

- **RQ3.1, "Comment construire des modèles transparents capables d'expliquer les recommandations tout en garantissant la précision de ces recommandations?"** Construire directement des SRs capables de générer des recommandations précises et d'expliquer ces recommandations est un sujet important. Les SRs existants reposent souvent sur des modèles à facteurs latents tels que la factorisation matricielle (Koren et al., 2009), la machine de factorisation (Rendle, 2010), les méthodes basées sur l'apprentissage profond (Liu and Wu, 2017), etc. Ces modèles peuvent faire des prédictions de notation précises. Néanmoins, la sémantique explicite des facteurs latents appris reste ambiguë, compliquant ainsi la tâche de générer des explications cohérentes et convaincantes pour les recommandations.

- **QR3.2, "Comment expliquer les SRs complexes existants à l'aide de techniques d'explication post-hoc tout en atténuant les limites de ces techniques?"** Conserver les SRs complexes existants en production et adopter des techniques d'explication post-hoc (Lipton, 2018) pour expliquer les recommandations est pratique. Cela permet de réutiliser les SRs existants, de réduire les coûts d'ingénierie et ces SRs obtiennent généralement de meilleures performances que ceux qui adoptent des modèles simples. SHAP (Lundberg and Lee, 2017) et LIME (Ribeiro et al., 2016) sont parmi les outils les plus en vue qui sont capable de fournir des explications post-hoc dans la littérature. Certains travaux (Guo et al., 2021, Singh and Anand, 2019, Nóbrega and Marinho, 2019) expliquent les recommandations en appliquant directement l'importance de chaque caractéristique calculée par LIME et SHAP. Les principales limites et préoccupations de l'application directe de ces méthodes sont : (1) Il se peut que les explications post-hoc ne soient pas fidèles aux SR originaux et parfois ces explications sont même erronées (Rudin, 2019) ; (2) Pour les utilisateurs ordinaires, comprendre l'importance des caractéristiques retournées par de telles méthodes n'est pas évident car des connaissances mathématiques complexes sont nécessaires. Par exemple, les sorties de SHAP incluent la figure "force_plot", cela nécessite que les utilisateurs aient des connaissances de base en théorie des jeux. Par conséquent, SHAP et LIME sont de bons outils pour que les développeurs de modèles visualisent les comportements des modèles et peuvent les aider à déboguer les modèles. Cependant, pour les utilisateurs ordinaires, il se peut que ces figures ne soient pas faciles à comprendre. La question cruciale qui se pose alors est: Comment pouvons-nous exploiter les méthodes d'explication post-hoc pour élucider n'importe quel modèle de recommandation tout en évitant ces contraintes ?

Différentes méthodes d'explication ont été proposées dans le domaine des SRs, mais il reste incertain quelle est la meilleure façon de fournir des explications dans un contexte donné pour une tâche donnée (Zhou et al., 2021). Cela nous amène à notre dernière question de recherche :

QR4, "Comment évaluer les explications dans les SRs?"

Évaluer les explications est difficile car les explications sont centrées sur l'humain (Miller, 2019), et les explications de référencent pour une tâche d'apprentissage automatique est souvent inconnue (Molnar et al., 2020), ce qui est également le cas pour les SRs. Pour répondre à cette question, nous évaluons la qualité des explications dans les SRs en mesurant dans quelle mesure les explications satisfont aux objectifs identifiés par Gedikli et al. (2014), Tintarev and Masthoff (2015).

Résumé des contributions

Dans cette thèse, nous apportons des contributions aux domaines de la généricité et de l'explicabilité dans les SRs en abordant les aspects suivants: (1) nous proposons un *Schéma Interpréter/Expliquer* (SIE) qui permet de désambiguïser les termes essentiels utilisés en IA, notamment dans les SRs, pour éviter la confusion et clarifier leurs significations. Sur la base du SIE, nous construisons un *Cadre Général pour la Génération d'Explications* (CGGE) qui peut intégrer les méthodes existantes de génération d'explications dans le contexte de l'IA; (2) nous résumons les motivations pour la recherche liée aux explications en IA et fournissons un aperçu des méthodes d'évaluation des explications dans les SRs; (3) nous instancions le cadre avec deux scénarios d'explication dans les SRs pour démontrer son efficacité et son applicabilité; (4) nous menons une étude préliminaire auprès des utilisateurs pour évaluer la qualité des explications.

Désambiguïsation de l'utilisation des termes (QR1): Nous examinons en profondeur et comparons les travaux existants sur l'utilisation de l'« explicabilité » et de l'« interprétabilité » dans le contexte de l'IA. Il est essentiel de noter les aspects sociaux des explications, car elles impliquent le transfert de connaissances par le biais de conversations ou d'interactions (Miller, 2019, Malle, 2006). Pour éclairer ce point, nous nous appuyons sur des concepts de gestion des connaissances (Arduin et al., 2015), où les humains possèdent un schéma d'interprétation (Jones et al., 2011) (également connu sous le nom de modèle mental (Tsuchiya, 1993)) qui leur permet d'interagir avec leur environnement par « sense-reading » et « sense-giving » (Polanyi, 1967). À partir de cela, nous proposons le SIE (Zhong and Negre, 2021) dans les systèmes d'IA. Nous proposons la définition de l'« explicabilité » et de l'« interprétabilité » basée sur le SIE et précisons davantage ce que signifient respectivement « explicabilité » et « interprétabilité »; à qui s'adressent l'« explicabilité » et l'« interprétabilité ». Avec les définitions proposées, la formulation de problèmes tels que la définition, la conception et l'évaluation de l'IA explicable peut être sérieusement abordée, ce qui rendra le système d'IA plus transparent lors de la prise de décisions. Avec le SIE et nos définitions, nous construisons le CGGE dans le contexte de l'IA. Dans notre cadre, les deux stratégies (Lipton, 2018) (Utilisation de modèles transparents et fourniture d'explications post-hoc) appliquées pour expliquer la sortie des systèmes d'IA peuvent être intégrées ensemble. Notez que le cadre s'applique également aux SRs, car les SRs sont une branche importante de la recherche en IA, et nous instancierons le cadre en appliquant les deux stratégies dans le contexte des SRs.

Motivations pour la recherche dans le domaine (QR2, QR4): Nous passons en revue en profondeur la littérature existante sur les explications en IA et les classons en trois motivations (Zhong and Negre, 2021): (1) La motivation commerciale qui provient des demandes des utilisateurs; (2) La motivation réglementaire qui vient des exigences réglementaires; (3) La motivation technique qui concerne le débogage et l'amélioration des systèmes d'IA existants. Nous résumons également les objectifs des explications dans les SRs : aider les utilisateurs à comprendre comment le système fonctionne (transparence), augmenter la confiance des utilisateurs dans le système (confiance), aider les utilisateurs à prendre de meilleures décisions (efficacité) plus rapidement (efficience), persuader l'utilisateur d'essayer ou de consommer (persuasivité), permettre aux utilisateurs d'indiquer au système quand les recommandations ne sont pas appropriées (scrutabilité) et augmenter la facilité d'utilisation ou le plaisir (satisfaction). Notez que ces objectifs sont souvent utilisés pour évaluer la qualité des explications dans les SRs (Gedikli et al., 2014, Tintarev and Masthoff, 2015).

Deux scénarios d'explication (QR3): Nous instancions le CGGE en appliquant les deux stratégies d'explication:

- **Pour aborder la question du rôle de la transparence (QR3.1)**: nous nous tournons vers les Cadres d'Argumentation (CAs), qui ont été utilisés pour améliorer l'explicabilité de l'IA (Vassiliades et al., 2021, Čyras et al., 2021). Les SAs peuvent représenter le processus de prise de décision de manière graphique, et les propriétés prédéfinies aident à raisonner comment parvenir aux meilleures décisions (Vassiliades et al., 2021). Le CA comprend également des moyens de définir des arguments pondérés et des relations dialectiques entre les arguments, qui contiennent des sémantiques pour décider de la force et de l'acceptabilité de ces arguments (Čyras et al., 2021).

  - **Sans contexte spécifique**: Nous proposons le modèle **A**ttribute-**A**ware **A**rgumentative **R**ecommender (A^3R) (Zhong and Negre, 2022a), une méthode qui combine les techniques basées sur la factorisation et les SAs. Grâce à l'aide des SAs, chaque étape de A^3R est dotée d'une sémantique explicite, permettant à A^3R de générer des explications pour les recommandations qui sont faciles à comprendre. Pour chaque interaction utilisateur-article, nous définissons un SA, où les caractéristiques des articles sont considérées comme des arguments, et les évaluations des utilisateurs envers ces caractéristiques sont définies comme la force des arguments. Des explications intrinsèques au modèle peuvent être générées pour chaque interaction utilisateur-article grâce aux SAs.
  - **Sensible au contexte**: Une croyance dominante dans le domaine des SRs est que le contexte des utilisateurs peut influencer leurs préférences (Adomavicius and Tuzhilin, 2011). Nous pensons que cela s'applique également aux SAs. Certains arguments « bons » dans un contexte particulier deviennt moins précis dans un autre contexte. Cependant, A^3R ne prend pas en compte les contextes des utilisateurs qui peuvent fortement influencer leurs préférences. Nous étendons cette idée en prenant en compte le contexte des utilisateurs: **C**ontext-**A**ware **F**eature **A**ttribution **T**hrough **A**rgumentation) (Zhong and Negre, 2023). Les caractéristiques de $CA-FATA$ incluent: (1) $CA-FATA$ combine des modèles de facteurs latents et le SA, donc chaque étape de calcul est dotée d'une sémantique explicite, ce qui aide à expliquer les recommandations générées; (2) Puisque les contextes des utilisateurs sont pris en compte, le CA correspondant est également sensible au contexte, ce qui signifie que $CA-FATA$ peut générer des explications adaptées au contexte (Zhang et al., 2020, Zhong and Negre, 2022e). Ainsi, $CA-FATA$ peut générer et expliquer des recommandations de manière argumentative et sensible au contexte.

- **Pour répondre à la question de l'utilisation des explications post-hoc (RQ3.2)**: nous nous inspirons des recherches connexes en psychologie, philosophie et sciences cognitives. Au lieu de demander directement pourquoi un événement P s'est produit, les gens préfèrent souvent demander pourquoi l'événement P s'est produit plutôt qu'un événement Q, où Q est une alternative (Lipton, 1990, Lim and Dey, 2009, Mackie, 1980). Générer des explications contrefactuelles (Wachter et al., 2017) qui sont facilement compréhensibles (Wachter et al., 2017) et de haute fidélité (Kaffes et al., 2021) est une solution potentielle. Contrairement à LIME (Ribeiro et al., 2016) et SHAP (Lundberg and Lee, 2017), qui approximent les modèles originaux, les explications contrefactuelles cherchent les changements minimaux nécessaires

pour modifier les sorties des modèles originaux. Par conséquent, les explications contrefactuelles ont une haute fidélité (Kaffes et al., 2021). Une explication contrefactuelle typique décrit une situation causale comme "Si A ne s'était pas produit, alors B ne se serait pas produit' (Molnar, 2020). Nous avons proposé **S**hap-**E**nhanced **C**ounterfactual **E**xplanation (SECE) (Zhong and Negre, 2022c) et montré que SHAP (Lundberg and Lee, 2017) peut être adapté pour générer des explications contrefactuelles qui sont contrastives et sélectionnées (Zhong and Negre, 2022c). À notre connaissance, il s'agit du premier travail qui combine la méthode de la valeur de Shapley et la méthode contrefactuelle pour aborder les explications dans les SRs: nous appliquons SHAP (Lundberg and Lee, 2017) pour sélectionner les candidats possibles d'explications contrefactuelles. Notre méthode peut générer des explications contrefactuelles fidèles et facilement compréhensibles.

La qualité de l'explication (RQ4): Nous résumons également les proxies prédéfinis couramment utilisés pour évaluer les explications dans les SRs (Zhong and Negre, 2022e). En même temps, étant donné que l'explication est centrée sur l'humain, nous menons une étude utilisateur (Zhong and Negre, 2022b) pour évaluer la qualité des explications générées par nos méthodes d'explication: dans quelle mesure les explications satisfont les objectifs des explications: transparence, confiance, efficacité, efficience, scrutabilité, satisfaction, persuasivité (Gedikli et al., 2014, Tintarev and Masthoff, 2015).

Guide de lecture

Le reste de cette thèse est organisé comme suit :

- Dans le Chapitre 2, nous présentons l'état de l'art lié à l'explication du point de vue de l'IA. Nous couvrons les concepts fondamentaux des explications en IA, comment les explications en IA ont pu être influencées par la recherche d'autres domaines tels que la philosophie, la psychologie, la science cognitive, comment générer des explications et comment les évaluer.

- Dans le Chapitre 3, nous passons en revue les travaux liés aux recommandations explicables en appliquant les notions introduites dans le Chapitre 2.

- Dans le Chapitre 4, nous décrivons SIE, qui permet de lever l'ambiguïté des termes: « explicabilité » et « interprétabilité ». Sur la base de SIE, nous construisons le CGCE. Nous montrons que les méthodes existantes pour générer les explications peuvent être intégrées dans ce cadre.

- Dans le Chapitre 5, nous présentons quelques bases techniques qui seront utilisées pour instancier le CGCE dans les Chapitres 6, 7 et 8.

- Dans le Chapitre 6, nous instancions le cadre général dans le scénario d'explication intrinsèque au modèle, où les recommandations peuvent être expliquées intrinsèquement par la conception. En commençant par un scénario sans contexte, nous introduisons A^3R qui utilise des cadres d'argumentation pour fournir à chaque calcul une sémantique explicite. Nos évaluations indiquent que A^3R surpasse de manière significative les modèles de recommandation basés sur l'argumentation existants et rivalise efficacement avec d'autres modèles de recommandation plus opaques et en boîte noire.

- Dans le Chapitre 7, nous développons davantage le modèle introduit dans le Chapitre 6 en intégrant le contexte des utilisateurs dans un nouveau modèle appelé $CA - FATA$. Comme ce modèle prend en compte le contexte des utilisateurs, le CA qu'il utilise devient également sensible au contexte. Cela signifie que $CA - FATA$ est capable de générer des explications sensibles au contexte. Par conséquent, $CA - FATA$ peut à la fois générer des recommandations et fournir des explications pour celles-ci de manière argumentative et consciente du contexte.
- Dans le Chapitre 8, nous instancions le cadre général dans le scénario d'explication post-hoc, où les modèles en boîte noire sont conservés et les explications post-hoc sont générées. Plus spécifiquement, nous utilisons SHAP pour sélectionner les candidats possibles aux explications contrefactuelles.
- Dans le Chapitre 9, nous comparons les deux scénarios d'explication et présentons le résultat d'une étude utilisateur qui évalue différents types d'explications.
- Dans le Chapitre 10, nous concluons la thèse, proposons des pistes pour des travaux futurs et énumérons les publications au cours du programme de doctorat.

Contents

I Introduction and related work **19**

1 Introduction **21**
1.1 Background . . . 21
1.2 Research motivations . . . 23
1.3 Summary of contributions . . . 25
1.4 Guide to read . . . 27

2 Explanations in AI **29**
2.1 A general view . . . 29
2.2 To explain or to interpret? . . . 30
2.3 Explanation: what? . . . 31
2.4 Explanation: why? . . . 33
2.5 Explanation: how? . . . 35
2.5.1 Transparent models: model-intrinsic explanations . . . 35
2.5.2 Post-hoc explanations . . . 36
2.6 Explanation: how to evaluate? . . . 37
2.7 Conclusions . . . 38

3 Explainable recommendations **41**
3.1 Recommender systems . . . 42
3.1.1 Recommendation problem . . . 42
3.1.2 Evaluating recommendations . . . 43
3.1.2.1 Evaluation strategies . . . 43
3.1.2.2 Evaluation criteria and metrics . . . 44
3.1.3 Recommendation approaches . . . 46
3.2 Explainable recommendations: an overview . . . 47
3.3 Content-based recommender systems . . . 48
3.3.1 CBRSs: generating recommendations . . . 48
3.3.2 CBRSs: explaining recommendations . . . 48
3.3.3 Summary of CBRSs . . . 49
3.4 Collaborative filtering (CF) . . . 50
3.4.1 Neighborhood approaches . . . 50
3.4.1.1 Neighborhood approaches: generating recommendations . . . 50
3.4.1.2 Neighborhood approaches: explaining recommendations . . . 50
3.4.1.3 Summary of Neighborhood approaches . . . 51

3.4.2 Latent factor models . . . 51
3.4.2.1 Latent factor models: generating recommendations . . . 51
3.4.2.2 Latent factor models: explaining recommendations . . . 52
3.4.2.3 Summary of latent factor models . . . 53
3.5 Hybrid approaches . . . 54
3.5.1 Hybrid approaches: generating recommendations . . . 54
3.5.2 Hybrid approaches: explaining recommendations . . . 55
3.5.3 Summary of hybrid approaches . . . 55
3.6 Context-aware recommender system . . . 55
3.6.1 CARSs: generating recommendations . . . 55
3.6.1.1 What is context? . . . 56
3.6.1.2 Paradigms for incorporating context . . . 56
3.6.2 CARSs: explaining recommendations . . . 57
3.6.3 Summary of CARSs . . . 58
3.7 Evaluating explanations in RSs . . . 58
3.8 Conclusions . . . 61

II Theoretical contribution: a general framework for generating explanations 63

4 General framework for generating explanations 65
4.1 The importance of generality . . . 65
4.2 A general definition . . . 66
4.3 *Interpret/Explain Schema* in AI system . . . 68
4.3.1 AI: to interpret or to explain? . . . 68
4.3.2 Definitions in the context of AI . . . 69
4.3.3 Discussions . . . 70
4.3.4 *General framework for generating explanations* . . . 72
4.4 Conclusions . . . 75

III Instantiations of the general framework for generating explanations 77

5 Preliminary backgrounds 81
5.1 Introduction . . . 82
5.2 Feature attribution . . . 83
5.3 Argumentation . . . 84
5.3.1 Argumentation frameworks (AFs) . . . 84
5.3.2 Weak balance and weak monotonicity . . . 86
5.4 LIME and SHAP . . . 87
5.5 Relation to our work . . . 89

6 Model-intrinsic instantiation: Attribute-aware argumentative recommender 91
6.1 Introduction . . . 92
6.2 Recommendations through argumentation . . . 93

6.3 Our model . . . 94
6.3.1 Problem setting . . . 94
6.3.2 Main steps of A^3R . . . 96
6.4 Explaining recommendations using A^3R . . . 98
6.4.1 Argumentation scaffolding . . . 98
6.4.1.1 Argumentation setting . . . 98
6.4.1.2 Proofs . . . 99
6.4.2 Explanation applications . . . 100
6.4.2.1 Toy explanation templates . . . 101
6.4.2.2 Interactive RSs . . . 102
6.4.2.3 Contrastive explanations . . . 103
6.4.3 Relations with other models . . . 104
6.5 Generating recommendations using A^3R . . . 105
6.5.1 Experiments setting . . . 106
6.5.2 Experiment results . . . 107
6.5.3 Abalation study . . . 109
6.5.4 Case study . . . 110
6.6 Conclusions . . . 112

7 Model-intrinsic instantiation: Context-aware feature attribution through argumentation 113
7.1 Introduction . . . 114
7.2 Our model . . . 115
7.2.1 Problem Setting . . . 115
7.2.2 Main steps of CA-FATA . . . 116
7.3 Explaining recommendations using CA-FATA . . . 120
7.3.1 Argumentation scaffolding . . . 120
7.3.1.1 Argumentation setting . . . 120
7.3.1.2 Proofs . . . 121
7.3.2 Explanation applications . . . 122
7.3.3 Relations with other models . . . 124
7.4 Generating recommendations using CA-FATA . . . 124
7.4.1 Experiments setting . . . 124
7.4.2 Experiment results . . . 126
7.4.3 Abalation study . . . 128
7.4.4 Case study . . . 129
7.5 Conclusions . . . 130

8 Post-hoc instantiation: Shap-enhanced counterfactual explanations for recommendations 133
8.1 Introduction . . . 134
8.2 Motivations . . . 135
8.2.1 Desired properties of explanations . . . 135
8.2.2 Counterfactual explanations . . . 136
8.3 Shap-Enhanced Counterfactual Explanations . . . 138
8.3.1 Problem formulation . . . 138

8.3.2 Main steps of the framework . . . 138
8.3.3 Complexity analysis . . . 141
8.4 Experiments . . . 141
8.4.1 Dataset and recommender model . . . 141
8.4.2 Evaluation settings . . . 142
8.4.3 Experiment results . . . 143
8.4.3.1 Time required for searching . . . 143
8.4.3.2 Success rate . . . 143
8.4.3.3 Length of explanations . . . 144
8.4.3.4 Instance-level vs list-level . . . 144
8.4.3.5 Parameter comparison . . . 145
8.5 Conclusions . . . 145

9 Comparison and evaluation of explanation methods 147
9.1 Comparison . . . 148
9.1.1 Model-intrinsic vs Post-hoc . . . 148
9.1.2 Local vs Global . . . 150
9.1.3 Model-specific vs Model-agnostic . . . 151
9.2 Evaluation of explanations . . . 151
9.2.1 Steps and experiment setup . . . 152
9.2.1.1 Major steps . . . 152
9.2.1.2 Baselines . . . 153
9.2.2 Results of user study . . . 154
9.2.2.1 Objective evaluation . . . 154
9.2.2.2 Subjective evaluation . . . 155
9.2.2.3 Correlation between different metrics . . . 156
9.2.2.4 Aggregating metrics . . . 156
9.2.3 Discussions . . . 158
9.3 Conclusions . . . 159

IV Concluding remarks 161

10 Conclusions and perspectives 163
10.1 Conclusions . . . 163
10.2 Perspectives for future work . . . 166
10.2.1 Improvements of our work . . . 166
10.2.2 Relations to other research topics in RSs . . . 167
10.3 List of publications . . . 168

Appendices 171

A The black-box model we have developed 173
A.1 More detail on the black-box recommender in Section 8.4.1 . . . 173

List of Figures

2.1 Relationships of *interpretability* and *explainability* in state-of-art works 31
2.2 *Sense-giving* and *sense-reading* in knowledge management. 32

3.1 The core idea of Vanilla MF. 52
3.2 Three paradigms for incorporating contexts into RSs 56

4.1 Interpret/Explain in AI systems 69
4.2 Explainable AI system, Interpretable AI system, Post-hoc interpretable AI system . 71

5.1 A demo of LIME (Ribeiro et al., 2016) 88

6.1 A graphical representation of the TAF corresponding to A^3R 95
6.2 A toy example showing how A^3R works. 95
6.3 The computation steps of A^3R 96
6.4 Interaction process between users and A^3R 102
6.5 An example of the TAF taken from A-I 104
6.6 A subgraph of Figure 6.5 104
6.7 Case study on **Netflix** 111

7.1 A graphical representation of an argumentation procedure 117
7.2 Illustration of the steps of CA-FATA. 118
7.3 A case study on Frappé dataset 130

9.1 Protocol of user study 153
9.2 A screenshot of the Likert-scale questionnaire 157

List of Tables

2.1 Potential goals of explanations in RSs . . . 34

3.1 Possible results of top-K recommendations . . . 46
3.2 Evaluation approaches for explanations in RSs . . . 62

4.1 Key notions and their target audience . . . 75

5.1 The key notions in Chapter 5 . . . 85
5.2 Comparison between LIME and SHAP . . . 89

6.1 The key notions in Chapter 6 . . . 94
6.2 Several possible explanation scenarios of A^3R . . . 101
6.3 Comparison of $MF, FM, NFM, AFM, A-I, A^3R$ . . . 105
6.4 Detailed statistics of the datasets applied in this chapter . . . 106
6.5 Rating prediction on four datasets . . . 108
6.6 Comparison between A^3R and baselines on precision, recall and f1 score . . . 108
6.7 Rating prediction on four datasets. . . . 109

7.1 The key notions in Chapter 7 . . . 116
7.2 Explanation templates under contextual situation *cs* . . . 123
7.3 Comparison of $A-I, A^3R$ and Ca-FATA . . . 124
7.4 Basic statistics of the two datasets . . . 125
7.5 Comparison between CA-FATA and baselines on RMSE and MAE . . . 126
7.6 Comparison between CA-FATA and baselines on precision, recall and f1 score. . . . 128
7.7 Performances of the variants of CA-FATA . . . 128
7.8 Performances of the variants of CA-FATA . . . 129

8.1 The key notions in this chapter . . . 136
8.2 Details of selected contextual factors in CoMoDa . . . 141
8.3 Time consumed to search for list-level and instance-level counterfactual explanations (in seconds) . . . 144
8.4 Percentage of recommendations explained . . . 144
8.5 The average length of counterfactual explanations returned (number of contextual conditions). . . . 145

9.1 Comparison of A^3R, CA-FATA and SECE . . . 148
9.2 Explanations to be presented. . . . 154
9.3 Efficiency, effectiveness, and persuasiveness of different types of explanations. . . . 155

9.4 Subjective evaluation of efficiency, effectiveness, persuasiveness, satisfaction, trust, and transparency . . . 155
9.5 Spearman Rank Order Correlation of metrics . . . 156
9.6 Overall evaluation of explanation types . . . 159

A.1 List of key symbols . . . 174
A.2 Examples of representation of contextual conditions (User-based and user clustered) (Step 1) . . . 175

Part I

Introduction and related work

Chapter 1

Introduction

Contents

1.1 Background 21
1.2 Research motivations 23
1.3 Summary of contributions 25
1.4 Guide to read 27

Recommender systems (RSs) have become an integral component of the intelligent web in the internet era, finding applications in numerous practical systems. As RSs are increasingly being integrated into people's daily lives, they have become critical decision-support tools that can significantly impact users' decision-making process (Lu et al., 2015, Ricci et al., 2015). Consequently, there is a growing demand from users to understand the rationale behind recommendations, particularly in high-stakes domains such as medicine and finance, to mitigate risks. Explainable RSs, which provide recommendations and offer explanations to justify those recommendations, have gained significant attention from both industry and academia (Zhang et al., 2020, Nunes and Jannach, 2017, Daher et al., 2017). "We recommend you this movie because it is a movie starred by Tom Hanks and you gave high ratings to his movies before" is a typical explanation in a movie recommendation scenario. It has been recognized that explanations are crucial for improving the efficiency, effectiveness, persuasiveness, transparency, trust, and satisfaction of RSs (Tintarev and Masthoff, 2015). Despite the emergence of various explanation methods, little research has been conducted on the generality of explainable RSs. In this thesis, we contribute to this topic by proposing a *General Framework for Generating Explanations* (GFGE) and instantiating this framework under two explanation scenarios: model-intrinsic explanations and post-hoc explanations. The aim of this chapter is to provide the research background, outline the research problems and main challenges, their practical significance and scientific value, describe the main contributions of this thesis, and outline its structure.

1.1 Background

From a broader perspective, *Artificial Intelligence* (AI) can be defined as a field of computer science that focuses on the design, development, and implementation of intelligent machines capable of emulating and performing tasks that typically require human intelligence (Russell and Norvig, 2016). These tasks include but are not limited to problem-solving, decision-making, learning, perception,

and natural language processing. AI has rapidly evolved to become a core technology with a wide range of applications in various real-world domains, such as healthcare, finance, recommendation, and communication (Russell and Norvig, 2016). Its advancements have contributed to significant improvements and innovations, revolutionizing industries and impacting society in numerous ways. On one hand, AI has achieved remarkable success through the use of complex models such as deep neural networks (Liu et al., 2017) or ensemble models (Sagi and Rokach, 2018), which possess the ability to perform complex tasks. On the other hand, the deployment of such opaque models with thousands, or even millions, of parameters can be a double-edged sword. The complexity of AI systems[1] makes it challenging to justify the outputs they generate, and without cogent explanations, these systems may not truly assist humans in making better decisions (Zhu et al., 2018). Numerous scholars have underscored the critical role of research in formulating explanations aimed at fostering responsible AI. Lipton (2018) introduced the key desiderata for research centered on generating explanations, comprising trust, causality, transferability, informativeness, and equitable and ethical decision-making. Building on these principles, Arrieta et al. (2020) advanced the rationale behind the research in explanations, emphasizing the promotion of trustworthiness, causality, transferability, informativeness, confidence, fairness, accessibility, interactivity, and privacy consciousness. From a practical standpoint, Hall and Gill (2019) posited that the impetus for research in explanation can be effectively condensed into two categories: intellectual and social.

Motivated by the aforementioned reasons, there has been significant attention from both academia and industry in developing AI systems capable of generating explanations to justify their outputs (Arrieta et al., 2020). According to Lipton (2018), two strategies can be used to explain the outputs of an AI system: (1) utilizing transparent models, where the inner workings of the models can be easily explained (model-intrinsic explanations), such as simple linear models and association rules; (2) providing post-hoc explanations without elucidating precisely how the models work. The term *eXplainable Artificial Intelligence* (XAI) quickly gained popularity after the DARPA[2] (Defense Advanced Research Projects Agency) called for research proposals on AI explainability, it refers to AI technologies that can provide justifications for their decision-making process. The increasing demand from users and regulations such as the General Data Protection Regulation (GDPR) (Voigt and Von dem Bussche, 2017) has sparked significant research interest in XAI in recent years. Some researchers also use the term *Interpretable Artificial Intelligence* to refer to a similar concept. However, there is a lack of consensus on the explicit definitions of XAI and *Interpretable Artificial Intelligence*, with some works using them interchangeably while others distinguish between them. Often, these distinctions are based on researchers' subjective intuitions without solid theoretical support (Miller, 2019). The ill-defined and ambiguous use of the terms *interpretability* and *explainability* has made it challenging to formulate problems in defining, designing, and evaluating AI systems that can provide explanations for their results. To advance the field, it is crucial to formalize and clarify these terms (Doshi-Velez and Kim, 2017).

RSs are frequently employed as decision-support tools, their role being to alleviate the impact of information overload by delivering pertinent suggestions. In circumstances where decision-making carries significant implications, the requirement for explanations supporting these recommendations intensifies. Concurrently, within the realm of RSs, a notable subdivision of AI, there has been a mounting interest in the development of RSs equipped with the capacity to furnish explanations for

[1] A complex and integrated set of components designed to perform tasks or solve problems using AI techniques. It typically consists of three main components: input data, a model or algorithm, and the resulting output or decision.

[2] https://www.darpa.mil/program/explainable-artificial-intelligence

their recommendations. However, similar to the challenges in the field of AI, the terms *interpretable recommendations* and *explainable recommendations* lack clear definitions, making it difficult to compare and generalize the existing methods for providing explanations. As a result, there is an urgent need for a general framework for developing RSs that can generate explanations for recommendations, despite the extensive research conducted in this area.

1.2 Research motivations

As explored in the previous section, there has been a marked upswing in research endeavors focused on the development of AI systems capable of offering explanations for their outputs. RSs, a crucial subset of AI, have seen corresponding growth in efforts toward equipping them with explanatory faculties. However, notwithstanding the considerable attention attracted by this field, there remains a noticeable absence of consensus regarding the definitions of frequently employed terms such as *explainability* and *interpretability*. This ambiguity poses challenges in the formalization, comparison, and evaluation of different explanation methods. Hence, it is crucial to establish clear definitions for these terms before delving into the study. In other words, a general framework capable of clarifying frequently employed, yet vaguely defined terms, aligning with the *generality* aspect mentioned in the title of this thesis. In this regard, the first research question we aim to address is:

RQ1, "What are the differences between *interpretability* and *explainability* in the context of AI, and more specifically in the context of RSs?"

To fully answer this question, the following two issues should be clarified:

- **RQ1.1, "What does *explainability* and *interpretability* mean respectively?"** Existing definitions on the two terms are usually based on researchers' subjective intuitions without solid theoretical support (Miller, 2019), which has made it challenging to formulate problems in defining, designing, and evaluating AI systems that can provide explanations for their results.

- **RQ1.2, "To whom *explainability* and *interpretability* is concerned?"** It is crucial to determine the target audience for these explanations (Vilone and Longo, 2021, Arrieta et al., 2020). Explanations directed at model developers, domain experts, and regulatory entities/agencies differ significantly from those intended for non-expert users.

In a nutshell, we first aim to disambiguate and settle the meaning of *explainability* and *interpretability* to facilitate the comparison and evaluation of different explanation methods in the context of AI, particularly within RSs. Having settled the use of terms *explainability* and *interpretability* in RSs, it would be pertinent to specify:

RQ2, "Why *explainability* and *interpretability* are remarkably demanded in RSs?"

While we have summarized some desiderata in Section 1.1, when it comes to RSs, we will lay out more details about the goals of explaining recommendations derived by RSs by a thorough literature review in Section 3.7: promoting the efficiency, effectiveness, persuasiveness, transparency, trust, satisfaction and scrutability of RSs (Tintarev and Masthoff, 2015). Note that these goals are also applied to evaluate explanations in RSs (Gedikli et al., 2014). In sequence, we arrive at the core research question of this thesis:

RQ3, "How to explain recommendations?"

Similar to the field of AI, explanations in RSs, which is an important branch of AI, can be broadly classified into two main categories (Zhang et al., 2020): (1) adopting transparent models; and (2) adopting post-hoc explanation techniques to generate post-hoc explanations. Transparent models, such as linear regression and association rules, are easy to understand. However, they often suffer from lower prediction accuracy, which is known as the trade-off between complexity and accuracy (Doshi-Velez and Kim, 2017). On the other hand, post-hoc explanation techniques, such as SHAP (SHapley Additive exPlanations) (Lundberg and Lee, 2017) and LIME (Local Interpretable Model-agnostic explanations) (Ribeiro et al., 2016), can be applied to explain any models thus offering more flexibility. However, one major concern with applying such techniques is that post-hoc explanations may not always faithfully represent the original RSs and may even be inaccurate (Rudin, 2019). Hence, both strategies have their merits, and it is not an either-or situation. This naturally leads us to ask:

- **RQ3.1, "How to build transparent models that can explain recommendations while ensuring the accuracy of recommendations?"** Directly building RSs capable of generating accurate recommendations and explaining such recommendations is an important topic. Existing RSs often rely on latent factor models such as matrix factorization (Koren et al., 2009), Factorization Machine (Rendle, 2010), deep learning-based methods (Liu and Wu, 2017), etc. These models can make accurate rating predictions. Nevertheless, the explicit semantics of the learned latent factors remain ambiguous, thereby complicating the task of generating coherent and cogent explanations for the recommendations.
- **RQ3.2, "How to explain existing complex RSs using post-hoc explanation techniques while mitigating the limitations of such techniques?"** Retaining existing complex RSs in production and adopting post-hoc explanation techniques (Lipton, 2018) to explain recommendations is practical. It allows for the reuse of existing RSs, reducing engineering costs and these RSs usually gain higher performances than those that adopt simple models. SHAP (Lundberg and Lee, 2017) and LIME (Ribeiro et al., 2016) are among the most prominent tools that can provide post-hoc explanations in literature. There are some works (Guo et al., 2021, Singh and Anand, 2019, Nóbrega and Marinho, 2019) that explain recommendations by directly applying the importance of each feature calculated by LIME and SHAP. The major limits and concerns of applying directly such methods are: (1) post-hoc explanations may not be faithful to the original RSs and sometimes such explanations may even be wrong (Rudin, 2019); (2) for plain users, understanding the importance of features returned by such methods is non-trivial since complex background mathematical knowledge is required. For example, the outputs of SHAP include the *force_plot* figure (used to visualize how a prediction is achieved by showing the contribution of each feature please refer to Lundberg and Lee (2017)), this requires that users have basic knowledge in game theory. Therefore, SHAP and LIME are good tools for model developers to visualize the behaviors of models and may help them debug models, however, for plain users, such figures may not be easy for them to understand. The pivotal question then emerges: how can we harness post-hoc explanation methods to elucidate any recommender model while sidestepping these constraints?

Various explanation methods have been proposed in the domain of RSs, however, it remains unclear what the best way is to provide explanations under a given context for a given task (Zhou et al., 2021). This leads us to our final research question:

RQ4, "How to evaluate explanations in RSs?"

Evaluating explanations is challenging as explanations are human-centered (Miller, 2019), and the ground truth explanation for a machine learning task is often unknown (Molnar et al., 2020), which is also the case for RSs. To address this question, we assess the quality of explanations in RSs by measuring the extent to which explanations satisfy the goals identified by Gedikli et al. (2014) and Tintarev and Masthoff (2015).

1.3 Summary of contributions

In this thesis, we make contributions to the fields of generality and explainability in RSs by addressing the following aspects: (1) we propose an *Interpret/Explain Schema* (IES) that allows disambiguating the essential terms used in AI, particularly in RSs, to avoid confusion and clarify their meanings. Based on IES, we construct GFGE that can integrate existing methods for generating explanations in the context of AI; (2) we summarize the drives for research related to explanations in AI and provide an overview of methods for evaluating explanations in RSs; (3) we instantiate the framework with two explanation scenarios in RSs to demonstrate its effectiveness and applicability; (4) we conduct a preliminary user study to evaluate the qualities of explanations.

Disambiguating the use of terms (RQ1): we thoroughly review and compare existing works on the use of *explainability* and *interpretability* in the context of AI. It is crucial to note the social aspects of explanations, as they involve the transfer of knowledge through conversations or interactions (Miller, 2019, Malle, 2006). To shed light on this, we draw upon concepts from knowledge management (Arduin et al., 2015), where humans possess an interpretative framework (Jones et al., 2011) (also known as mental model (Tsuchiya, 1993)) that enables them to interact with their environment through *sense-giving* and *sense-reading* (Polanyi, 1967). With this, we propose the IES (Zhong and Negre, 2021) in AI systems. We propose the definition of *explainability* and *interpretability* based on the IES and further specify what does *explainability* and *interpretability* means respectively; to whom *explainability* and *interpretability* is concerned. With the proposed definitions, problem formulation such as definition, design, and evaluation of explainable AI can be seriously engaged, which will make the AI system more transparent when making decisions. With IES and our definitions, we construct the GFGE in the context of AI. Under our framework, the two strategies (Lipton, 2018) (Utilizing transparent models and providing post-hoc explanations) applied to explain the output of AI systems can be integrated together. Note that the framework also applies in RSs, since RSs is an important branch of AI research, and we will instantiate the framework by applying the two strategies in the context of RSs.

Drives for research in the area (RQ2, RQ4): we thoroughly review existing literature on explanations in AI and classify them into three drives (Zhong and Negre, 2021): (1) commercial drive that comes from the demands of users; (2) regulatory drive that comes from regulatory demands; (3) technical drive that concerns debugging and improving existing AI systems. We also summarize the goals of explanations in RSs: helping users understand how the system works (transparency), increasing user's confidence in the system (trust), helping users make better (effectiveness) decisions more quickly (efficiency), persuading the user to try or consume (persuasiveness), allowing users to tell the system when recommendations are not appropriate (scrutability) and increasing the ease of use or enjoyment (satisfaction). Note that these goals are often used to evaluate

the quality of explanations in RSs (Gedikli et al., 2014, Tintarev and Masthoff, 2015).

Two explanation scenarios (RQ3): we instantiate GFGE by applying the two explanation strategies:

- **To address the question of the role of transparency (RQ3.1)**: we turn to Argumentation Frameworks (AFs), which have been applied to enhance the explainability of AI (Vassiliades et al., 2021, Čyras et al., 2021). AF can represent the decision-making process in a graphical way, and the predefined properties help to reason how to reach the best decisions (Vassiliades et al., 2021). AF also includes ways to define weighted arguments and dialectical relations between arguments, which contain semantics to decide how strong and acceptable these arguments are (Čyras et al., 2021).

  - **Context-free**: we propose **A**ttribute-**A**ware **A**rgumentative **R**ecommender (A^3R) (Zhong and Negre, 2022a), a model that combines factorization-based methods and AFs. With the help of AFs, each step of A^3R is endowed with explicit semantics, allowing A^3R to generate explanations for recommendations that are easy to understand. For each user-item interaction, we define an AF, where features of items are regarded as arguments, and users' ratings towards features are defined as the strength of arguments. Model-intrinsic explanations can be generated for each user-item interaction with the help of the AFs.
  - **Context-aware**: a prevailing belief in the domain of RSs is that users' context can influence their preferences (Adomavicius and Tuzhilin, 2011). We believe that this also applies to AFs. Some *good* arguments under particular contexts may become less accurate given another context. Nevertheless, A^3R does not consider users' contexts that can strongly influence users' preferences. We extend this idea by considering users' context and propose **C**ontext-**A**ware **F**eature **A**ttribution **T**hrough **A**rgumentation) (Zhong and Negre, 2023). The traits of $CA-FATA$ include: (1) $CA-FATA$ combines latent factor models and AFs, therefore, each computation step is endowed with explicit semantics, which helps to explain the generated recommendations; (2) since users' contexts are considered, the corresponding AF is also context-aware, which means that $CA-FATA$ can generate context-aware explanations (Zhang et al., 2020, Zhong and Negre, 2022e), therefore, $CA-FATA$ can generate and explain recommendations in an argumentative and context-aware way.

- **To answer the question of using post-hoc explanations (RQ3.2)**: we draw inspiration from related research in psychology, philosophy, and cognitive science. Instead of directly asking why an event P happened, people often prefer asking why event P happened rather than an event Q, where Q is a foil (Lipton, 1990, Lim and Dey, 2009, Mackie, 1980). Generating counterfactual explanations (Wachter et al., 2017) that are easily understandable (Wachter et al., 2017) and of high fidelity (Kaffes et al., 2021) is a potential solution. Unlike LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017), which approximate original models, counterfactual explanations seek the minimum changes necessary to alter the outputs of the original models. Therefore, counterfactual explanations have high fidelity (Kaffes et al., 2021). A typical counterfactual explanation describes a causal situation as "If A did not occur, then B would not have happened" (Molnar, 2020). We proposed **S**hap-**E**nhanced

Counterfactual **E**xplanation (SECE) (Zhong and Negre, 2022c) and show that SHAP (Lundberg and Lee, 2017) can be adapted to generate counterfactual explanations that are contrastive and selected (Zhong and Negre, 2022c). To the best of our knowledge, this is the first work that combines the Shapley value method and counterfactual method to address explanations in RSs: we apply SHAP (Lundberg and Lee, 2017) to select possible candidates of counterfactual explanations. Our method can generate counterfactual explanations that are faithful and easily understandable.

The quality of explanation (RQ4): we also summarize existing predefined proxies frequently used for evaluating explanations in RSs (Zhong and Negre, 2022e). At the same time, since explanation is human-centered, we conduct a user study (Zhong and Negre, 2022b) to evaluate the qualities of explanations generated by our explanation methods: the extent to which explanations satisfy the goals of explanations: transparency, trust, effectiveness, efficiency, scrutability, satisfaction, persuasiveness (Gedikli et al., 2014, Tintarev and Masthoff, 2015)

1.4 Guide to read

The rest of this thesis is organized as follows:

- In Chapter 2, we depict state-of-art related to explanation from the view of AI. We cover the fundamental concepts of explanations in AI, how explanations in AI may have been influenced by research from other domains such as philosophy, psychological science, cognitive science, how to generate explanations and how to evaluate explanations.

- In Chapter 3, we review related work about explainable recommendations by applying the notions introduced in Chapter 2.

- In Chapter 4, we depict the IES, which allows disambiguating the use of terms: *explainability* and *interpretability*. Based on the IES, we construct GFGE. We show that existing methods for generating explanations can be integrated into this framework.

- In Chapter 5, we lay out some technical background that will be used to instantiate GFGE in Chapters 6, 7, and 8.

- In Chapter 6, we instantiate GFGE in the model-intrinsic explanation scenario, where recommendations can be explained inherently by design. Starting with a context-free scenario, we introduce A^3R which utilizes AFs to endow each computation with explicit semantics. Our experiments indicate that A^3R significantly surpasses (in terms of prediction accuracy) existing argumentation-based recommendation models and competes effectively with other more opaque, black-box recommendation models.

- In Chapter 7, we further develop the model introduced in Chapter 6 by incorporating the users' context into a new model named $CA-FATA$. As this model takes users' contexts into account, the AFs it utilizes also becomes context-aware. This means that $CA-FATA$ is capable of generating context-aware explanations. Consequently, $CA-FATA$ can both generate recommendations and provide explanations for them in an argumentative, context-aware manner.

- In Chapter 8, we instantiate GFGE in the post-hoc explanation scenario, where black-box models are retained and post-hoc explanations are generated. More specifically, we leverage SHAP to select possible candidates of counterfactual explanations
- In Chapter 9, we compare the two explanation scenarios and present the result of a user study that evaluates different types of explanations.
- In Chapter 10, we conclude the thesis, propose avenues for future work, and list the publications during the Ph.D. program.

Chapter 2

Explanations in AI

Contents

2.1 A general view **29**
2.2 To explain or to interpret? **30**
2.3 Explanation: what? **31**
2.4 Explanation: why? **33**
2.5 Explanation: how? **35**
2.5.1 Transparent models: model-intrinsic explanations 35
2.5.2 Post-hoc explanations 36
2.6 Explanation: how to evaluate? **37**
2.7 Conclusions **38**

In this chapter, we offer an exhaustive review of studies focusing on explanations within the realm of *Artificial Intelligence* (AI). Our aim is to guide readers in comprehending the fundamental principles of explanations in AI, which will subsequently be utilized in our discussion on explainable recommendations. The chapter is organized as follows: Section 2.1 provides a general overview of research related to explanations in AI, emphasizing some frequently used yet ill-defined notions. Section 2.2 discusses existing work that attempts to disambiguate the terms *interpretability* and *explainability*. Section 2.3 introduces research on explanations from social science, cognitive science, philosophy, and knowledge management, which will contribute to the development of the *Interpret/Explain Schema* (IES) and the *General Framework for Generating Explanations* (GFGE) in Chapter 4. Section 2.4 summarizes the motivations behind research on explanations in AI. Section 2.5 reviews two primary families of techniques employed for generating explanations in AI. Section 2.6 examines existing approaches for evaluating the quality of explanations in AI. Finally, we compare the two families of techniques and provide four guiding principles for selecting between them.

2.1 A general view

In recent years, AI has been widely applied, bringing significant changes to people's daily lives and becoming a core technique in various real-world applications, including *Recommender Systems* (RSs) (Zhang et al., 2020). However, some people may question the reliability of these techniques

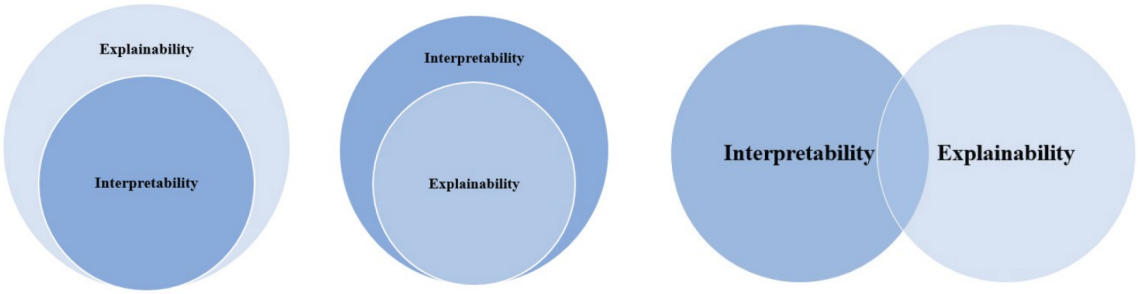
in deployment. In particular, when AI is used for making high-stakes decisions in areas such as healthcare and criminal justice, individuals often want to know why a system makes certain decisions to mitigate risks, as it can be difficult to trust a system without explanations. Moreover, since 2018, the European Union has mandated that algorithms used in decision-support systems must provide explanations, a principle known as the "right to explanation" (Voigt and Von dem Bussche, 2017). People desire AI systems that can provide high-quality results and reasonable explanations simultaneously. The definition, design, optimization, and evaluation of such AI systems have attracted significant attention. Researchers often claim that their models are interpretable or explainable, referring to *interpretability* and *explainability*, respectively. However, no strict definitions of these terms have been achieved. Some researchers distinguish between them (Lipton, 2018, Doshi-Velez and Kim, 2017, Montavon et al., 2018), while others use them interchangeably (Miller, 2019, Molnar, 2020, Du et al., 2019, Arrieta et al., 2020). These claims are typically based on researchers' subjective intuitions (Miller, 2019) without solid theoretical support, and so far, no consensus has been reached. The vague definitions and ambiguity in using the terms *interpretability* and *explainability* have made it difficult to formulate problems in defining, designing, and evaluating AI systems that can provide explanations for the generated results. To advance research on *interpretability* and *explainability* in AI, it is imperative to critically engage with and rigorously define these terms (Doshi-Velez and Kim, 2017).

2.2 To explain or to interpret?

In the AI research community, there is no common consensus or strict definitions concerning *interpretability* and *explainability*, despite significant efforts devoted to this research subject. Numerous discussions have taken place about the definitions of these two terms.

According to Lipton (2018), interpretable models can be classified into two categories. The first category is transparent models, which means that how the models work can be explained and understood by humans. The second category is models that can provide post-hoc explanations, meaning that they can offer explanations after generating results, without necessarily elucidating precisely how the models work. Lipton distinguishes between *interpretability* and *explainability*, in that the former enhances the latter. Gilpin et al. (2018) argue that *interpretability* and *explainability* should be distinguished because explainable models are interpretable by default, but the reverse is not always true. They propose that explaining explanations is an approach to evaluating *interpretability*. In another line of research, some researchers use *interpretability* and *explainability* interchangeably. Miller (2019) discusses explanations in AI from a social science perspective. This is the first attempt to link explanation research in AI to psychology, social science, and cognitive science. The author surveys philosophy, psychology, and cognitive research related to explanation. According to Miller, *interpretability* is the degree to which an observer can understand the cause of a decision and is widely cited. The author equates *interpretability* and *explainability*. The surveys and reviews listed above are not exhaustive, but they represent three different relationships concerning *interpretability* and *explainability* identified from state-of-the-art works, as presented in Figure 2.1. For more reviews and surveys about *interpretability* and *explainability* in the context of AI, see the work of Hoffman et al. (2018), Zhang et al. (2020) and Mittelstadt et al. (2019).

Figure 2.1(a) illustrates that *interpretability* is a subset of *explainability*, meaning that *interpretability* enhances *explainability*. This relationship has been suggested by several works such as Lipton (2018), Montavon et al. (2018), Guidotti et al. (2018), Mittelstadt et al. (2019), Rudin



(a) Interpretable models are explainable by default

(b) Explainable models are interpretable by default

(c) Explainable and interpretable are used interchangeably

Figure 2.1: Relationships of *interpretability* and *explainability* in state-of-art works

(2019). In these works, the authors argued that interpretation pertains to the internal mechanisms of models and how they work, while explanation pertains to why certain results have been generated. An interpretable model is explainable by default, since the reasoning process behind the results (e.g., a recommendation or a classification) is clear, indicating that the results can be explained using understandable logic. Figure 2.1(b) shows another suggested relationship between *interpretability* and *explainability* that distinguishes them, with the reverse implication. This view is presented by Gilpin et al. (2018), which defines explainable models as interpretable by default, but not the other way around. This suggests that *explainability* implies *interpretability*. In this sense, the correctness of explanations should be justified, revealing the causal relationships behind them. Figure 2.1(c) illustrates that the terms *interpretability* and *explainability* can be used interchangeably. According to Molnar (2020) and Arrieta et al. (2020), the term interpretable is more often used when referring to models, while explainable is more often used when referring to results. Interpreting a model can also mean producing explanations for individual predictions (Molnar, 2020).

To summarize, no consensus has been achieved upon the use of *interpretability* and *explainability*. A lack of agreement on what is explanation may have been the reason why researchers use the terms differently.

2.3 Explanation: what?

To clarify ambiguity, it is important to note that scholars from a variety of disciplines have investigated the concept of explanation long before the advent of AI. Philosophers have delved into the nature, purpose, and structure of explanations, while cognitive and social psychologists have examined how individuals attribute and evaluate the behavior of others in physical environments. Additionally, cognitive psychologists and scientists have studied how people generate and evaluate explanations.

What is explanation? Lewis (1986) (one of the most significant philosophers of the XXth century) posited that: "*To explain an event is to provide information about its causal history*[3]. *In*

[3]Although causality is crucial in explanation, the analysis of causality lies outside the scope of this thesis.

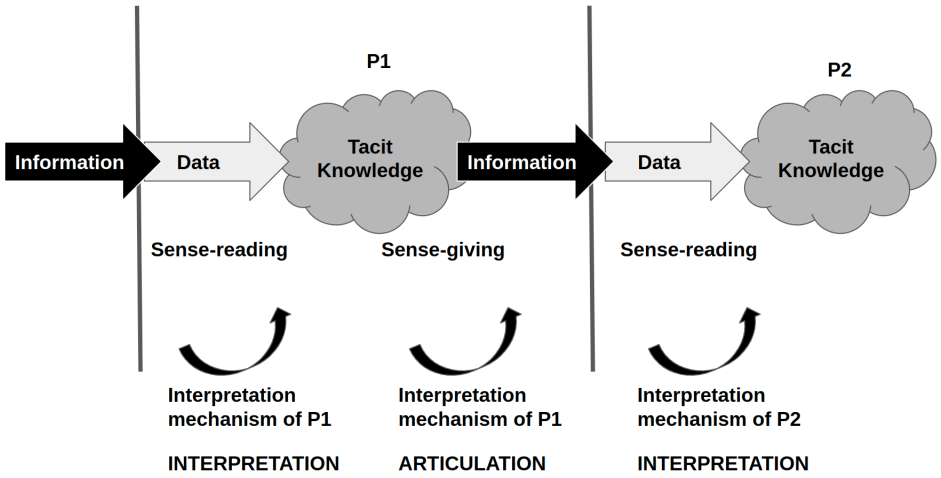


Figure 2.2: *Sense-giving* and *sense-reading* in knowledge management (Arduin et al., 2015), where P1 and P2 are two different persons.

explaining, someone possessing explanatory information about an event's causal history attempts to convey it to someone else." According to Josephson and Josephson (1996): "*An explanation is an assignment of causal responsibility.*" Malle (2006) claimed that regarding explanations: "*Explanations involve social interactions, such as creating a shared meaning of something, transferring knowledge, influencing people's beliefs, and so on.*" Lombrozo (2006) and Miller (2019) observed that explanation is both a process and a result. An explanation can be viewed as a product as it can function as an answer to a why question (Dennett, 1989, Lewis, 1986, Lipton, 1990). On the other hand, an explanation can also be seen as a process that involves abductive inference to identify an appropriate explanation for a given event (Chin-Parker and Bradner, 2010). Additionally, the process of explanation requires knowledge transfer (through conversations or interactions (Miller, 2019, Malle, 2006)) between the explainer and the explainee.

In fact, the assertions of Lombrozo (2006) and Miller (2019) are remarkably consistent with the concepts in knowledge management (Despres and Chauvel, 1999, Arduin et al., 2015). Human beings possess an interpretative framework (Jones et al., 2011), also known as a mental model (Tsuchiya, 1993), through which new information is filtered and stored. This allows them to interact with the world, process information, and ultimately create knowledge. Each individual has their own unique interpretative framework, leading to different interpretations of the same data even when two people see the same thing. Arduin et al. (2015) further emphasize that interpretation is central to knowledge management. Continuously, humans interpret information through the process of *sense-reading* (Polanyi, 1967). For instance, at this moment, as I send this text to you, each recipient interprets the information based on your own interpretative framework. I use my interpretative framework to give meaning to the information I create, a process called *sense-giving* (Polanyi, 1967). In turn, each of you will perceive and interpret the information, creating your own knowledge through *sense-reading* (Polanyi, 1967). Thus, it is through interpretation that humans select and process data to create their own knowledge. The combination of *sense-giving* and *sense-reading* results in an explanation, as illustrated in Figure 2.2. This aligns with Miller's assertion (Miller, 2019) that *explanations are social and involve conversations.* From a philosophical standpoint, interpretation is a subjective action, while explanation entails interaction. With

this, we will propose the *Explain/Interpret Schema* in Chapter 4.

What are the desired properties of explanations? A summary of existing research on explanations in social science, cognitive science, and philosophy reveals three common desired properties of explanations:

- **Explanations are contrastive:** people often prefer to ask "why did event P occur rather than event Q?" instead of simply questioning "why did event P occur?" Here, P and Q represent the fact and the foil, respectively. This distinction results in two types of questions: plain-fact and contrastive. Plain-fact questions require a comprehensive explanation with an uninterrupted causal chain across time (Van Bouwel and Weber, 2002), while contrastive questions focus on the differences between P and Q, which are easier to derive (Lipton, 1990). As a result, desired explanations tend to emphasize these differences rather than providing a complete causal analysis, leading to the observation that explanations are selective by nature.

- **Explanations are selected:** instead of pursuing the entire cause of an event, people typically choose a limited number of causes to form an explanation. One possible reason is that comprehensive causal analyses can be too complex to grasp (Hilton, 2017). Contrasts or differences between facts and foils serve as the primary criteria for selecting explanations, suggesting that explanations should be tailored to address the specific needs of the explainees.

- **Explanations are social:** explanations serve as a conduit for knowledge transfer during conversations or interactions, taking into account the explainer's perception of the explainee's beliefs (Miller, 2019, Malle, 2006). Consequently, it is essential to consider human involvement when developing explanation methods (Vilone and Longo, 2021). Considering that explanations generally involve interactions, it is crucial to determine the target audience for these explanations (Vilone and Longo, 2021, Arrieta et al., 2020). Explanations directed at model developers, domain experts, and regulatory entities/agencies differ significantly from those intended for non-expert users. In this thesis, our main emphasis is on explanations targeted at non-expert users, unless stated otherwise. Additionally, it is worth noting that the analyses presented in this section stem from a broader perspective (in AI), yet they remain applicable to the domain of RSs.

The above three properties are desired when designing explanations in the context of AI (Miller, 2019). In this thesis, we will design models to generate explanations in RSs that fulfill at least one of the aforementioned properties.

2.4 Explanation: why?

Numerous researchers have synthesized and enumerated the reasons (or desiderata of explanation) for research on explanations in AI (Lipton, 2018, Arrieta et al., 2020, Hall and Gill, 2019). We further categorize these desiderata into three driving forces: commercial drive, regulatory drive, and technical drive (Zhong and Negre, 2021).

Commercial drive: a key driving force for explanation in AI is the commercial drive. In today's competitive landscape, many companies rely on AI techniques as their core competency

Table 2.1: Potential goals of explanations in RSs

Aims	Descriptions
Efficiency	Help users make decisions more quickly
Effectiveness	Helps users make better decisions
Persuasiveness	Change users' behaviours
Trust	Increase user's confidence in the system
Transparency	Help users understand how the system works
Satisfaction	Increase the ease of use or enjoyment
Scrutability	Allow users to tell the system when recommendations are not appropriate

to provide innovative services. However, humans are naturally curious and tend to trust decisions backed by logical reasoning. When AI is used to make high-stakes decisions, consumers seek explanations or reasoning to support their decision-making process, as explanations play a crucial role in building trust (Arrieta et al., 2020). For instance, users are more likely to trust a well-explained recommendation compared to one generated by a black-box model, as it provides transparency and justification for the recommendation (Tintarev and Masthoff, 2022). In this context, trustworthiness (promoting users' trust) (Kim et al., 2015, Ribeiro et al., 2016), fairness (avoiding biased decisions) (Chouldechova, 2017, Voigt and Von dem Bussche, 2017), and interactivity (promoting scrutability) (Tintarev and Masthoff, 2015) fall under this category.

Regulatory drive: regulations such as the General Data Protection Regulation (GDPR) (Voigt and Von dem Bussche, 2017) also drive the need for explainable AI, as they mandate that consumers have the legal right to obtain explanations. This makes providing explanations to users necessary, especially when decision-making processes are influenced by RSs (Edwards and Veale, 2017).

Technical drive: finally, the technical drive is another driving force. After years of research and real-world applications, researchers have realized that a model solely based on prediction accuracy cannot always be trusted, as accuracy is an incomplete description of real-world tasks (Doshi-Velez and Kim, 2017). Not knowing the reasoning process behind result generation may lead model builders to create erroneous models. In contrast, understanding the true reasoning behind models assists designers in debugging and improving models (Louizos et al., 2017).

More specifically, the primary objectives (or benefits) of explanations in RSs include promoting transparency, effectiveness, efficiency, persuasiveness, trust, satisfaction, and scrutability (Tintarev and Masthoff, 2022). In practical terms, explanations can assist users in identifying items that truly align with their interests (effectiveness) more efficiently (efficiency), persuading users to try or consume items (persuasiveness), increasing users' loyalty to the system (trust), enhancing the understandability of RSs (transparency), improving ease of use or enjoyment (satisfaction), and enabling users to communicate with the system when recommendations are unsuitable (scrutability). Table 2.1 summarizes these aims, primarily from the users' perspective, which is the central focus of this thesis. These goals are commonly used as metrics to evaluate the quality of explanations in RSs (Tintarev and Masthoff, 2015, Gedikli et al., 2014), and we will revisit these objectives (metrics) in Section 3.7.

2.5 Explanation: how?

Upon reviewing the literature, it appears that existing works identify two primary strategies for generating explanations (Lipton, 2018, Guidotti et al., 2018): (1) employing transparent models, where the inner workings of the models can be readily explained (model-intrinsic explanations), such as simple linear models and association rules (Kotsiantis and Kanellopoulos, 2006); (2) offering post-hoc explanations without explicitly detailing the underlying mechanisms of the models. In the following, we will provide an overview of the two broad categories of methods and their applications in the realm of RSs.

2.5.1 Transparent models: model-intrinsic explanations

Transparency encompasses three distinct levels: simulatability (encompassing the entire model), decomposability (pertaining to individual components such as parameters) and algorithmic transparency (related to the training algorithm) (Lipton, 2018, Arrieta et al., 2020).

- **Simulatability:** in the context of model transparency, it refers to the ability of a person to understand the entire model at once. This often implies that simpler models are more interpretable. For instance, sparse linear models produced by lasso regression are considered more interpretable than dense linear models. An interpretable model should be easily presented to the user through visual or textual artifacts.

- **Decomposability:** when users cannot simulate an entire model, they may still seek to understand the meaning of each component within the model. Decomposability refers to the capacity to provide intuitive explanations for each part of a model, encompassing input, parameter, and calculation. This characteristic, also referred to as intelligibility (Lou et al., 2012), enhances the understanding and interpretation of a model's behavior. For example, General Additive Model (GAM) (Hastie, 2017) falls into this category. However, not all models can achieve decomposability, as it necessitates that each input be readily interpretable, thus excluding models with highly engineered or anonymous features. For instance, each node in a decision tree could correspond to a plain text description, while parameters in a linear model might represent the strengths of associations between features and labels.

- **Algorithmic transparency:** algorithmic transparency involves a user's ability to comprehend the process a model utilizes to generate the output from input data. For instance, linear models are deemed transparent as their error surfaces can be understood and rationalized, enabling users to anticipate the model's behavior in different situations. Conversely, contemporary deep learning methods, including deep architectures, do not possess this degree of algorithmic transparency. The loss landscape of these models may be obscure, and their solutions require approximation using heuristic optimization techniques like stochastic gradient descent (Datta et al., 2016).

It is important to recognize that models generally regarded as transparent may not always exhibit transparency. The trade-offs between model size and computation for generating a single prediction can vary among models. For instance, in the case of decision trees, the time required for inference might expand considerably in comparison to the model size.

2.5.2 Post-hoc explanations

When models become overly complex, such as deep neural networks, it becomes necessary to develop separate methods to extract valuable information and explain the model's decisions. This approach may stem from human cognitive science, wherein individuals may make intuitive decisions and subsequently seek explanations for those decisions (Miller, 2019, Zhang et al., 2020), which falls within the realm of post-hoc explanation approaches. Depending on the scope of explanations, post-hoc explanations can be roughly divided into two levels:

- **Local vs Global:** local explanations offer detailed insights into individual instances or predictions, allowing users to comprehend the decision-making process behind specific AI predictions. Notably, techniques such as counterfactual explanations (Wachter et al., 2017) clarify what would need to change in the input for the model's prediction to alter, thereby highlighting important factors the model considered. Similarly, LIME (Local Interpretable Model-agnostic Explanations) (Ribeiro et al., 2016) generates localized linear approximations of the model's decision boundary, and SHAP (SHapley Additive exPlanations) (Lundberg and Lee, 2017) provides instance-level feature importance based on cooperative game theory, both return features contributing to a specific prediction. On the other hand, global explanations aim to interpret a model's overall behavior, providing a holistic understanding of how the model makes predictions across all instances. DeepLIFT (Deep Learning Important FeaTures) (Shrikumar et al., 2017), a global explanation technique, measures the contribution of each input feature to the model's output by comparing it to a reference input, thus exposing the model's overall feature importance structure. Permutation feature importance (Breiman, 2001) is another global technique, providing an overall metric of feature importance by assessing the increase in the model's prediction error after permuting the feature's values. While local explanations can be invaluable in understanding case-specific decisions and debugging outliers, they may not give a complete picture of the model's functioning and might sometimes be inconsistent across instances (Ribeiro et al., 2016). In contrast, global explanations provide a comprehensive overview but can be overly generalized and fail to explain specific, nuanced decisions (Gilpin et al., 2018). As such, an ideal approach might involve a combination of both local and global techniques, providing both instance-level insights and an overall understanding of the model's operation (Molnar, 2020).

- **Model-specific vs model-agnostic:** model-specific explanation techniques are designed with particular types or classes of models in mind. As such, they exploit the inherent properties and structures of these models to elucidate their internal workings. For example, TreeSHAP (Lundberg et al., 2018) is actually a variant of SHAP (Lundberg and Lee, 2017). It allows computing the marginal contribution of each feature in a more efficient way but only applies to tree-based models. Zeiler and Fergus (2014) developed a filter and activation-based method to explain the behavior of convolutional neural networks. In contrast, model-agnostic explanation techniques are not restricted to a specific type of model. They aim to understand the relationship between the model's inputs and outputs, without delving into the model's internal structure. These techniques can be used across a wide variety of models, adding a layer of versatility to their applications. LIME (Ribeiro et al., 2016) is one such technique that generates explanations by approximating the model locally with an interpretable model. It does not rely on the internal mechanics of the model, making it a flexible tool for various types of models. SHAP (Lundberg and Lee, 2017) also falls under this category. It explains

the output of any model as a sum of the effects each feature has on the prediction, thus making the model output interpretable. However, both model-specific and model-agnostic techniques have their strengths and limitations. Model-specific techniques can provide deep insights into specific types of models but are not universally applicable. Model-agnostic techniques, on the other hand, offer wide applicability but may not delve as deeply into model-specific structures. Thus, the choice between these techniques should be informed by the particular use-case and the model in question (Molnar, 2020).

2.6 Explanation: how to evaluate?

The evaluation methods can be generally classified into three strategies, as proposed by Doshi-Velez and Kim (2017). These approaches aim to offer a systematic means of assessing the quality of explanations produced by AI systems:

- **Application-grounded evaluation:** in this approach, the evaluation of explanations is performed by conducting user studies with real-world users in a specific domain. The idea is to measure how well the explanations provided by the AI system help users achieve their goals, such as making better decisions, debugging, or understanding the model's reasoning. By testing the explanations with real users in a practical setting, this evaluation method ensures that the explanations are relevant and useful for the intended audience. For instance, Zhang et al. (2014) conducted an A/B test to verify how explanations can influence the sales of mobile phones in an online shopping website called JingDong[4].

- **Human-grounded evaluation:** this approach involves simulating the target user population by conducting user studies with laypeople or domain experts who are not involved in the development of the AI system. The participants are asked to perform tasks or answer questions based on the explanations provided by the AI system. This evaluation method assesses the quality of the explanations in terms of understandability, informativeness, and usability for the intended audience without relying on real-world application scenarios. Conducting user studies is a typical evaluation method that falls into this type. For instance, Gedikli et al. (2014) conducted a user study to compare and evaluate different explanations in RSs.

- **Functionally-grounded evaluation:** in contrast to the first two approaches, functionally-grounded evaluation does not involve human participants. Instead, this approach uses quantitative proxies or automated metrics to evaluate the quality of explanations. The idea is to design tasks that can be automatically performed and evaluated, such as measuring the consistency of explanations, their fidelity to the underlying model, or their ability to predict human judgments. This evaluation method provides a scalable and objective way to assess explanations, but it may not capture all aspects of human interpretability and usability. When explanations take the form of text, BLEU (Papineni et al., 2002) and ROUGE (ROUGE, 2004) scores and typically used to evaluate the quality of the text. Tan et al. (2021) defined Probability of Necessity (PN) and Probability of Sufficiency (PS) for assessing the qualities of counterfactual explanations.

In summary, the three strategies provide complementary methods for evaluating explanations in AI systems. Researchers and practitioners may opt for one or a combination of these evaluation

[4] http://www.jd.com

techniques, taking into account the context and objectives of the AI system to ensure that the generated explanations are comprehensible, valuable, and reliable for the target users. For a more comprehensive review of explanation evaluation approaches, refer to Mohseni et al. (2021) and Hoffman et al. (2018).

2.7 Conclusions

It is not possible to definitively say which is better—model-intrinsic explanation techniques or post-hoc explanation techniques in AI—because the suitability of these techniques depends on the specific context and goals of a given project. Here are some comparisons between the two:

- **Model complexity:** model-intrinsic explanation techniques emphasize inherently interpretable models by incorporating explanations directly into the model's structure, thereby providing clear insights into the decision-making process. Examples of such models include linear regression (Hastie et al., 2009), GAMs (Hastie, 2017), decision trees (Breiman, 2001), and argumentation-based methods (Vassiliades et al., 2021). However, when dealing with complex models like deep neural networks, post-hoc explanation techniques might be more suitable, as they can offer insights into the model's decision-making process after training (Ribeiro et al., 2016).

- **Performance trade-offs:** model-intrinsic explanations provide enhanced transparency, but they may result in lower predictive performance compared to more complex models (Doshi-Velez and Kim, 2017). Conversely, post-hoc explanations can be applied to a broad range of models, including those with high predictive performance; however, they might not deliver the same level of transparency as model-intrinsic explanations (Gilpin et al., 2018).

- **Domain knowledge:** model-intrinsic explanation techniques are often more suitable when domain knowledge is crucial for comprehending and validating the model's decision-making process (Guidotti et al., 2018). In contrast, post-hoc explanation techniques can be valuable for investigating relationships within the data that may not be captured by simpler models (Lundberg and Lee, 2017).

- **Stakeholder requirements:** when stakeholders require a clear understanding of a model's reasoning, model-intrinsic explanations are often more appropriate (Rudin, 2019). These techniques are typically more intuitive and easier to comprehend since they are designed with interpretability in mind. On the other hand, if predictive performance is a higher priority and stakeholders are comfortable with complex models, post-hoc explanation techniques may be a better choice (Ribeiro et al., 2016). These techniques can be applied to a wide variety of models, including complex ones like deep neural networks, which increases their flexibility (Ribeiro et al., 2016). However, a significant concern with post-hoc explanation techniques is that they may not always accurately represent the original reasoning process and could even be misleading (Rudin, 2019).

In conclusion, this chapter has provided a comprehensive examination of explanations in the context of AI, equipping readers with a robust understanding of the key concepts and techniques. By elucidating the distinctions between interpretation and explanation, delving into the nature of explanations, and exploring various methods for generating and evaluating explanations, this

chapter imparts essential knowledge about existing research related to explanations in AI. The prevalence of a lack of consensus on notions (particularly *interpretability* and *explainability*) in the AI domain is apparent. To tackle this issue, we have presented concepts from knowledge management to disambiguate existing notions. Furthermore, the interdisciplinary perspectives presented in this chapter contribute to a more comprehensive understanding of explanations and their applications in AI. These notions are also applicable to RSs, which will be discussed in the following chapter.

Chapter 3

Explainable recommendations

Contents

3.1 Recommender systems 42
3.1.1 Recommendation problem 42
3.1.2 Evaluating recommendations 43
3.1.3 Recommendation approaches 46
3.2 Explainable recommendations: an overview 47
3.3 Content-based recommender systems 48
3.3.1 CBRSs: generating recommendations 48
3.3.2 CBRSs: explaining recommendations 48
3.3.3 Summary of CBRSs 49
3.4 Collaborative filtering (CF) 50
3.4.1 Neighborhood approaches 50
3.4.2 Latent factor models 51
3.5 Hybrid approaches 54
3.5.1 Hybrid approaches: generating recommendations 54
3.5.2 Hybrid approaches: explaining recommendations 55
3.5.3 Summary of hybrid approaches 55
3.6 Context-aware recommender system 55
3.6.1 CARSs: generating recommendations 55
3.6.2 CARSs: explaining recommendations 57
3.6.3 Summary of CARSs 58
3.7 Evaluating explanations in RSs 58
3.8 Conclusions 61

In the previous chapter, we presented several fundamental and essential concepts related to explanations within the context of AI. In this chapter, we will review the state-of-the-art in explainable recommendations. Generally, explainable recommendations have two primary goals (Zhang et al., 2020): (1) generating high-quality recommendations, and (2) providing explanations to justify the rationale behind the generated recommendations. The structure of this chapter is as follows: Section 3.1 introduces the definition of the recommendation problem, evaluation of recommendations,

and several commonly used approaches. Section 3.2 provides a review of research in explainable recommendations. We then introduce various methods for generating explainable recommendations, applying the concepts discussed in Chapter 2. For each method, we first present the corresponding models for generating recommendations, followed by the techniques employed for explaining the recommendations. In Section 3.7, we introduce existing methods for evaluating explanations in *Recommender Systems* (RSs). Section 3.8 summarizes some recurring challenges in this field and serves as a transition to the subsequent chapters, where we propose our solutions to address these issues.

3.1 Recommender systems

RSs are a class of software tools and techniques designed to provide personalized recommendations of items that are likely to be of interest to individual users (Ricci et al., 2021, Burke, 2007, Resnick and Varian, 1997). The "item" could be products (e.g. movies, commodities) or services (e.g. financial, medical). To formalize the recommendation, we introduce the following notations. $\mathcal{U}$ denotes the set of users in the RS; $\mathcal{I}$ denotes the set of item; $\mathcal{R}$ denotes the user-item interaction records, r_{ui} denotes the rating a user $u \in \mathcal{U}$ gives to an item $i \in \mathbf{I}$, where $r_{ui} \in \mathcal{S}$ and $\mathcal{S}$ can be explicit ratings given by users (e.g. $[1, 5]$) or binary interactions (e.g. $\mathbf{S} = \{\text{like, dislike}\}$); $\hat{r}_{(u,i)}$ denotes the predicted rating a user u may give to item i when $\mathcal{S}$ are explicit ratings given by users; rel_{ui} denotes the predicted relevance of item i to user u when $\mathcal{S}$ are binary interactions; $\mathcal{U}_i$ denotes the set of users that have rated item i; similarly, $\mathcal{I}_u$ denotes the items rated by user u; intuitively, $\mathcal{U}_{ij}$ denotes the users that have rated both item i and j, $\mathcal{I}_{uv}$ denotes the items that have been rated by both user u and v; $\mathcal{N}_i(u)$ denotes the set of similar users that have rated item i; $\mathcal{N}_u(i)$ denotes the set of similar items rated by user u.

3.1.1 Recommendation problem

The task of RSs is either to predict the rating that users may give to items or the relevance of items to users, which is called prediction of ratings and prediction of top-K items respectively (Nikolakopoulos et al., 2021). In particular, the rating prediction task gained widespread popularity after the Netflix Prize competition (Bennett et al., 2007).

Scenario 3.1 *The rating prediction scenario involves the task of a RS to learn a utility function $f : \mathcal{U} \times \mathcal{I} \to \mathcal{S}$ that predicts the rating $\hat{r}_{(u,i)}$ that a user u might assign to an item i.*

Typically, the values in $\mathcal{S}$ are continuous numbers within the range of $[1, 5]$, while $\hat{r}_{(u,i)}$ reflects the preferences of users towards items. Consequently, items are ranked based on their values of $\hat{r}_{(u,i)}$, and recommendations are made based on this order.

In many real-world scenarios, RSs only recommend a small subset of items that are likely to be of interest to the users, known as top-K recommendation (Deshpande and Karypis, 2004). In such cases, it is more suitable to predict whether a user will like an item or not.

Scenario 3.2 *In the top-K recommendation scenario, the goal of a RS is to learn a utility function $f : \mathcal{U} \times \mathcal{I} \to \mathcal{S}$ that predicts the relevance rel_{ui} of an item i to a user u, indicating how likely it is that the user will like the item.*

Algorithm 1: A general recommendation algorithm

Input : A user u, set of items $\mathcal{I}$,the number of items to be returned K
A prediction function $Predict$
Output: A list of K items

for $i \in \mathcal{I} \backslash \mathcal{I}_u$ **do**
  $\hat{r}_{(u,i)}$ or $rel_{ui} = Predict(u, i)$
Order $\mathcal{I} \backslash \mathcal{I}_u$ according to $\hat{r}_{(u,i)}$ or rel_{ui} in a descending manner
Return K first items in $\mathcal{I} \backslash \mathcal{I}_u$

The values in $\mathcal{S}$ fare typically binary variables, such as $like, dislike$. In both rating prediction and top-K recommendation scenarios, the main task of a RS is to learn a utility function f. Algorithm 1 (adapted from Negre (2017)) presents a general recommendation algorithm that can be applied to both scenarios described above. This approach is considered general because it does not require a specific method for data preprocessing, candidate recommendation generation, or ordering (i.e., the utility function f). Instead, these actions are considered as parameters of the approach, which can be instantiated in various ways to alter the method of calculating recommendations. By changing these parameters, the method of calculating recommendations is modified.

3.1.2 Evaluating recommendations

The quality of recommendations is a crucial aspect of explainable RSs. Therefore, we will review approaches for evaluating recommendations. Application designers face a wide array of RSs to choose from when integrating such a system into their applications. Consequently, selecting the most appropriate RSs largely depends on the designer's goals. Designers typically rely on experiments that evaluate the quality of recommendations to make informed decisions. Based on these evaluations, the designer can choose the best-performing RS that meets the structural constraints of the real-world application, such as the type, timeliness, and reliability of available data, as well as memory limitations (Gunawardana et al., 2021).

3.1.2.1 Evaluation strategies

From existing works, three evaluation strategies can be identified: offline, user studies and online experiments (Gunawardana et al., 2021).

Offline experiments: Offline experiments utilizing existing datasets and a protocol that models user behavior are frequently employed to estimate RS performance measures such as prediction accuracy. This approach assumes that user behavior during data collection will closely resemble user behavior when the RS is deployed, thus allowing for reliable comparisons. When conducting offline experiments, the data is typically divided into two parts: a training set used for learning the utility function and a test set for assessing the RS's performance. Two common conventions for splitting data are the *Random split* and *Chronological split* (Gunawardana et al., 2021). In the *Random split*, user interactions are shuffled, and a certain percentage is randomly selected for each set (e.g., 80% for training and 20% for testing) (Balog et al., 2019, Kim and Suh, 2019). If timestamps are available for interactions, an alternative method for splitting the dataset is to partition the interactions occurring before a specific time threshold into the training set and those occurring

afterward into the test set. This approach simulates a scenario in which the utility function is learned at the time threshold and applied to provide recommendations without incorporating subsequent interactions (Wu et al., 2022, Unger et al., 2020). A key assumption of offline evaluation is that user behavior during data collection will closely resemble user behavior during the RS's deployment. Consequently, the results obtained from this evaluation may not accurately reflect the actual performance in a real-world RS. Additionally, the available data might not be adequate for carrying out a comprehensive evaluation. In such situations, conducting user studies can provide valuable insights into the system's performance.

User studies: a typical user study involves recruiting a group of test subjects who are assigned to perform a set of interactions with the RS (Pu et al., 2012). Throughout the interaction, the subjects' behavior is quantitatively measured and recorded, capturing metrics such as task completion percentage, accuracy, and time taken to complete the task. Additionally, qualitative questions can be employed to assess recommendation quality by collecting subjective feedback about the user's experience (Gunawardana et al., 2021). These questions offer insights into aspects like user interface enjoyment and perceived ease of task completion, which may not be directly observable through quantitative measurements.

Online evaluation: an RS's true effectiveness depends on several factors, including the user's intent, personality, and context, such as their familiarity with certain items and the interface through which recommendations are presented. As a result, the most compelling evidence of the system's true value comes from online evaluations, where real users perform actual tasks using the system. One commonly used online evaluation method is *A/B testing* (Amatriain, 2013), which involves redirecting a small percentage of users to different alternative RSs and recording their interactions with these systems.

Online evaluations have the distinct advantage of reflecting real users' preferences towards recommendations. However, the cost of such evaluations can be prohibitive, and there is a risk of negatively impacting the user experience by testing underperforming approaches. As a solution, some companies adopt an offline-online testing process that leverages the strengths of both evaluation types (Amatriain, 2013). In this process, offline evaluations are used as an initial indicator to make informed decisions, followed by A/B testing to validate hypotheses previously tested through offline evaluations.

3.1.2.2 Evaluation criteria and metrics

When assessing the quality of recommendations, several criteria are typically considered, including coverage and serendipity (Ge et al., 2010), novelty (Konstan et al., 2006), diversity (Zhang and Hurley, 2008), accuracy (Gunawardana et al., 2021), and explainability (Zhang et al., 2020). Among these criteria, accuracy (Gunawardana et al., 2021) is the most extensively discussed in the literature. In this section, we will review the commonly used metrics for evaluating accuracy. Metrics for evaluating explainability (Zhang et al., 2020) will be discussed in Section 3.7. The metrics for evaluating accuracy can be further divided according to the recommendation scenarios: *rating prediction* and *top-K recommendations* (defined in Scenarios 3.1 and 3.2 in Section 3.1.1).

Rating prediction: Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE)

(Gunawardana et al., 2021) are perhaps the most widely used metrics for evaluating the accuracy of rating prediction, compared to MAE, RMSE disproportionately penalizes large errors.

$$RMSE = \sqrt{\frac{1}{|\mathcal{T}|} \sum_{(u,i) \in \mathcal{T}} (\hat{r}_{(u,i)} - r_{ui})^2} \tag{3.1}$$

$$MAE = \frac{1}{|\mathcal{T}|} \sum_{(u,i) \in \mathcal{T}} |\hat{r}_{(u,i)} - r_{ui}| \tag{3.2}$$

where $\mathcal{T}$ denotes the test set.

Top-K ranking: in this scenario, it is important to evaluate the relevance of recommendations. HR@K (Gunawardana et al., 2021) and NDCG@K (Järvelin and Kekäläinen, 2002) are two such metrics. HR@K indicates whether the tested items are among the top-K recommended items and NDCG@K quantifies the position of the tested items. The larger HR@K and NDCG@K are, the better the performance.

$$HR = \frac{1}{|\mathcal{T}|} \sum_{o=1}^{|\mathcal{T}|} hits(o) \tag{3.3}$$

where N is the total number of positive interactions in the test set; o is the oth interaction in the test set; $hits(o) = 1$ means that the target item (the item in the oth interaction in the test set) appears in the top-K recommendation list other $hits(o) = 0$ indicates that the target item is not in the top-K recommendation list.

HR@K only consider the presence of relevant item among the top-K recommendations without considering the rank of the relevant. Ideally, relevant items should be ranked on first. NDCG@K further quantifies the rank of relevant items and is defined in the following way:

For each user u in the test set $\mathcal{T}$, Discounted Cumulative Gain (DCG) quantifies the rank of relevant items:

$$DCG_u@K = \sum_{j=1}^{K} \frac{r_{uj}}{log_2(j+1)} \tag{3.4}$$

where rel_{uj} indicate the relevance of the jth item among the top-K recommendations to user u. $rel_{uj} = 1$ indicates that the jth item is relevant to user u; $rel_{uj} = 1$ indicates that the jth item is irrelevant to user u. Then NDCG@K is computed as follows:

$$NDCG@K = \frac{1}{|\mathcal{T}|} \sum_{(u,i) \in \mathcal{T}} \frac{DCG_u@K}{DCG_u^*@K} \tag{3.5}$$

where $DCG_u^*@K$ denotes the best $DCG_u@K$: the more relevant an item, the higher it rank among the top-K recommendations. In an ideal scenario, the most relevant item is ranked first, followed by the second most relevant item, and so on. Usually, for each user in the testset, only one item is considered to be relevant, therefore, Equation 3.5 can be further simplified as follows:

$$NDCG = \frac{1}{|\mathcal{T}|} \sum_{j=1}^{|\mathcal{T}|} \frac{1}{log_2(p_j+1)} \tag{3.6}$$

Table 3.1: Possible results of top-K recommendations

	Recommended	Not recommended
Relevant	True positive (tp)	False positive (fp)
Irrelevant	False negative (fn)	True negative (tn)

where p_j is the position of the target item in the top-K recommendation list (starting from 1) for the jth interaction in the testset.

Precision@K, Recall@K, and F1@K are often employed to evaluate the quality of top-K recommendations. Table 3.1 illustrates all possible outcomes for the top-K recommendations. True positive refers to recommended items that are truly relevant to users; False positive denotes recommended items that are actually irrelevant to users; False positive indicates items that are relevant to users but not recommended; lastly, True negative means items that are irrelevant and are not recommended. The frequently used metrics are computed as follows:

$$Precision = \frac{\#tp}{\#tp + \#fp} \tag{3.7}$$

$$Recall = \frac{\#tp}{\#tp + \#fn} \tag{3.8}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3.9}$$

Precision measures the percentage of relevant items among the items recommended; *Recall* measures the percentage of items recommended among the relevant items to users; *F1* score summarizes the *Precision-Recall* by the harmonic mean of them where *Precision* and *Recall* are equally weighted. It is noteworthy that additional measures exist for assessing the accuracy of RSs. These include the mean average precision (Baeza-Yates et al., 1999), the area under the ROC curve (Bradley, 1997), the expected reciprocal rank (Chapelle et al., 2009), and the average reciprocal hit rank (Deshpande and Karypis, 2004). For a more comprehensive examination of these measures, we direct readers to the survey by Gunawardana et al. (2021).

3.1.3 Recommendation approaches

Algorithm 1 (on Page 43) depicts a general approach for generating recommendations. Depending on how the utility function f is learned, the following approaches can be identified from existing works according to the dimension of the utility function (Adomavicius and Tuzhilin, 2005, Adomavicius et al., 2005).

- 2-D utility function: traditionally, RSs cater to applications that exclusively involve two types of entities: users and items. Therefore, the utility function is bi-dimensional ($f : \mathcal{U} \times \mathcal{I} \rightarrow \mathcal{S}$)
  - *Content-Based Recommender Systems* (CBRSs) rely on descriptive features to construct representations of both items and users, which are then utilized to generate personalized recommendations (Musto et al., 2012).

  - Collaborative filtering (CF) methods generate recommendations for users by analyzing their patterns of ratings or usage (such as purchases), without requiring any additional information about either the items or users (Koren et al., 2021). The neighborhood approach and latent factor models are the two main techniques that belong to this approach.
  - Hybrid approaches (Burke, 2002) usually combines the two approaches above.

- 3-D utility function: in numerous applications, from recommending restaurants to suggesting vacation packages, it may be insufficient to only account for users and items. The contextual information of users is a critical factor that should not be overlooked when generating recommendations. For example, a restaurant RS that incorporates users' companionship could suggest entirely different restaurants based on whether users are dining with colleagues or family. Similarly, when proposing vacation packages, a user's preferences can drastically differ depending on the season - summer or winter. Therefore, it's imperative to offer recommendations that are relevant and carefully tailored to these diverse contexts. *Context-Aware Recommender Systems* (CARSs) are designed to integrate contextual information into the recommendation process to suggest items to users based on specific circumstances (Adomavicius et al., 2021). In such cases, the utility of CARSs is multidimensional ($f : \mathcal{U} \times \mathcal{I} \times Context \rightarrow \mathcal{S}$).

The classification mentioned above is based on the work of Adomavicius and Tuzhilin (2005). It is important to note that there are alternative classifications, such as knowledge-based approaches (Burke, 2000), social network-based approaches (Papadimitriou et al., 2012), and deep-learning-based approaches (Liu and Wu, 2017), among others. However, these additional approaches can, to a certain extent, be integrated into the classification outlined above.

3.2 Explainable recommendations: an overview

In recent years, explainable recommendations (sometimes also called interpretable recommendations) have garnered considerable attention from both academia and industry (Zhang et al., 2020). From a human-computer interaction (HCI) perspective, explainable recommendations can be classified based on information resources or display styles. These classifications primarily include relevant user (Herlocker, 2000) or relevant item explanations (Sarwar et al., 2001), feature-based explanations (Vig et al., 2009, Rago et al., 2018, Zhong and Negre, 2022a), social relation explanations (Sharma and Cosley, 2013, Park et al., 2017), and auto-generated explanations by deep learning models (Costa et al., 2018).

From a machine learning perspective, the explanations generated strongly depend on the models utilized in RSs. For instance, recommendations generated by CBRSs typically employ item features for explanations (Vig et al., 2009, Rago et al., 2018), while recommendations generated by Collaborative Filtering (CF) generally rely on relevant user (Herlocker, 2000) or relevant item explanations (Sarwar et al., 2001). CARSs often explore the effects of contexts for explanations (Sato et al., 2018, Zhang et al., 2020, Zhong and Negre, 2022e).

Similar to AI, explainable recommendation research follows two strategies: (1) developing transparent models that allow for easy tracing of the recommendation generation process, and (2) retaining complex models and adopting post-hoc explanation techniques. This taxonomy is also in line with the one proposed by Friedrich and Zanker (2011). Deep learning recommenders (Costa

et al., 2018) can be explained by applying post-hoc explanation techniques (Zhong and Negre, 2022c, Nóbrega and Marinho, 2019, Peake and Wang, 2018).

In the following sections, we will introduce several typical recommender models and their corresponding explanation methods.

3.3 Content-based recommender systems

CBRSs operate under the premise that user preferences remain consistent over time, even as these preferences are constructed during the user's interactions with the system, such as in conversational settings. This is attributed to the fact that CBRSs recommend items that exhibit similarity to those the user has previously shown interest in, utilizing descriptive features of the items. In this section, we begin by examining the techniques employed for generating recommendations, followed by an introduction to feature explanation methods that are closely related to CBRSs.

3.3.1 CBRSs: generating recommendations

Typically, three components can be identified for CBRSs (Musto et al., 2012):

- **Content analyzer:** this component represents the content of items coming from information sources in a form suitable for the next processing steps. As a running example, a movie can be characterized by its features like actors, directors and genres.
- **Profile learner:** this module collects data to model user preferences and builds a user profile, which is a model that generalizes the observed data. User preferences are typically collected as ratings on a discrete scale (e.g. $[1, 5]$) or as binary variables (e.g. $\{like, dislike\}$).
- **Filtering Component:** this component predicts the utility $\hat{r}_{(u,i)}$, which is usually quantified as the similarity between u and i.

3.3.2 CBRSs: explaining recommendations

One advantage of CBRSs is that recommendations can be easily explained. Depending on the application scenario, content-based recommendations can be generated and explained by leveraging the features of items. Ferwerda et al. (2018) conducted a user study, and the results demonstrated that content-based explanations fostered user trust and satisfaction in the system. Similar conclusions were reported by Gedikli et al. (2014).

Symeonidis et al. (2009) employed movie features (e.g., director, category, and actor) to generate and explain recommendations. More specifically, recommendations were explained by showing users the relevance of each feature to their preferences. Similarly Vig et al. (2009) proposed Tagsplanations that explain movie recommendation by leveraging tags of movies. Rago et al. (2018, 2021) recently proposed explaining recommendations through an argumentative approach, where each feature is considered an argument that may support, neutralize, or oppose recommending items. We extend this idea by incorporating factorization-based methods (Zhong and Negre, 2022a), which will be detailed in Chapter 6. In these works, the process of generating items can be traced, making these methods part of the transparent models we introduce in Chapter 2.

Recommendations can also be explained after the recommendation process through a method known as post-hoc explanations, which are introduced in Section 2.5. Musto et al. (2019) employed

text summarization techniques to automatically generate summaries of relevant review excerpts to explain recommendations. Singh and Anand (2019) applied LIME (Ribeiro et al., 2016) to identify the most relevant words for document classification, where predictions were generated by neural networks. Similarly, Nóbrega and Marinho (2019) used LIME to determine important features that influenced predictions in a movie recommendation scenario, with recommendations generated by Factorization Machine (FM) (Rendle, 2010). Note that FM also falls under the category of latent factor models, a topic we will delve into in Section 3.4.2.1. Although post-hoc explanations offer more flexibility since they are independent of the recommendation process, they may not always accurately represent the original reasoning systems and could potentially be misleading (Rudin, 2019).

3.3.3 Summary of CBRSs

CBRSs offer several advantages, we list some of these advantages as following:

- **User-independence:** CBRSs leverage only the ratings provided by target users and the features of items, without relying on other users' rating records. As a result, CBRSs are less susceptible to issues of data sparsity.

- **Explainability:** recommendations can be explained (made more understandable to users) by providing a list of item features that they may prefer, even for those without expertise, as suggested by Vig et al. (2009).

CBRSs, however, have several shortcomings:

- **Limited content availability:** the effectiveness of CBRSs is directly linked to the quantity and quality of the features that are associated with items. However, gathering sufficient information about items to differentiate them can be challenging, as it often requires domain knowledge and not all necessary information may be readily available.

- **Over-specialization:** these systems recommend items that have high scores when compared to the user's profile, resulting in recommendations that are similar to those the user has already rated. This shortcoming is commonly referred to as the "serendipity problem" (Kotkov et al., 2016) which highlights the limited degree of novelty in recommendations generated by content-based systems. For instance, if a user has only rated horror movies, the system will recommend only similar movies.

- **User cold-start:** CBRSs require a sufficient amount of ratings in order to accurately model a user's profile. If a user has provided only a limited number of ratings, it becomes a challenging task to model their preferences.

The models we propose in Chapters 6 and 7 both leverage the content (termed as feature) of items for the natural explainability of CBRSs. However, we combine this approach with latent factor models to promote the performance of prediction.

3.4 Collaborative filtering (CF)

There are primarily two categories of collaborative filtering approaches: the neighborhood approaches and latent factor models (Koren et al., 2021). In this section, we will introduce both families of CF approaches along with their respective explanation techniques.

3.4.1 Neighborhood approaches

Neighborhood approaches, also known as memory-based methods, operate under the assumption that similar users rate items similarly, and similar items appeal to the same users. These approaches are referred to as *User-Based Collaborative Filtering* and *Item-Based Collaborative Filtering*, respectively (Koren et al., 2021, Deshpande and Karypis, 2004).

3.4.1.1 Neighborhood approaches: generating recommendations

User-Based Collaborative Filtering This approach predicts $\hat{r}_{(u,i)}$ by leveraging the ratings given to i by u's similar users. Formally, it is written as:

$$\hat{r}_{(u,i)} = \frac{\sum\limits_{v \in \mathcal{N}_i(u)} sim_{(u,v)} r_{vi}}{\sum\limits_{v \in \mathbf{N}_i(u)} |sim_{(u,v)}|} \tag{3.10}$$

where $sim_{(u,v)}$ is the similarity between user u and v. Usually, $sim_{(u,v)}$ can be computed by *Pearson Correlation* or *Cosine Similarity*.

Item-Based Collaborative Filtering This approach exploits the ratings given to similar items by the target user. The idea is formalized as follows:

$$\hat{r}_{(u,i)} = \frac{\sum\limits_{j \in \mathcal{N}_u(i)} sim_{(i,j)} r_{ui}}{\sum\limits_{j \in \mathbf{N}_u(i)} |sim_{(i,j)}|} \tag{3.11}$$

where $sim_{(i,j)}$ is the similarity between item i and j. Usually, $sim_{(i,j)}$ can be computed by *Pearson Correlation* or *Cosine Similarity*.

3.4.1.2 Neighborhood approaches: explaining recommendations

Explaining recommendations based on similar users or similar items can be relatively straightforward. For instance, "This item is similar to the items you interacted with before" (Linden et al., 2003) is an example of a relevant-item style explanation. Additionally, explanations can be presented to the user by highlighting the resemblance between the recommended item and other items they have previously enjoyed. Herlocker (2000) proposed explaining recommendations by aggregating the histogram of neighbors' ratings, which falls under the category of relevant-user style explanations. Generally, relevant-item explanations are more readily comprehensible to users, as they are acquainted with items they have previously interacted with (Zhang et al., 2020). As a result, these items can serve as plausible explanations for users. Conversely, relevant-user explanations might be less convincing, since the target user may lack knowledge about other "similar" users, which could potentially undermine the credibility of the explanations. Furthermore, revealing information about other users might also pose privacy issues.

3.4.1.3 Summary of Neighborhood approaches

Neighborhood approaches are widely used in the early days of RSs (Deshpande and Karypis, 2004, Linden et al., 2003, Herlocker, 2000), the merits of these approaches include the following:

- **Ease of implementation**, as we introduce above, neighborhood approaches assume that similar users rate items similarly, and similar items appeal to the same users. These ideas of predicting ratings are rather easy and intuitive, which leads to the next merit: simple to explain.
- **Simplicity to explain recommendations**, recommendations are generally explained by indicating the "similar users" or "similar items" (Herlocker et al., 2000, Linden et al., 2003).

There are limitations to neighborhood-based approaches:

- **Limited expressiveness** (Herlocker, 2000), as the performance of these methods heavily depends on the measurement of similarity between users (or items) and the number of the most similar users (or items) to be considered.
- **Sensitivity to data sparsity**, as the accuracy of these approaches deteriorates when there are limited available ratings. For instance, when the number of user-item interactions is limited, two users or items may not have any common ratings, which is especially the case when new users or new items are added to the system.

3.4.2 Latent factor models

Latent factor models, also known as model-based approaches (Koren et al., 2021), aim to discover underlying features that can be utilized to forecast ratings. Latent factor models comprise pLSA (Hofmann, 2004), Latent Dirichlet Allocation (Blei et al., 2003), and matrix factorization (Koren et al., 2009). Among these models, matrix factorization (MF) has gained popularity due to its attractive accuracy and scalability, particularly after the Netflix Prize competition (Bennett et al., 2007). In the following, we will compare several existing MF-based methods.

3.4.2.1 Latent factor models: generating recommendations

Vanilla MF (Koren et al., 2009) is the inner product of vectors that represent users and items. Each user is represented by a vector $\mathbf{p}_u \in \mathbb{R}^d$, each item is represented by a vector $\mathbf{q}_i \in \mathbb{R}^d$, and $\hat{r}_{(u,i)}$ is computed by the inner product of $\mathbf{p}_u$ and $\mathbf{q}_i$. The core idea of Vanilla MF is depicted in Figure 3.1.

$$\hat{r}_{(u,i)} = \mathbf{p}_u {\mathbf{q}_i}^T \tag{3.12}$$

Factorization machine (FM) (Rendle, 2010) takes into account user-item interactions and other features, such as users' contexts and items' attributes. It captures the second-order interactions of the vectors representing these features (see Equation 3.13), thereby enriching FM's expressiveness. However, interactions involving less relevant features may introduce noise, as all interactions share the same weight (Xiao et al., 2017).

$$\hat{y}_{FM}(\mathbf{X}) = w_0 + \sum_{j=1}^{n} w_j x_j + \sum_{j=1}^{n} \sum_{k=j+1}^{n} \mathbf{v}_j^T \mathbf{v}_k x_j x_k \tag{3.13}$$

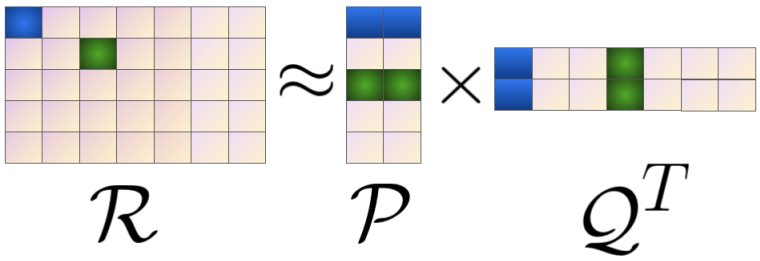


Figure 3.1: The core idea of Vanilla MF.

where $\mathbf{X} \in \mathbb{R}^n$ is the feature vector, n denotes the number of features, w_0 is the global bias, w_j is the bias of the j-th feature and $\mathbf{v}_j^T \mathbf{v}_k$ denotes the bias of interaction between j-th feature and k-th feature, $\mathbf{v}_j \in \mathbb{R}^d$ is the vector representing j-th feature.

Neural Factorization machine (NFM) (He and Chua, 2017) is essentially derived from FM (Rendle, 2010) and extends FM with a *Bilinear Interaction*: $f_{BI}(V_x) = \sum_{j=1}^{n} \sum_{k=j+1}^{n} x_i \mathbf{v}_j \odot \mathbf{v}_k x_k$, where $\odot$ denotes the element-wise product. The primary idea is that the *Bilinear Interaction* encodes the second-order interactions of vectors representing features. Subsequently, a stack of fully connected layers is incorporated to capture the high-order interactions of vectors, as shown in Equation 3.14.

$$\hat{y}_{NFM}(\boldsymbol{X}) = w_0 + \sum_{j=1}^{n} w_j x_j + \boldsymbol{h}^T \sigma_L(\boldsymbol{W}_L(\dots \sigma_1(\boldsymbol{W}_1 f_{BI}(V_x) + \boldsymbol{b}_1)\dots) + \boldsymbol{b}_L) \tag{3.14}$$

where L is the number of fully connected layers. It is trivial to see that by setting L to 0 and $\mathbf{h}^T$ to $\mathbf{1}$, FM is exactly recovered.

Attentional Factorization Machines (AFM) (Xiao et al., 2017) captures the second-order interactions of vectors representing users, items, and other features, similar to FM. Additionally, it models the attention of each interaction using a multi-layer perceptron (see Equation 3.15).

$$\hat{y}_{AFM}(\mathbf{X}) = w_0 + \sum_{j=1}^{n} w_j x_j + \mathbf{p}^T \sum_{j=1}^{n} \sum_{k=j+1}^{n} a_{jk}(\mathbf{v}_j \odot \mathbf{v}_k) x_j x_k \tag{3.15}$$

where a_{jk} is the importance of feature interaction, $\odot$ is the element-wise product. By setting $\mathbf{p}^T$ to $\mathbf{1}$ and a_{jk} to 1, FM is exactly recovered.

3.4.2.2 Latent factor models: explaining recommendations

Vanilla MF gained popularity following the Netflix Prize competition (Bennett et al., 2007) due to its simplicity and efficiency. However, it does not account for other information, such as item features, and using this straightforward operation to model user preferences may be inadequate when user-item interactions are sparse. Furthermore, the latent factors learned by Vanilla MF

lack explicit semantics, complicating the explanation of generated recommendations. Explanations for Vanilla MF recommendations typically involve identifying "relevant-user" or "relevant-item" (Zhang et al., 2020), but these explanations are not personalized and might not be intuitive to users. In particular, "relevant-user" explanations can be difficult to understand, as users often have no knowledge of other users' interactions (Zhang et al., 2020).

Abdollahi and Nasraoui (2017) introduced Explainable Matrix Factorization (EMF) to generate and explain recommendations. The fundamental assumption is that if an item i is explainable for a user u, then the representations of i and u in the latent space should be close to each other: $\boldsymbol{p}_u - \boldsymbol{q}_i \rightarrow 0$. More specifically, an explainability regularizer is added to the target function, forcing $\boldsymbol{p}_u$ and $\boldsymbol{q}_i$ to be close. As a result, recommendations can be explained by highlighting relevant-user or relevant-item. For example, a recommendation might be explained as, "This item is recommended because similar users also bought this item." Heckel et al. (2017) proposed factorizing the co-clustering information of users and items. Based on the co-clustering results, relevant-user or relevant-item explanations can be generated. Abdollahi and Nasraoui (2017), Heckel et al. (2017) designed models that are algorithmically transparent; however, these methods are neither simulatable nor decomposable.

Concurrently, Vanilla MF can be elucidated through a post-hoc method, signifying that Vanilla MF is regarded as a black-box. Wang et al. (2018) introduced an association rule mining technique to clarify Vanilla MF. For each user, the recommendation model uses the user's history as input and produces recommendations as output. The combined input and output create a transaction, and transactions from all users are used to derive association rules. These association rules can subsequently be applied to explain the recommendations generated by Vanilla MF if an item suggested by the black-box model can also be recommended using the association rules, this item is explainable by the rules, such as "$\{X \Rightarrow Y\}$: Because you watched X, we recommend Y".

FM captures the second-order interactions of the vectors representing these features (see Equation 3.13). However, interactions involving less relevant features may introduce noise, as all interactions share the same weight (Xiao et al., 2017). Explainability is another concern, as the explicit semantics of some interactions between features are unclear. FM is typically explained in a post-hoc way. For instance, Nóbrega and Marinho (2019) applied LIME (Ribeiro et al., 2016) to identify the most important item features, which falls under feature-based explanation.

NFM Although NFM can capture higher-order interactions of vectors, the explicit semantics of such interactions remain unclear. As a result, providing explanations for recommendations is a challenging task.

AFM Like FM and NFM, the explicit semantics of interactions between features remain unclear. Consequently, the learned attention is not beneficial for semantically explaining the predictions. Moreover, it has been argued that attention and explanation are distinct concepts (Jain and Wallace, 2019), suggesting that the effectiveness of the attention mechanism for generating explanations may have been overestimated.

3.4.2.3 Summary of latent factor models

Latent Factor Models contain a wide range of models and are a widely used approach in RSs, leveraging dimensionality reduction to identify latent features to explore observed user-item interactions. Here is a summary of their advantages:

- **Predictive Power:** latent Factor Models often exhibit strong predictive performance, capa-

ble of capturing complex patterns of user-item interactions. They are particularly effective at discovering the latent features that underpin user preferences, which can help with predictions (Koren et al., 2021).

- **Scalability:** latent Factor Models are computationally efficient and can be scaled to handle large datasets, making them well-suited for real-world applications with millions of users and items (Rendle, 2010, 2012).
- **Implicit Feedback Handling:** latent Factor Models can handle both explicit and implicit feedback, which increases their versatility in various application scenarios (Hu et al., 2008, Koren et al., 2021).

The following are the disadvantages of latent factor models:

- **Explainability of recommendations:** explaining recommendations generated by latent factor models is non-trivial. The latent features they identify do not have a clear, human-understandable meaning, which can make the recommendations they produce difficult to explain (Zhang et al., 2020, Goodman and Flaxman, 2017). Often, researchers turn to post-hoc explanation techniques to explain recommendations (Wang et al., 2018, Nóbrega and Marinho, 2019, Zhong and Negre, 2022c).
- **Cold Start Problem:** these models struggle with new users or items that have little to no interaction data, a challenge known as the cold start problem (Schein et al., 2002).
- **Over-Specialization:** these models may overfit to the observed user-item interactions, leading to a lack of diversity in recommendations. They may recommend too many similar items, failing to explore user preferences broadly (Celma and Herrera, 2008).

3.5 Hybrid approaches

Hybrid approaches (Burke, 2002) have been proposed to combine the advantages of CBRSs and CF-based approaches while minimizing the disadvantages of both. This blended approach aims to mitigate the limitations of individual methods and increase the overall effectiveness and accuracy of recommendations.

3.5.1 Hybrid approaches: generating recommendations

In hybrid RSs, recommendations can be made based on a weighted combination or sequencing of different recommendation techniques. For instance, a system could apply collaborative filtering to find users with similar preferences and then use content-based filtering to find items similar to those the user has rated highly before (Burke, 2002). As summarized by Jannach et al. (2010), there are primarily three types of hybrid approaches:

- Monolithic exploitation of different features: this approach integrates features or knowledge sources from various paradigms into a single, cohesive recommendation component.
- Parallel utilization of multiple systems: this method involves the concurrent application of several existing recommendation techniques, with their outputs combined using a weighting or voting scheme.

- Pipelined invocation of different systems: this strategy consists of a sequence of recommendation techniques executed consecutively. In this pipeline, the recommendations of each subsequent technique are informed by its predecessor, suggesting that one RS preprocesses input for the subsequent system.

3.5.2 Hybrid approaches: explaining recommendations

Providing explanations for hybrid RSs can be challenging due to the mixture of techniques. The explanations typically need to incorporate reasoning from each method used. For example, a system might explain that a recommendation was made because it was liked by similar users (reflecting the collaborative filtering component) and because it matches the user's preferences for certain item features (reflecting the content-based component) (Tintarev and Masthoff, 2007). For instance, Rago et al. (2018, 2021) combined CF and content-based approaches, enabling the use of similar users and item features to explain recommendations in an argumentative way.

3.5.3 Summary of hybrid approaches

Hybrid approaches, which integrate multiple information sources for generating recommendations, inherently inherit the diverse array of advantages and disadvantages associated with these sources (see Sections 3.3.3, 3.4.1.3 and 3.4.2.3).

One of the main advantages of hybrid approaches is their ability to compensate for the weaknesses of individual methods, improving the accuracy and coverage of recommendations. For instance, they can help mitigate the cold start problem that's common in CF, where the system struggles to make accurate recommendations for new users or items due to a lack of historical data. By incorporating content-based components, which don't rely on past user-item interactions, hybrid systems can still make relevant recommendations in these scenarios (Burke, 2002).

The main disadvantage of hybrid approaches is their complexity. Implementing and maintaining multiple recommendation techniques can be challenging, requiring more computational resources and potentially leading to longer response times. Furthermore, effectively combining results from different methods is non-trivial, as it often requires careful tuning or machine learning methods. Also, generating user-friendly explanations in hybrid systems is more complex due to the need to combine reasoning from multiple techniques (Tintarev and Masthoff, 2007).

3.6 Context-aware recommender system

CARSs assume that users' preferences are dynamic and change over different contexts. Incorporating such information in RSs helps improve recommendation quality, as more personalized propositions can be generated (Adomavicius et al., 2005).

3.6.1 CARSs: generating recommendations

In this section, we introduce how CARSs generate recommendations: what is context? and how CARSs use context to generate more personalized recommendations.

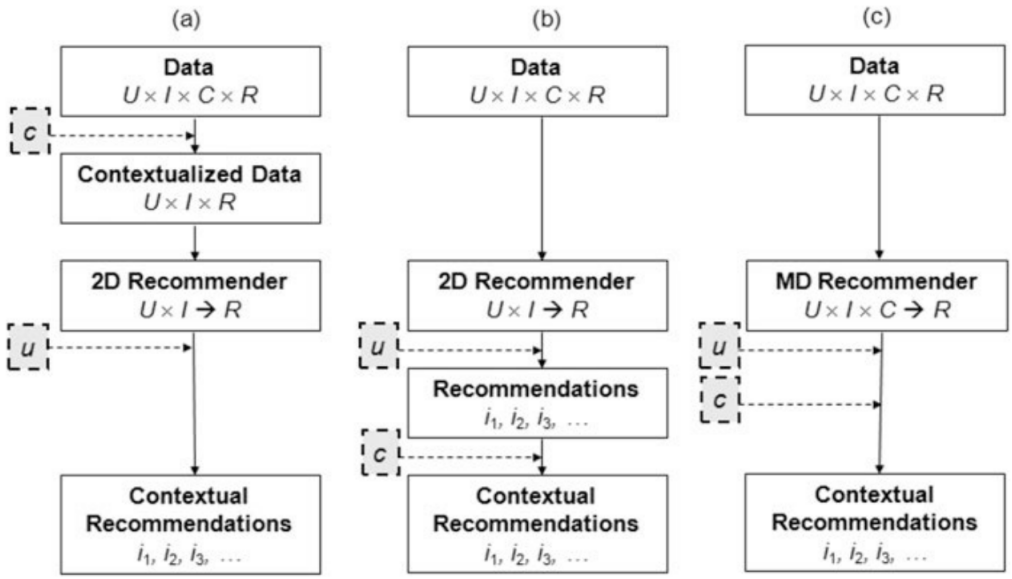


Figure 3.2: Three paradigms for incorporating contexts into RSs. Note that C refers to contexts (Adomavicius and Tuzhilin, 2011).

3.6.1.1 What is context?

Context has been investigated in various research disciplines, including computer science (especially in *Artificial Intelligence* (AI) and ubiquitous computing), cognitive science, and linguistics, as it is a multifaceted concept (Dourish, 2004, Bazire and Brézillon, 2005). However, there is no common consensus on the definition of context among researchers from various domains, such as psychology, AI, and information retrieval. Context has been studied in RSs from different viewpoints, depending on the application domain (Adomavicius et al., 2021). Abowd et al. (1999) defined context as any information that can be used to characterize the situation of an entity. For example, in a movie RS, factors such as time (e.g., workday, weekend), companion (e.g., friends, lovers, or alone), and mood (happy, sad, or neutral) can influence a user's choice of movies. In a trip RS, weather (sunny, rainy), season (spring, summer), and distance (near, far) can influence a user's choice of destination. The "time", "companion" and "season" are contextual factors and each contextual factor, it can have several possible values, these values are contextual conditions and the contextual conditions compose contextual situations.

3.6.1.2 Paradigms for incorporating context

In 2-D RSs, the utility function is bi-dimensional ($f : \mathcal{U} \times \mathcal{I} \rightarrow \mathcal{S}$), while in CARSs it is multidimensional ($f : \mathcal{U} \times \mathcal{I} \times Context \rightarrow \mathcal{S}$). There are mainly three approaches for incorporating context into RSs: *pre-filtering*, *post-filtering* and *contextual modeling* (Adomavicius et al., 2021), which is illustrated in Figure 3.2.

Pre-filtering: in this paradigm, context is used to select a set of relevant data that is then used to learn any canonical two-dimensional RSs (e.g., MF). Adomavicius et al. (2005) proposed a reduction-based approach in which a data filtering query is constructed using the specified context. Baltrunas and Ricci (2009, 2014) propose a different approach to contextual pre-filtering through

the item splitting technique, which involves splitting each item into multiple fictitious items based on the different contexts in which they can be consumed. Baltrunas and Amatriain (2009) propose the concept of micro-profiling (also known as user splitting) in which a user profile is divided into multiple sub-profiles, each representing the user's preferences in a specific context. Instead of using a single user model, the predictions are made using these contextual micro-profiles. For more surveys on the *pre-filtering* approach please refer to Adomavicius et al. (2021).

Post-filtering: In the post-filtering paradigm, as illustrated in Figure 3.2, context-free approaches are initially used to predict utility (refer to Section 3.4 and 3.3). Subsequently, context is employed to adjust the recommendation list either by filtering out irrelevant recommendations for a given context or re-ranking the recommendation list based on a particular context (Adomavicius and Tuzhilin, 2005, Panniello et al., 2009, Villegas et al., 2018).

Contextual modeling: the contextual modeling approach integrates context directly into the recommendation function as an explicit predictor of a user's utility for an item. This essentially involves using predictive models (such as decision trees, regressions, probabilistic models, deep learning models, or other techniques) or heuristic calculations that incorporate context in addition to the user and item data. Various contextual modeling methods have been proposed. CAMF-C (Baltrunas et al., 2011b), FM (Rendle, 2010), AFM (Xiao et al., 2017) are factorization-based methods. More recently, there have been attempts that apply neural networks to better capture users' preferences under different contextual situations (Unger et al., 2016, 2020). Explaining recommendations generated by these methods is non-trivial, we, therefore, turn to argumentation, which is intuitively understandable and explainable.

Contextual modeling, which has demonstrated its proficiency in enhancing the performance of recommendations in comparison to alternative pre- and post-filtering methods in numerous instances, stands out among the three paradigms (Adomavicius et al., 2021, Panniello et al., 2014). As such, it has gained substantial popularity as a preferred approach for integrating context into RSs. The model we propose in Chapter 6 falls into contextual modeling.

3.6.2 CARSs: explaining recommendations

Users' preferences can vary across contexts, and CARSs can capture these differences (Adomavicius and Tuzhilin, 2011). Naturally, this concept can be applied when explaining recommendations, leading to context-aware explanations (Zhang et al., 2020, Zhong and Negre, 2022e, Baltrunas et al., 2011a, Sato et al., 2018). To the best of our knowledge, Baltrunas et al. (2011a) were the first to explore context-aware explanations. Through a user study, they demonstrated that context-aware explanations were preferred over context-free explanations in travel recommendations. However, the user study only involved 20 participants, which may limit the statistical credibility of the conclusion. Misztal and Indurkhya (2015) proposed a prototype to explain recommendations by identifying the most relevant context for each movie type. Following this concept, explanations for recommendations of the same type are always the same, lacking personalization. Sato et al. (2018) modeled users' preferences under various contextual situations using factorization machines (Rendle, 2010). Recommendations can be explained through item-context pairs. Their user study results indicated that incorporating contextual information in explanations enhances user-perceived persuasiveness and usefulness.

Context-aware explanations can also be generated using a post-hoc approach. Musto et al. (2020) proposed post-hoc context-aware natural language explanations for recommendations produced by a CBRS. Participants were asked to provide basic demographic data to indicate their

interests in movies. Explanations were generated by leveraging distributional semantics in reviews posted by other users. For instance, if a participant indicates that they are watching a movie with their partner and a movie is recommended by the content-based recommender, the system can analyze the distributional semantics in other users' reviews about the movie. If the item description "romantic ending" frequently appears in the context of "watching with a partner", then an explanation such as "We recommend this movie for you and your partner because it is a good choice, as it has a romantic ending" can be generated. Their user study suggested that users favored context-aware explanations over context-free ones.

3.6.3 Summary of CARSs

We now summarize the advantages and disadvantages of CARSs. The following are the advantages of CARSs:

- **Increased personalization and a better understanding of user preferences:** by leveraging the context, CARSs gain a deeper understanding of user preferences, which can vary based on different situations or contexts. Therefore, CARSs can provide more personalized and relevant recommendations (Adomavicius et al., 2021).
- **Improved User Experience:** CARSs can enhance user experience by delivering timely and situation-specific recommendations, contributing to user satisfaction and system trustworthiness (Baltrunas and Ricci, 2009, Adomavicius et al., 2021).

However, there are some challenges that remain to be resolved:

- **Increased Complexity:** incorporating context adds an additional dimension to the recommendation problem, making the system more complex. It may also require more computational resources (Baltrunas and Ricci, 2009).
- **Data Sparsity:** with more dimensions to consider, there might not be enough data available for each specific context-user-item combination, which can lead to data sparsity issues and affect the quality of recommendations (Adomavicius and Tuzhilin, 2011).
- **Context-aware explanations:** various context-aware models have been proposed, however, context-aware explanation is less explored (Zhang et al., 2020). Current research focusing on context-aware explanations primarily revolves around conceptualizing prototypes and conducting user studies. These pioneering efforts underscore the significance of providing context-aware explanations for recommendations; however, few have actually constructed CARSs capable of generating such context-aware explanations. In conclusion, the domain of context-aware explanations in RSs merits deeper investigation.

3.7 Evaluating explanations in RSs

Explainable recommendations address the rationale behind item suggestions, promoting transparency, effectiveness, efficiency, persuasiveness, trust, satisfaction, and scrutability (Tintarev and Masthoff, 2022) (Ultimately, these goals also address the second research question: **"Why *explainability* and *interpretability* are remarkably demanded in RSs?"**). As such, the evaluation

of these explanations is critical for gauging model appropriateness and aiding in model selection. Nevertheless, this task poses significant challenges, primarily due to the complexities in securing ground truth and emulating human emotional responses (Chen et al., 2022). Over the past few decades, numerous promising evaluation strategies have been developed to address these challenges. Notably, recommendation explanations may target various audiences, such as users, providers, and model designers. Different evaluation perspectives are needed for these distinct targets. In this thesis, our primary focus is on explanations aimed at non-expert users. Consequently, the evaluation is also specifically tailored for non-expert users' explanations.

The objectives of explanations include enhancing transparency, effectiveness, efficiency, persuasiveness, trust, satisfaction, and scrutability (Tintarev and Masthoff, 2022). These goals are commonly employed as metrics for evaluating the quality of explanations in RSs (Tintarev and Masthoff, 2015, Gedikli et al., 2014). In the following, we examine existing research that assesses explanation quality through these objectives.

- Efficiency: in RSs, an explanation is deemed efficient if it assists users in more rapidly selecting the best-recommended item (Tintarev and Masthoff, 2015). The evaluation of efficiency depends on the methods employed by the RS. For instance, in knowledge-based RSs, explanations can be generated by illustrating the trade-offs between item properties, and the efficiency of these explanations can be gauged by the number of interactions required for users to make a decision. In the case of conversational RSs, efficiency can be measured by the total number of dialogue steps needed before a recommendation is accepted (McCarthy et al., 2005). In other situations, efficiency is typically assessed by the amount of time a user takes to make a decision, such as the time required to select an item or provide a rating (Gedikli et al., 2014).

- Effectiveness: an effective explanation in RSs aids users in making better decisions, particularly in evaluating the quality of the recommended items (Tintarev and Masthoff, 2015). Generally, effectiveness can be assessed by measuring the extent to which users' preferences for recommended items change before and after consumption (Bilgic and Mooney, 2005). In the study conducted by Bilgic and Mooney (Bilgic and Mooney, 2005), users rated a book twice: first, when presented with an explanation, and second, after having read the book. In this scenario, the smaller the difference between the two ratings, the more effective the explanation is considered to be.

- Persuasiveness: recall that the persuasiveness of explanations in RSs refers to the ability to convince users to try or consume the recommended items. A common evaluation approach approximates persuasiveness by calculating the difference between users' liking of the recommended item before and after consumption, as mentioned by Bilgic and Mooney (2005). A positive difference implies high persuasiveness, while a small or negative difference suggests low persuasiveness. A more intuitive method is proposed by Cosley et al. (2003), where users are asked to rate an item twice: first without an explanation and then with an explanation. Persuasiveness is measured by the difference between the two ratings. To the best of our knowledge, no existing work has compared these two approaches, which could be an interesting future research direction. Both approaches require users to rate the same items twice, a design known as a *Within-Subject* study (Keren, 2014). An alternative strategy, known as a *Between-Subject* study (Keren, 2014), would involve randomly assigning users to two groups, with one group receiving explanations and the other group not. Both groups would

be asked to rate the same items. By examining the differences in ratings between the two groups, the persuasiveness of explanations can be quantified. Furthermore, since explaining is a human-centered process, it could be beneficial to assess whether explanations are perceived as efficient, effective (Vig et al., 2009), and persuasive (Herlocker et al., 2000) through self-reports from users.

- Trust: defined as users' perceived confidence in RSs (Chen and Pu, 2005). Explanations that clearly articulate why a particular item was recommended can make the system appear more transparent and trustworthy (Herlocker et al., 2000). Likert-scale questionnaires are a common method for assessing trust. These questionnaires provide quantitative data that can be easily analyzed to discern trends or patterns in user trust. For instance, questions like "I trust the recommendations given by this system" could be rated on a scale of 1 to 5, with 5 being "Strongly Agree" and 1 being "Strongly Disagree". However, it is worth noting that while Likert scales provide an easy-to-interpret metric, they may not capture all the nuances of user trust, such as cultural or individual variations (Wang and Benbasat, 2007).

- Transparency in RSs encompasses two aspects (Gedikli et al., 2014): (i) objective transparency, which involves explaining the system's actual algorithm; and (ii) user-perceived transparency, which concerns the degree to which users perceive a system as transparent. It may be more appropriate to present user-oriented explanations rather than actual algorithms (Vig et al., 2009) for several reasons: algorithms can be too complex for users to comprehend, and they may contain intellectual properties that should be protected. Consequently, transparency is typically assessed via Likert-scale questionnaires, with participants directly asked about their perception of a system's transparency when accompanied by the explanation being evaluated.

- Satisfaction: users' loyalty (McNee et al., 2003) or the likelihood of using the system to find items of interest (Cramer et al., 2008) can serve as indirect indicators of satisfaction. Greater loyalty and a higher likelihood of users utilizing the system for search tasks imply increased satisfaction. Satisfaction can also be assessed directly by asking users whether they prefer the system with explanations or not. Typically, these evaluations are conducted using Likert-scale questionnaires.

- Scrutability: refers to the ability of users to inform the system when recommendations are deemed inappropriate. It is usually evaluated through user studies, as demonstrated in various research (Balog et al., 2019, Tintarev and Masthoff, 2022, Wasinger et al., 2013).

Various predefined proxies have been employed for evaluating explanations in RSs, such as Mean Explainability Precision (MEP) and Mean Explainability Recall (MER) defined by Abdollahi and Nasraoui (2017), Model Fidelity introduced by Peake and Wang (2018), and Probability of Necessity (PN) and Probability of Sufficiency (PS) for assessing counterfactual explanations (Tan et al., 2021). When explanations take the form of text, evaluation metrics like BLEU (Papineni et al., 2002) and ROUGE (ROUGE, 2004) scores can be used.

In the previous chapter, we introduced three levels of approaches for evaluating explanations in the context of AI: application-grounded evaluation, which involves real-world users; human-grounded evaluation, which necessitates simplified tasks; and functionally-grounded evaluation, which uses predefined quantitative proxies. Table 3.2 summarizes some existing methods for evaluating explanations in RSs and classifies them according to the three levels of evaluation. Due to

the limit of table length, we list only the representative for each category. Some works actually adopt more than one metric. For instance, Gedikli et al. (2014) evaluated and compared different explanation methods in terms of efficiency, effectiveness, persuasiveness, trust, satisfaction, and transparency. Actually, there are fewer studies conducting application-grounded evaluation compared to those employing user studies and proxies-based evaluation. The main reason is that carrying out real-world evaluations can be expensive, and at times, inappropriate explanations may undermine the system owners' benefits (Tintarev and Masthoff, 2022, Chen et al., 2022). As a result, most existing works assess the quality of explanations in RSs through user studies or predefined proxies. Note that Table 3.2 only presents some representative works or those that will be mentioned later in this thesis. For a more comprehensive survey, please refer to Chen et al. (2022) and Tintarev and Masthoff (2022).

3.8 Conclusions

Throughout this chapter, we have delved into the realm of explainable recommendations, a crucial area of research that aims to generate high-quality recommendations while providing understandable explanations for users. We have systematically reviewed the state-of-the-art in explainable recommendations, including different recommender approaches, such as content-based, collaborative filtering, hybrid, and context-aware methods. In addition, we have discussed the evaluation strategies, criteria, and metrics for recommendations and explanations in RSs. This comprehensive review has shed light on the key challenges that the field of explainable recommendations faces:

1 Lack of consensus on the definition and the use of some notions related to explanations;

2 The comprise between the accuracy and transparency of RSs is still worth more effort;

3 New techniques to incorporate user feedback and preferences to create context-aware recommendations and context-aware explanations to justify such recommendations worth further exploration;

4 Given the comprise between accuracy and transparency, retaining complex models does help to achieve higher performances. Is it possible to make use of post-hoc explanation methods to explain any RSs while avoiding the limits of these methods?

5 Evaluating the qualities of explanations is another challenging task.

In the following chapters of this thesis, we propose solutions to address some of the challenges and contribute to the ongoing progress in the field. Chapter 4 tackles the first challenge and addresses the ambiguity of commonly used terms, introduces *Interpret/Explain Schema* (IES) that helps clarify these terms, and constructs a *General Framework for Generating Explanations* (GFGE) in the context of AI. Chapter 6 tackles the comprise between accuracy and transparency, where we instantiate GFGE in a model-intrinsic explanation scenario. Concretely, we develop a model that can generate accurate recommendations and devise argumentative explanations to justify such recommendations. Chapter 7 addresses the third challenge and presents an extension of the model introduced in Chapter 6, we combine argumentation frameworks with context-aware methods. Chapter 8 concentrates on the fourth challenge and instantiates GFGE in a post-hoc explanation scenario. Chapter 9 leverages the fifth challenge by comparing the two explanation scenarios and presenting the preliminary results of a user study that compares different explanation methods.

Table 3.2: Evaluation approaches for explanations in RSs

Metrics	Application-grounded		Human-grounded		Functionally-grounded
	Subjective	Objective	Subjective	Objective	Objective
Efficiency			(Balog and Radlinski, 2020)	(McCarthy et al., 2005)	
Effectiveness			(Vig et al., 2009)	(Bilgic and Mooney, 2005)	
Persuasiveness	(Zhang et al., 2014)	(Xian et al., 2021)	(Kouki et al., 2019)	(Bilgic and Mooney, 2005)	
Trust			(Balog and Radlinski, 2020)		
Transparency			(Gedikli et al., 2014)	(Vig et al., 2009)	
Satisfaction			(Balog and Radlinski, 2020)	(McNee et al., 2003)	
Scrutability			(Li et al., 2020b)	(Balog et al., 2019)	(Tan et al., 2021)
BLEU and ROUGE scores					(Papineni et al., 2002, ROUGE, 2004)
PN and PS					(Tan et al., 2021)
Model Fidelity					(Zhong and Negre, 2022c)
MEP and MER					(Abdollahi and Nasraoui, 2017)
Length					(Kaffes et al., 2021)

Part II

Theoretical contribution: a general framework for generating explanations

Chapter 4

General framework for generating explanations

Contents

4.1 The importance of generality 65
4.2 A general definition 66
4.3 *Interpret/Explain Schema* in AI system 68
4.3.1 AI: to interpret or to explain? 68
4.3.2 Definitions in the context of AI 69
4.3.3 Discussions 70
4.3.4 *General framework for generating explanations* 72
4.4 Conclusions 75

In this chapter, we focus on developing the *General Framework for Generating Explanations* (GFGE) within the context of *Artificial Intelligence* (AI), which is also applicable to *Recommender Systems* (RSs) because RSs is an important branch of AI. We begin by revisiting certain concepts discussed in Chapter 2 and Chapter 3 to emphasize the significance of using general terms consistently when developing explainable AI techniques. Subsequently, we present an *Interpret/Explain Schema* (IES) (Zhong and Negre, 2021) designed to clarify the often-used yet ambiguously-defined notions. Ultimately, this chapter seeks to address the first research question outlined in Chapter 1: **"What are the differences between *interpretability* and *explainability* in the context of AI, and more specifically in the context of RSs?"** We will further clarify "What does *explainability* and *interpretability* mean respectively?" and "To whom *explainability* and *interpretability* is concerned?"

The publication related to this chapter is:

- Zhong, J. and Negre, E. (2021). Ai: To interpret or to explain? In Congrès Inforsid(INFormatique des ORganisations et Systèmes d'Information et de Décision).

4.1 The importance of generality

As introduced in Chapter 2, AI technologies have become increasingly prevalent, profoundly impacting various aspects of human life, including RSs (Zhang et al., 2020). However, concerns

about AI's effectiveness and reliability persist, especially in high-stakes domains like healthcare and financial services, which drive commercial interest. Furthermore, the European Union requires decision support systems to provide explanations under the "right to explanation" principle (Voigt and Von dem Bussche, 2017), adding a regulatory dimension to the demand for explainable AI. Additionally, understanding the reasoning behind AI systems aids designers in debugging and improving models, providing a technical incentive for the development of AI systems that deliver accurate results and justifiable explanations. Researchers often claim that their models are either interpretable or explainable, yet precise definitions remain elusive. Some researchers distinguish between the two (Lipton, 2018, Doshi-Velez and Kim, 2017, Montavon et al., 2018), while others use them interchangeably (Miller, 2019, Molnar, 2020, Du et al., 2019, Arrieta et al., 2020).

Despite extensive efforts, the AI research community has not yet reached a consensus on the definitions of *interpretability* and *explainability*. Lipton (2018) categorizes interpretable models into transparent models, which can be easily understood by humans, and those that provide post-hoc explanations, which offer explanations after generating results without necessarily elucidating their inner workings. Lipton distinguishes between *interpretability* and *explainability*, asserting that *interpretability* serves as the foundation for *explainability*. Gilpin et al. (2018) also argue that *interpretability* and *explainability* should be differentiated, as explainable models are inherently interpretable, but not all interpretable models are explainable. However, some researchers use the terms *interpretability* and *explainability* interchangeably (Miller, 2019, Molnar, 2020, Du et al., 2019, Arrieta et al., 2020), further complicating the issue, as depicted in Figure 2.1. The lack of clear definitions obstructs the efforts to define, design, and evaluate AI systems capable of providing meaningful explanations (Doshi-Velez and Kim, 2017).

The lack of consensus on definitions and criteria for *interpretability* and *explainability* has led to a diverse landscape of approaches in the field. Some focus on creating inherently interpretable models, while others work on developing techniques for extracting explanations from existing models. These approaches include local explanation methods, such as LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017), which attempt to provide explanations for individual predictions. Additionally, global explanation methods, like feature importance rankings and rule extraction techniques, seek to provide insights into the overall behavior of models (Guidotti et al., 2018). The ongoing research and development of these methods underscore the importance of achieving a deeper understanding of *interpretability* and *explainability* in AI, as well as addressing the current challenges and limitations in their definitions and applications.

4.2 A general definition

The ambiguity surrounding the use of *interpretability* and *explainability* may stem from their linguistic origins. According to the Merriam-Webster Dictionary, "interpret"[5] means to explain or present the meaning of something in understandable terms, while "explain"[6] means to make something plain or understandable, or to give the reason for or cause of something. Although the two words have similar meanings, which may have led to their interchangeable usage, subtle differences do exist between them. For instance, interpreting certain articles in the General Data Protection Regulation (GDPR) might involve understanding them as the "right to explanations," while explaining the GDPR could entail providing reasons for the specific regulations in place. It appears

[5] https://www.merriam-webster.com/dictionary/interpret
[6] https://www.merriam-webster.com/dictionary/explain

that interpreting answers a "what" type question, while explaining tends to address a "why" type question. Consequently, if something is interpretable, it has an assigned meaning that can be presented in understandable terms; if something is explainable, it can be made comprehensible.

In Section 2.3, we reviewed existing literature on the definition of explanation in the context of AI. The concept of explanation has roots in various disciplines, such as philosophy, cognitive and social psychology, and cognitive science. Lewis (1986) posited that explaining an event involves providing information about its causal history. Josephson and Josephson (1996) defined explanation as assigning causal responsibility, while Malle (2006) emphasized the social interaction aspect of explanation. Lombrozo (2006) and Miller (2019) observed that explanation can function both as a process and a result, serving as an answer to a "why" question or as a process involving abductive inference.

Lombrozo and Miller's assertions align with concepts in knowledge management (Despres and Chauvel, 1999, Arduin et al., 2015), where individuals possess unique interpretative frameworks (Jones et al., 2011) or mental models (Tsuchiya, 1993) for filtering and storing information. Arduin et al. (2015) emphasized the importance of interpretation in knowledge management through *sense-reading* (interpreting information) and *sense-giving* (assigning meaning to created information), as illustrated in Figure 2.2 on Page 32. The combination of these two processes results in an explanation. According to Miller (2019), explanations are inherently social and involve conversations. From a philosophical standpoint, interpretation is a subjective action, while explanation involves interaction.

Integrating the linguistic definitions and interpretative framework, we provide the following definition for '*explainability*' from a general and philosophical perspective:

Definition 4.1 *Explainability: the ability to make an event understandable; the ability to give the reason or cause of an event.*

Existing literature on explanations presents us with three key dimensions: (1) a focus on either causal explanations (Lewis, 1986, Josephson and Josephson, 1996) or non-causal explanations (Miller, 2019, Malle, 2006, Lombrozo, 2006); (2) from a cognitive science perspective, humans' decision-making process can be categorized into two types: decisions based on explicit reasoning, which can be readily explained, and decisions where explanations are sought post-hoc (Miller, 2019); (3) Given that explanations often involve human interactions (Lewis, 1986, Miller, 2019), the social aspects of explanations are significant and should be given due attention. We will now discuss how our proposed Definition 4.1 aligns with these three dimensions:

Causal or non-causal? The definition provided above takes cues from the works of Lewis (1986), Josephson and Josephson (1996), Malle (2006), Lombrozo (2006), and Miller (2019). When it comes to explanations, Lewis (1986) and Josephson and Josephson (1996) underscore the significance of causality, while Miller (2019), Malle (2006) and Lombrozo (2006) highlight the role of interactions and knowledge transfer and suggest that explanations need not always identify the causes of an event. Our definition of '*explainability*' encompasses both causal and non-causal explanations. The phrase "give the reason or cause of an event" is reflective of a causal explanation, while making an event "understandable" pertains to a non-causal explanation.

Intrinsic or post-hoc? Our definition incorporates the cognitive processes underlying human decision-making. At times, humans use thoughtful, systematic reasoning to make decisions and are able to clearly express the reasoning behind their choices - aligning with the "give the reason or cause of an event" aspect of our definition. However, in other circumstances, humans may

make decisions first and then seek justifications to support or legitimize their actions (Miller, 2019, Lipton, 2018). This corresponds to the "make an event understandable" part of our definition. These cognitive processes potentially explain why two strategies have evolved: the use of model-intrinsic explanations and the generation of post-hoc explanations. For a more comprehensive comparison, please refer to Sections 2.5 and 2.7.

Social aspects: our definition also underscores the social dimensions of explanations. The phrase "make an event understandable" and the articulation of reasoning behind choices both involve interactions between the explainer and the explainee, indicating a transfer of knowledge. This aligns with the perspectives of Miller (2019) and Lombrozo (2006). As such, our concept of *explainability* naturally paves the way for our definition of *interpretability*.

As for *interpretability*, we give the following definition:

Definition 4.2 Interpretability*: the degree to which an observer can understand the meaning of an event.*

The definition of *interpretability* is based on our concept of *explainability* and is inspired by knowledge management literature (Despres and Chauvel, 1999, Arduin et al., 2015). Humans utilize a unique interpretative framework (Jones et al., 2011) or mental model (Tsuchiya, 1993) to filter, process, and store information. This framework influences how individuals interact with the world, create knowledge, and lead to varied interpretations of the same data. Arduin et al. (2015) emphasize the significance of interpretation in knowledge management, referring to the process of making sense of information as *sense-reading* (Tsuchiya, 1993). When information is shared, each recipient interprets it based on their own mental model. The sender, through their interpretative framework, assigns meaning to the information in a process known as *sense-giving* (Tsuchiya, 1993). The combination of *sense-reading* and *sense-giving* results in individualized knowledge creation and forms an explanation, as illustrated in Figure 2.2 on Page 32. In summary, interpretation is a subjective action, while explanation involves interactions.

4.3 *Interpret/Explain Schema* in AI system

Considering that the goal of AI technologies is to emulate human intelligence within machines, thereby enabling them to mimic human thought processes and actions, it is logical to define *interpretability* and *explainability* within the context of AI, drawing from the concepts of interpretative frameworks and mental models.

4.3.1 AI: to interpret or to explain?

An AI system typically comprises three elements, aside from human involvement: input data, the model, and the produced results, as illustrated in Figure 4.1. The system's outcomes are made visible to users through an interactive interface. In this structure, system developers play a crucial role in interpreting the data, model, and results through their interpretative frameworks - a process referred to as *sense-reading* (Tsuchiya, 1993). The interpreted information is then conveyed to the end-users through the same lenses, in what is described as *sense-giving* (Tsuchiya, 1993). Users, in turn, perceive and make sense of this information, an act of *sense-reading* through their interpretative frameworks. This chain of information flow - from developers to users through the

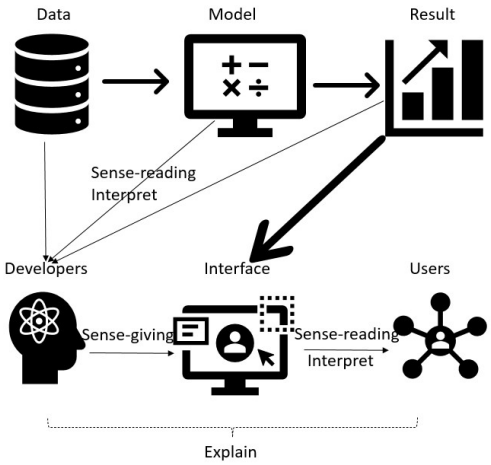


Figure 4.1: Interpret/Explain in AI systems

interface - forms the explanation within the AI system, as depicted in Figure 4.1. To summarize, within an AI system, developers interpret the information drawn from data, model, and results, and convey explanations to the users, justifying the generation of particular results (Doshi-Velez and Kim, 2017, Miller, 2019). These interactions within an AI system and its stakeholders highlight the crucial role of *explainability* and *interpretability* in AI, which are key for trust, transparency, and user satisfaction (Arrieta et al., 2020, Ribeiro et al., 2016). It also underscores the need for a robust and user-friendly interface that effectively bridges the gap between the AI system's internal operations and its external users (Lipton, 2018).

4.3.2 Definitions in the context of AI

Consequently, within the context of AI, we posit that an explanation is fundamentally user-oriented[7], addressing the question of why specific results have materialized. The process of interpretation encompasses the following facets: (1) interpretation of input data; (2) interpretation of the model; (3) post-hoc interpretation following model training; and (4) interpretation of the explanation itself. The interpretation of results is inextricably linked with the interpretation of data and models. By synthesizing the general definitions of *interpretability* and *explainability* in Definition 4.1 and Definition 4.2 respectively, we propose the subsequent definitions.

Definition 4.3 *Interpretability of data: the degree to which one (mainly a developer) can understand the information contained in data, which usually consists of data analysis and data visualization.*

In an AI system, as depicted in Figure 4.1, input data is considered interpretable, given that developers can extract insights from it using statistical analysis or data visualization techniques.

[7]In this thesis, our main emphasis is on explanations targeted at non-expert users.

However, this *interpretability* (Definition 4.3) may sometimes be constrained by the complexity and volume of the data, which could make extracting meaningful insights challenging. Techniques like dimensionality reduction or clustering can help make large and complex datasets more interpretable (Van der Maaten and Hinton, 2008).

Definition 4.4 *Explainability of model results: the ability to make model results understandable; the ability to give the reason or cause of model results.*

Definition 4.5 *Interpretability of model: the degree to which one (mainly a developer) can understand the mechanisms of model*

Definition 4.6 *Post-hoc interpretability: the degree to which one (mainly a developer) can explain a model without elucidating precisely how it works*

Definition 4.4 pertains to the ease with which an AI system's results can be explained. Readers may have noticed that we do not include post-hoc explainability, this is because explaining the results of AI systems can be realized either through the *interpretability* of the models themselves, as outlined in Definition 4.5, or via post-hoc explanation techniques, as described in Definition 4.6. In the first case, the explanations can be referred to as model-intrinsic explanations while in the second case, the explanations can be referred to as post-hoc explanations. Detailed exploration of model-intrinsic explanations and post-hoc explanations can be found in Sections 2.5 and 2.7.

Definition 4.7 *Interpretability of explanations: the degree to which one (mainly a user) can understand a given explanation.*

Definition 4.7 is concerned with the perception and understanding of explanations generated by an AI system, predominantly from the viewpoint of the users. This implies that Definition 4.7 is also an indicator of the quality of explanations produced by the AI system. The process and importance of evaluating explanations within the broader scope of AI is extensively discussed in Section 2.6. The targeted audience is explicitly stated in each of these definitions: Definitions 4.3, 4.4, 4.5, 4.6, and 4.7. These definitions serve to maximize clarity by ensuring that the intended audience understands the context and applicability of each definition. It further underscores the various levels of engagement and understanding that different stakeholders - from developers and system architects to end-users - have with AI systems.

Drawing upon the aforementioned definitions, it becomes evident that there is a need to distinguish between the terms "interpret" and "explain". Interpretation is a subjective process, whereas explanation inherently involves interactions. Consequently, an *Explainable AI System* can be characterized as an AI system capable of elucidating the rationale behind specific results it generates. In cases where the model deployed within this system possesses interpretability, the system may be classified as an *Interpretable AI system*. If the model within the system facilitates post-hoc interpretation, then the system can be referred to as a *Post-hoc Interpretable AI system*. The interrelationships between these concepts are visually depicted in Figure 4.2.

4.3.3 Discussions

In refining the definitions of *interpretability* and *explainability* in the realm of AI, the following three pivotal questions can be clarified: (1) who is the target audience for interpretation and explanation? (2) what motivates the need for *interpretability* and *explainability*? and (3) under what circumstances should we favor interpretable or explainable models?

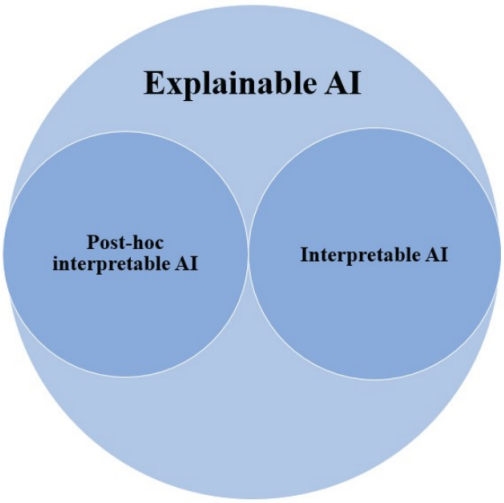


Figure 4.2: Explainable AI system, Interpretable AI system, Post-hoc interpretable AI system

- The first question relates to the target audience for interpretation and explanation. System developers, given their role in designing the system, are typically interested in every component of an AI system. They are expected to understand the function and significance of each part. Conversely, plain users are predominantly concerned with understanding why certain results are produced. The *interpretability* of data, which focuses on unraveling the information embedded in the data, caters mainly to developers. Similarly, the *interpretability* of models, addressing the internal workings of models, is primarily targeted at developers. When the internal mechanics of models are unclear, post-hoc *interpretability* becomes crucial, also aimed at developers. The *interpretability* of explanations is mainly directed at users, focusing on the quality of explanations. Therefore, the user's ability to interpret an explanation could reflect the quality of the explanation itself. *Explainability* of model results, chiefly targeting users, addresses the rationale behind specific results (Doshi-Velez and Kim, 2017, Miller, 2019).
- The second question examines the reasons behind the necessity for *interpretability* and *explainability*. As discussed in Section 2.4, humans require *interpretability* and *explainability* for a variety of reasons, including trust, fairness, and accountability among others (Arrieta et al., 2020), which can be grouped into commercial drives, regulatory drives, and technical drives (Zhong and Negre, 2021).
- The third question investigates the appropriate circumstances for employing interpretable or explainable models. In scenarios where the consequences of a decision made by the system are inconsequential, less transparent models such as deep learning techniques could be utilized to assure result quality, with explanations provided to users using post-hoc explanation techniques. This might apply to situations like movie recommendations. However, when the cost of a wrong decision is significant, interpretable models are often favored over black-box models to allow for more control and understanding of the decision-making process (Rudin, 2019, Carvalho et al., 2019).

The aforementioned considerations, which are essential to the wider field of AI, hold equally sig-

nificant relevance to the field of RSs. In the context of RSs, the *explainability* of recommendations aligns closely with Definition 4.4. This *explainability* can be realized either through the inherent *interpretability* of the RSs—anchored in Definition 4.5—or through the application of post-hoc explanation techniques, as delineated in Definition 4.6.

Moreover, the *interpretability* of explanations in RSs pertains to the users' ability to comprehend the generated recommendations effectively. Therefore, it serves as an insightful metric for the quality of explanations provided by these systems. For a more granular dissection of explanation evaluation within the field of RSs, the reader is directed to Section 3.7, where we delve deeper into the methods for evaluating explanations in RSs.

4.3.4 *General framework for generating explanations*

Building upon the IES shown in Figure 4.1 and the definitions provided in Section 4.3.2, we introduce a GFGE, as illustrated in Algorithm 2, which follows the IES detailed in Figure 4.1: developers interpret information from the data, model, and outcomes, subsequently translating these insights into explanations for users, which serve to explain the production of specific results (Doshi-Velez and Kim, 2017, Miller, 2019). What sets this framework apart is its inherent versatility. It does not prescribe specific methodologies for data preprocessing, the model under scrutiny, or the explanation approach. Instead, these components serve as parameters within Algorithm 2. They can be instantiated in various ways, affording the flexibility to adapt the explanation scenario for the outputs produced by any given model.

The input of Algorithm 2 includes $\mathcal{M}$, $\mathcal{P}$, $InterpretData$, $InterpretModel$, $InterpretOutput$, $PostHocInterpretation$ and $Aggregation$. $InterpretData$, $InterpretModel$ and $InterpretOutput$ correspond to the *sense-reading* process indicated in Figure 2.2 on Page 32 and $Aggregation$ relates to the *sense-giving* process.

- $\mathcal{M}$: the model in deployment
- $\mathcal{D}$: the input data whose outputs given by $\mathcal{M}$ is to be explained.
- $\mathcal{P}$: a function that does data pre-treatment, since the data at hand may not always satisfy the requirement of $\mathcal{M}$.
- $InterpretData$: a function that interprets information from $\mathcal{D}$, which corresponds to the *interpretability* of data in Definition 4.3.
- $InterpretModel$: a function that interprets information from $\mathcal{M}$, which corresponds to the *interpretability* of models in Definition 4.5.
- $InterpretOutput$: a function that interprets information from output of $\mathcal{M}$, which actually concerns how easily the output of $\mathcal{M}$ can be explained: explainability of model results in Definition 4.4.
- $PostHocInterpretation$: a function that interprets information from data and outputs without elucidating the internal mechanisms of $\mathcal{M}$. Typically, when $\mathcal{M}$ is not interpretable, it indicates that $InterpretModel(\mathcal{M})$ simply returns a prediction. For instance, the semantics of the latent factors of Vanilla Matrix Factorization (see Equation 3.12) remain unclear. Under such a situation, post-hoc interpretation is needed. For example, a surrogate

Algorithm 2: General Framework for Generating Explanations (GFGE)

Input : $\mathcal{M}$: a model in deployment

$\mathcal{D}$: a single data point or a set of data points whose outputs given by $\mathcal{M}$ is to be explained

$\mathcal{P}$: a function that does data pre-treatment

$InterpretData$: a function that interprets information from data

$InterpretModel$: a function that interprets information from model

$InterpretOutput$: a function that interprets information from the output of $\mathcal{M}$

$PostHocInterpretation$: a function that interprets information from $\mathcal{M}$ without elucidating its inner mechanism.

$Aggregation$: a function that aggregates information from $InterpretData$, $InterpretOutput$, $InterpretModel$ and $PostHocInterpretation$ to generate an explanation for the output.

Output: Explanations for the output returned by the model given the data

1 $\mathcal{D} \leftarrow \mathcal{P}(D)$

2 $output \leftarrow \mathcal{M}(\mathcal{D})$; ▷ output could be classification, a recommendation or a decision

3 $InfoData \leftarrow InterpretData(\mathcal{D})$

4 $InfoModel \leftarrow InterpretModel(\mathcal{M})$; ▷ When $\mathcal{M}$ is not interpretable, $InfoModel$ returns simply the output of $\mathcal{M}$, the internal mechanism remains unclear.

5 $InfoOutput \leftarrow InterpretOutput(output)$

6 $InfoPostHoc \leftarrow PostHocInterpretation(\mathcal{M})$

7 $Explanation \leftarrow Aggregation(InfoData, InfoModel, InfoOutput, InfoPostHoc)$

8 Return $Explanation$

model (e.g. LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017)). Therefore, $PostHocInterpretation$ corresponds to post-hoc interpretability in Definition 4.6.

- $Aggregation$: a function that aggregates information from $InterpretData$, $InterpretOutput$, $InterpretModel$ and $PostHocInterpretation$ to generate an explanation for the output.

In Section 2.6 of Chapter 2, we have surveyed existing explanation methods from the following aspects: model-intrinsic vs post-hoc; local vs global; model-specific vs model-agnostic. In the following, we will show how these discussions can be integrated into our framework depicted by Algorithm 2.

Model-intrinsic vs Post-hoc: the models used in an AI system can range from transparent to opaque. Transparent models, such as linear models, are easily interpretable because developers can readily understand the explicit meanings of the model parameters, such as the weights. This allows for a straightforward explanation of why a particular result or a group of results has been generated, and even how the overall system functions (Arrieta et al., 2020, Lipton, 2018). In this case, results can be explained by interpreting the model itself, see Figure 4.1 and line 4 in Algorithm 2. In this case, no post-hoc explanation method is needed. On the contrary, when dealing with opaque models like deep neural networks, the explicit meanings of parameters are not immediately apparent, making interpretation more challenging. In these cases, developers often turn to post-hoc explanation techniques to explain the system's outputs. These techniques essentially in-

volve establishing connections between input data features and the resulting outputs (Wang et al., 2018), constructing a simpler, interpretable model to mimic the behavior of the original, more complex model (Lundberg and Lee, 2017, Ribeiro et al., 2016), utilizing data visualization methods to reveal patterns and correlations (Van der Maaten and Hinton, 2008), or employing counterfactual reasoning to explain the decision-making process (Wachter et al., 2017). In essence, these post-hoc techniques are inherently linked to the interpretation of data, providing an added layer of transparency to model operations, see Figure 4.1 and line 6 in Algorithm 2.

Local vs Global: As discussed in Section 2.5.2, local and global explanations play distinct roles in interpreting the decision-making process of AI models, which is presented from line 3 in Algorithm 2. Local explanation techniques such as counterfactual explanations (Wachter et al., 2017), LIME (Ribeiro et al., 2016), and SHAP (Lundberg and Lee, 2017) concentrate on individual instances, thus illuminating the rationale behind particular AI determinations. In contrast, global explanation methods like DeepLIFT (Shrikumar et al., 2017) and permutation feature importance (Breiman, 2001) proffer a comprehensive understanding of the model's behaviors across all instances. In the context of Algorithm 2, if the input data comprises a single data point[8], the ensuing explanation is inherently local. However, if the data encompasses a group of data points, the generated explanations should encompass the general behavior across all these points. For transparent models, explanations can be furnished for both individual data points and the entire data set. However, opaque models, such as neural networks, pose challenges when it comes to explaining their behavior across all data points. Consequently, these models are typically explained by locally inspecting individual data points through techniques such as LIME (Ribeiro et al., 2016), and SHAP (Lundberg and Lee, 2017).

Model-specific vs Model-agnostic: model-specific techniques are tailored for a specific type or family of models. In this sense, model-intrinsic explanations are model-specific because they originated from interpretable models. As a matter of fact, this aspect mainly exists in post-hoc explanation scenarios (see line 6 in Algorithm 2). Examples of model-specific explanation techniques include TreeSHAP (Lundberg et al., 2018), visual explanation for convolutional network (Zeiler and Fergus, 2014), etc. In contrast, model-agnostic explanations are independent of the underlying model and can be applied to various types of models. These explanations focus on understanding the input-output relationships of the model without necessarily relying on the model's internal structure. Examples of model-agnostic explanation techniques include LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017). In the context of Algorithm 2, if the model is interpretable then explanations are model-specific. On the contrary, models applied to explain results returned by black-box models could be model-specific or model-agnostic. For more discussions, please refer to Section 2.5.2.

The framework depicted in Algorithm 2 is generic in the sense that it does not require a specific method for data preprocessing, the results to be explained, or the explanation method. Instead, they are considered as parameters of Algorithm 2, which can be instantiated in various ways to alter the way of explaining the results returned by a model. Besides, it seamlessly integrates the different facets of existing explanation methods in the context of AI. These facets range from model-intrinsic vs post-hoc explanations, to local vs global perspectives, and from model-specific vs model-agnostic

[8]The meaning of a data point varies across different domains. For instance, in the realm of RSs, a data point refers to a user-item interaction.

Table 4.1: Key notions and their target audience

Notions	Target audience
Interpretability of data	Mainly developers
Explainability of model results	Developers and plain users
Interpretability of model	Mainly developers
Post-hoc interpretability of model	Mainly developers
Interpretability of explanations	Mainly users

techniques.

4.4 Conclusions

In this chapter, we embarked on the task of constructing GFGE within the context of AI (Zhong and Negre, 2021). This work was motivated by the first research question presented in Chapter 1, which sought to unravel the differences between *interpretability* and *explainability* in AI. Our objective is to provide a clear understanding of the meanings of *explainability* and *interpretability* as well as to identify the target audience for these concepts. The importance of generality in this discussion was emphasized in section 4.1, where we highlighted the pervasive influence of AI technologies, their profound impact on various aspects of human life, including RSs, and the consequential demand for explainable AI. Recognizing this demand, we acknowledged the confusion in the AI research community surrounding the use of *interpretability* and *explainability*, which this chapter aimed to alleviate.

In response to this challenge, we proposed general definitions for these two terms in Section 4.2 and proceeded to develop the *Interpret/Explain Schema* (IES) for AI systems in Section 4.3. IES offers a logical structure to define *interpretability* and *explainability*, drawing from the concepts of interpretative frameworks (Jones et al., 2011) and mental models (Tsuchiya, 1993). In the ensuing discussions, we addressed three pivotal questions regarding the target audience for *interpretability* and *explainability* (See Table 4.1), the motivations for needing *interpretability* and *explainability*, and the circumstances under which we should favor model-intrinsic explanations or post-hoc explanations. We then propose GFGE in Algorithm 2, this framework seamlessly integrates the two explanation scenarios: through interpretable models (inherently explainable) or through post-hoc explanations. The aforementioned considerations, which are essential to the broader field of AI, hold equally significant relevance to the specialized field of RSs.

In conclusion, this chapter has offered a structured approach to understanding *interpretability* and *explainability* in AI, which is also applicable in the domain of RSs (since RSs is a branch of AI), providing clear definitions and discussing their implications. This sets the stage for the further exploration of these concepts in the subsequent chapters, where we aim to instantiate the two explanation scenarios.

Part III

Instantiations of the general framework for generating explanations

The *General Framework for Generating Explanations* (GFGE) we present in section 4.3.4 seamlessly integrates the two explanation scenarios: through interpretable models (model-intrinsic explanations) or through post-hoc explanations. In this part, we will instantiate GFGE under two explanation scenarios. We detail the first explanation scenario, emphasizing the design of interpretable models whose predictions can be explained inherently. To this end, we introduce the **A**ttribute[9]-**A**ware **A**rgumentative **R**ecommender (A^3R) (Zhong and Negre, 2022a). However, recognizing that certain arguments deemed "good" in a particular context might become less persuasive in a different context, we extend the idea of A^3R and propose **C**ontext-**A**ware **F**eature **A**ttribution **T**hrough **A**rgumentation (CA-FATA) (Zhong and Negre, 2023). CA-FATA leverages users' contexts, resulting in more accurate predictions like CARSs (Adomavicius et al., 2021), and ensuring that the explanations are adapted to users' contexts.

We then instantiate GFGE in the post-hoc scenario. The diversity of explanation methods for different RSs increases engineering costs, and the increasing complexity of RSs sometimes makes direct explanations unfeasible. This has led to the popularity of post-hoc explanation methods, such as SHAP (Lundberg and Lee, 2017), that generate explanations via simpler surrogate models. However, these methods present challenges, such as potentially lacking fidelity[10] to the original RSs (Rudin, 2019) and requiring an understanding of complex mathematical concepts, making them less user-friendly. To address these issues, we propose **S**hap-**E**nhanced **C**ounterfactual **E**xplanation (SECE) that generates counterfactual explanations using SHAP, drawing on relevant research in psychology, philosophy, and cognitive science.

This part is organized as follows: Chapter 5 provides the foundational background for the A^3R, CA-FATA, and SECE. The development of the A^3R model will be explored in Chapter 6, and the details of the CA-FATA model will be presented in Chapter 7. The instantiation of post-hoc explanation scenario will be introduced in Chapter 8. Chapter 9 compares the models proposed in Chapters 6, 7, 8, and presents the results of a primary user study where we evaluate different explanations methods in RSs.

[9]Note that the "attribute" here is the same as the "feature" in the following when we proposed A^3R we used the term "attribute". Then we realize that A^3R actually formulates the feature attribution in an argumentative way.

[10]The explanations generated may not be aligned with the mechanisms of RSs.

Chapter 5

Preliminary backgrounds

Contents

5.1 Introduction 82
5.2 Feature attribution 83
5.3 Argumentation 84
5.3.1 Argumentation frameworks (AFs) 84
5.3.2 Weak balance and weak monotonicity 86
5.4 LIME and SHAP 87
5.5 Relation to our work 89

Having identified the differences between interpretability and explainability, we arrive at the core research question of this thesis **"How to explain recommendations?"** defined in Chapter 1. More specifically, we first answer this question by addressing **"How to build transparent models that can explain recommendations while ensuring the accuracy of recommendations?"** We propose **A**ttribute **A**ware **A**rgumentative **R**ecommender (A^3R) (Zhong and Negre, 2022a) in Chapter 6 and **C**ontext-**A**ware **F**eature **A**ttribution **T**hrough **A**rgumentation (CA-FATA) (Zhong and Negre, 2023) in Chapter 7. A^3R and CA-FATA are two instantiations of the *General Framework for Generating Explanation* (GFGE) highlighted by Algorithm 2 in Section 4.3.4, focusing on model-intrinsic explanations. We will then answer the question from another perspective: **"How to explain existing complex RSs using post-hoc explanation techniques while mitigating the limitations of such techniques?"** To this end, we propose generating counterfactual explanations guided by SHAP: **S**HAP-**E**nhanced **C**ounterfactual **E**xplanations (SECE) (Zhong and Negre, 2022c). SECE is also an instantiation of the *general framework for generating explanation* depicted in Algorithm 2 in Section 4.3.4, focusing on post-hoc explanations.

This chapter lays out the technical background for developing A^3R in Chapter 6, CA-FATA in Chapter 7 and SECE in Chapter 8. Central to the three approaches is the concept of explainable recommendations framed as a feature attribution problem, which entails identifying the contribution of individual features or variables to a model's overall output. A^3R leverages on the power of argumentation by viewing each feature as an argument that can either support, attack or remain neutral to a prediction. More precisely, A^3R defines feature attribution as an argumentation process in which each computation has explicit semantics, making it inherently interpretable. However, acknowledging that the persuasiveness of certain arguments labeled as "good' may decrease in

a different context, we propose an extension of A^3R: CA-FATA. SECE generates counterfactual explanations guided by SHAP (Lundberg and Lee, 2017), it identifies the minimal changes to features such that the output of the model in deployment is different. In this sense, SECE also applies the notion of feature attribution, while the important features returned by SHAP are validated by counterfactual reasoning.

5.1 Introduction

The models we present in Chapters 6, 7 and 8 identify the contribution of individual features or variables to a model's output, transforming explainable recommendations into feature attribution problems. The process of feature attribution, a long-standing practice in machine learning, demystifies model behaviors by pinpointing influential features for an outcome and potential areas for model enhancement (Afchar and Hennequin, 2020, Zhong and Negre, 2022a, Rago et al., 2018, Nóbrega and Marinho, 2019, Lundberg and Lee, 2017, Ribeiro et al., 2016, Zhong and Negre, 2022c). The lineage of feature attribution methods traces back to General Additive Models (GAMs) (Hastie, 2017). Despite GAMs' inherent interpretability, they are often criticized for their limited expressivity (Molnar, 2020). To understand complex Artificial Intelligence (AI) systems, some recent works employ gradient-based methods (Li et al., 2021, Selvaraju et al., 2017, Ancona et al., 2019). These approaches quantify a feature's significance in a function by calculating the derivative of the function with respect to that feature. However, such methods may fail on tasks necessitating a moderately local understanding (Bilodeau et al., 2022). Furthermore, the interpretation of such gradients can prove daunting for non-experts. Surrogate models like LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017) emerged to counter gradient-based methods' limitations. Despite their prominence, they too present drawbacks. LIME inherently grapples with stability issues (Visani et al., 2022). As for SHAP (Lundberg and Lee, 2017), the attribution of feature importance via formally definable properties - such as local accuracy, missingness, and consistency - may misalign with users' explanation expectations (Kumar et al., 2020). More existing works on feature attribution will be reviewed in Section 5.2.

In light of the challenges associated with existing feature attribution methods, it is reasonable to explore new avenues for improving the explainability of recommendations. Since argumentation inherently offers interpretability, one such approach is to leverage argumentation techniques to attribute feature importance, which addresses **RQ3.1**: **"How to build transparent models that can explain recommendations while ensuring the accuracy of recommendations?"**. This part introduces two such argumentation-based models: A^3R, which operates without context, and CA-FATA, which builds upon A^3R by incorporating user contexts. Both A^3R and CA-FATA attribute importance to item features by treating them as arguments that either support, attack, or neutralize a prediction. This process takes place under Argumentation Frameworks (AFs), offering explicit semantics to each computation and thus ensuring the interpretability of the model. Experiments carried out on four datasets from two distinct domains—movies and books—demonstrate that A^3R offers competitive rating prediction performance compared to existing methods and significantly improves prediction accuracy over current state-of-the-art argumentative recommendation methods. Additionally, CA-FATA permits the integration of user contexts, leading to more accurate predictions. Experiments on two real-world datasets show that CA-FATA outperforms existing argumentation-based methods, matching the performance of existing context-free and context-aware methods. Furthermore, CA-FATA ensures the explainability of its recommen-

dations, marking it as a promising framework for practical applications.

It would also be reasonable to build on existing feature attribution methods and avoid the limits they represent. Among the existing feature attribution methods, we are interested in surrogate models. To avoid their limits, we apply counterfactual reasoning to validate the important features returned by the surrogate models, which addresses **RQ3.2: "How to explain existing complex RSs using post-hoc explanation techniques while mitigating the limitations of such techniques?"** More concretely, we propose generating counterfactual explanations guided by SHAP (Lundberg and Lee, 2017). The main idea is to apply SHAP first to identify features that are potentially important. These important features serve as intelligent starting for searching counterfactual explanations.

The remainder of this chapter is structured as follows: Section 5.2 introduces feature attribution methods; Section 5.3 presents the essential concepts pertaining to argumentation, the two desired properties of argument strength function (*weak balance* (Rago et al., 2018) and *weak monotonicity* (Rago et al., 2021)) that will be further explored in Chapters 6 and 7. Section 5.4 discusses more details on the two most prominent surrogate models: LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017). Finally, Section 5.5 introduces how these concepts relate to our work.

5.2 Feature attribution

There are numerous methods available to identify the most important features in the field of machine learning. Chronologically, feature attribution can be traced back to GAMs (Hastie, 2017). Recently, gradient-based methods (Li et al., 2021, Selvaraju et al., 2017, Smilkov et al., 2017) have gained popularity due to the widespread use of deep neural networks. Developing surrogate models (Lundberg and Lee, 2017, Ribeiro et al., 2016) to approximate black-box models by human-interpretable models is another approach that has been widely adopted.

GAMs extend linear regression models by incorporating non-linear relationships between dependent and independent variables and are typically in the following form:

$$g(y) = f_1(x_1) + f_2(x_2) + f_2(x_2) + \cdots + f_i(x_i) + \cdots + f_n(x_n) \tag{5.1}$$

where $x_1, x_2, \ldots, x_i, \ldots, x_n$ are the features applied to make predictions, x_i is the ith feature and $i \in [1, n]$, n is the number of features utilized to make predictions; y is the target value to be predicted, g is called the link function and f_i is the shape function (Lou et al., 2012). GAMs have a built-in interpretability feature that allows tracing the contribution of each feature, thereby facilitating the identification of the most important features. In this sense, GAMs can, to some extent, generate model-intrinsic explanations. Nevertheless, the expressivity of GAMs is limited compared with more complex models such as ensemble models (Molnar, 2020).

Gradient-based methods have become a popular means of identifying important features in predictive functions $f(x_1, x_2 \ldots x_i \ldots x_n)$ by computing the derivative of f with respect to the feature x_i: $\frac{\partial f}{\partial x_i}$ (Li et al., 2021, Selvaraju et al., 2017, Ancona et al., 2019). This is based on the intuition that the sensitivity of the model output to changes in the input, as quantified by $\frac{\partial f}{\partial x_i}$, reveals the importance of the feature. However, as noted by Ancona et al. (2019), gradient-based methods can be sensitive to noisy gradients and may struggle with simple tasks that require

understanding a moderately local region (Bilodeau et al., 2022). Additionally, interpreting these gradients can be challenging for non-experts.

Surrogate models aim to approximate the behavior of black-box models with more interpretable ones, without revealing the internal workings of black-box models. Among the many surrogate models available, LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017) have emerged as two prominent methods. LIME and SHAP can both be regarded as additive feature attribution methods and can be unified as shown in Equation 5.1 (Lundberg and Lee, 2017). However, since LIME and SHAP are post-hoc methods, it has been argued that the explanations may not accurately reflect the reasoning of black-box models, and may even be misleading (Rudin, 2019). Additionally, studies have demonstrated that LIME and SHAP can be easily deceived into producing explanations that do not reveal underlying biases (Slack et al., 2020).

In view of the limits of existing feature attribution methods, we first define a novel feature attribution framework that leverages argumentation. Specifically, the framework maps the prediction process into argumentation procedures, treating features as arguments and incorporating a strength function that reflects how each argument (feature) influences the prediction. Another approach we propose is to avoid the limits of existing post-hoc explanation methods through counterfactual reasoning.

5.3 Argumentation

In this section, we will introduce the basic notions related to argumentation, the *weak balance* (Rago et al., 2018) and *weak monotonicity* (Rago et al., 2021) that will be applied in Chapters 6 and 7.

5.3.1 Argumentation frameworks (AFs)

To standardize the terminology used in the rest of this part, we present the key notions in Table 5.1. Among the existing AFs, three types can be identified: Abstract Argumentation Framework (AAF) (Dung, 1995), Bipolar Argumentation Framework (BAF) (Cayrol and Lagasquie-Schiex, 2005), Tripolar Argumentation Framework (TAF) (Gabbay, 2016). An AAF is composed of a set of pairs $< \mathcal{A}, \mathcal{R}^- >$, where $\mathcal{R}^-$ denotes a set of attack relations between arguments such that $\forall \mathbf{a}_1, \mathbf{a}_2 \in \mathcal{A}$, $(\mathbf{a}_1, \mathbf{a}_2) \in \mathcal{R}^-$ denotes that argument $\mathbf{a}_1$ attacks argument $\mathbf{a}_2$. The relation "attacks" indicates a contradiction between two arguments. For example, considering $\mathbf{a}_1$ "This user does not like the feature of this item (one actor of a movie)" and $\mathbf{a}_2$ "This item can be recommended to this user". It is evident that $\mathbf{a}_1$ attacks $\mathbf{a}_2$. BAFs contain a set of triplets, $< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+ >$, $\mathcal{R}^-$ represents attack as in AAF. Similarly, $\mathcal{R}^+$ denotes a set of support relations between arguments such that $\forall \mathbf{a}_1, \mathbf{a}_2 \in \mathcal{A}$, $(\mathbf{a}_1, \mathbf{a}_2) \in \mathcal{R}^+$ denotes that argument $\mathbf{a}_1$ supports argument $\mathbf{a}_2$. TAFs contain a set of quadruples: $< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+, \mathcal{R}^0 >$, where $\mathcal{R}^-$ represents the attack relations, $\mathcal{R}^+$ denotes the support relations and $\mathcal{R}^0$ means neutralizing relations. In this work, we have chosen to adopt TAFs that comprise three types of relations between arguments: attack, support, and neutralizing. This is because features of items may support, attack, or neutralize the recommendation of items, indicating users' preferences towards features. As we will discuss in Chapters 6 and 7, the strength of the arguments in the TAFs presented in this chapter is based on the users' ratings towards features. Consequently, the TAFs in this chapter can be viewed as instances of Weighted Argumentation Frameworks as defined by Bistarelli and Taticchi (2022) where the weight denotes the strength of arguments. Argumentation-based methods have gained attention for building decision-support

Table 5.1: The key notions in Chapter 5

Notion	Description
$\mathcal{R}^+, \mathcal{R}^-, \mathcal{R}^0$	Support, attack and neutral relations among arguments
$\mathcal{R}^-$(a,b)	Argument a attacks argument b, the same for "+" and "0"
$\mathcal{A}$	A set of arguments
rec^i	An argument stating "the item can be recommended to the target user"
at_i	The features of item i
$\mathcal{R}^+(a) = \{b \vert (b,a) \in \mathcal{R}^+\}$	The arguments (features) that support a
$\mathcal{R}^-(a) = \{b \vert (b,a) \in \mathcal{R}^+\}$	The arguments (features) that attack a
$\mathcal{R}^0(a) = \{b \vert (b,a) \in \mathcal{R}^+\}$	The arguments (features) that neutralize a
$< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+, \mathcal{R}^0 >$	A tripolar argumentation framework
$< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+ >$	A bipolar argumentation framework
$< \mathcal{A}, \mathcal{R}^- >$	An abstract argumentation framework
$\sigma(a)$	The strength of argument a
λ	A black-box model to be explained
l	An interpretable linear model
$\mathcal{L}$	A set of simpler linear models g
$x = \{x_1, x_2, x_3 \dots x_p\}$	An instance to be explained
p	The number of features used in λ
$\delta_x(x^{'})$	Proximity between $x^{'}$ and instance x
$\Omega(l)$	The complexity of l
$X^{'}$	A set of instance x's neighbors
ϕ_j	Margin contribution of feature x_j

tools due to their natural explainability. In the field of RSs, such methods have received increasing attention. For example, Briguez et al. (2012, 2014) proposed a music RS based on Defeasible Logic Programming (DeLP) rules (García and Simari, 2004), which can handle incomplete and contradictory information in RS. Toniolo et al. (2020) proposed a decision-making system that provides medical treatments to patients with multiple chronic health problems. To justify the recommended treatments during argumentation dialogues, they employed *Satisfiability Modulo Theories* (Barrett and Tinelli, 2018).

In recent years, the prominence of argumentation-based methods in eXplainable Artificial Intelligence (XAI) has grown significantly (Vassiliades et al., 2021, Čyras et al., 2021, Zeng et al., 2018). This is primarily attributable to the clarity of relationships presented by AFs, such as support and attack. AFs deliver explicit semantics to computations, allowing for a visual depiction of decision-making processes and an explanation of optimal decisions via well-defined properties (Vassiliades et al., 2021). Weighted arguments represent the strength and dialectical relationships between arguments, encapsulating elements like support and attack. The strength function of arguments can be carefully constructed to satisfy generalized notions of *weak balance* (Rago et al., 2018) and *weak monotonicity* (Rago et al., 2021). These notions characterize how arguments influence the decision-making process (we will further discuss these in Section 5.3.2). These methods offer graphical representations of the decision-making process, aiding in explaining decisions. The A^3R (Zhong and Negre, 2022a) we propose is closely related to a recently proposed argumentation

recommender: the *Aspect-Item framework (A-I)* introduced by Rago et al. (2018, 2021). Both A-I and A^3R use argumentation to predict users' ratings towards items, treating items and features as arguments that may attack or support each other to explain recommendations in an argumentative manner. A^3R can largely improve the performance of recommendations in terms of prediction accuracy. Context-Aware Recommender Systems (CARSs) (Adomavicius et al., 2021) (more details have been presented in Section 3.6), an essential research area in recommender systems (RSs), can model users' preferences in various contexts with increased granularity, thereby generating personalized recommendations that are contextually apt for users. We argue that context is equally crucial in AFs since the effectiveness of certain arguments may fluctuate across different contexts. Therefore, incorporating context when employing argumentation is a critical consideration (Teze et al., 2018, García and Simari, 2004). CA-FATA extends A^3R by considering users' contexts to generate context-aware recommendations and explanations.

Concretely, to study how arguments influence each other, the strength function of arguments can be carefully designed to satisfy desired properties such as *weak balance* (Rago et al., 2018) and *weak monotonicity* (Rago et al., 2021). In the next subsection, we will detail the two properties.

5.3.2 Weak balance and weak monotonicity

The concept of *weak balance*, as defined by Rago et al. (2018), is a generalization of the notion of "strict balance" proposed by Baroni et al. (2018). Similarly, the idea of *weak monotonicity*, defined by Rago et al. (2021), is a generalization of the concept of "strict monotonicity" as defined by Baroni et al. (2019). The two properties allow for deriving intuitive explanations in an argumentative way. At their core, *weak balance* and *weak monotonicity* utilize counterfactual reasoning to analyze the influence of arguments. Specifically, the concept of *weak balance* concerns the impact of an argument on its affectees when the argument is the sole factor affecting them, while the idea of *weak monotonicity* focuses on how the potency of an argument changes when one of its affecters is silenced relative to the neutral point. In the following, we will delve into these concepts, providing comprehensive definitions and detailing their implications.

Weak balance: One potential approach to analyze the effects of one argument on another is to isolate the affecter and examine its impact on the affectee. The intuition behind this approach is that if the affecter increases the strength of the affectee, then it supports the affectee. This idea has been formalized as *weak balance* by Rago et al. (2018). According to *weak balance*, relations under AFs such as attacks (or supports, neutralizes) can be characterized as connections among affecters and affectees in the following way: if one affecter is isolated as the single argument that affects the affectee, then the former reduces (or increases, does not change) the latter's predicted rating with respect to the neutral point. In other words, relations under AFs can be analyzed by examining the connections between affecters and affectees. Formally, *weak balance* can be defined as follows:

Definition 5.1 *Given a TAF* $< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+, \mathcal{R}^0 >$, $\sigma(at)$ *satisfies the property of* weak balance *if, for any* $x, y \in \mathcal{A}$:

- *if* $\mathcal{R}^+(x) = \{y\}$, $\mathcal{R}^-(x) = \emptyset$ *and* $\mathcal{R}^0(x) = \emptyset$ *then* $\sigma(x) > 0$;
- *if* $\mathcal{R}^-(x) = \{y\}$, $\mathcal{R}^+(x) = \emptyset$ *and* $\mathcal{R}^0(x) = \emptyset$ *then* $\sigma(x) < 0$;
- *if* $\mathcal{R}^0(x) = \{y\}$, $\mathcal{R}^+(x) = \emptyset$ *and* $\mathcal{R}^-(x) = \emptyset$ *then* $\sigma(x) = 0$.

Weak monotonicity: To study the effects of one argument on another, another alternative approach is to mute the affecter, setting its strength to 0, and observe how the strength of the affectee changes. This idea is intuitive: if the affecter supports the affectee, then muting the affecter would decrease the strength of the affectee; if the affecter attacks the affectee, then muting the affecter would increase the strength of the affectee; if the affecter neutralizes the affectee, then muting the affecter would not change the strength of the affectee. This intuition has been formalized as *weak monotonicity* by Baroni et al. (2019). This property is formulated for two TAFs: from $< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+, \mathcal{R}^0 >$ to $< \mathcal{A}', \mathcal{R}^{-'}, \mathcal{R}^{+'}, \mathcal{R}^{0'} >$ after modifying certain arguments (e.g. muting certain features). Formally, *weak monotonicity* is defined as follows:

Definition 5.2 *Given two TAFs* $< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+, \mathcal{R}^0 >$ *and* $< \mathcal{A}', \mathcal{R}^{-'}, \mathcal{R}^{+'}, \mathcal{R}^{0'} >$, $(x, y) \in (\mathcal{R}^- \cup \mathcal{R}^+ \cup \mathcal{R}^0) \cap (\mathcal{R}^{-'} \cup \mathcal{R}^{+'} \cup \mathcal{R}^{0'})$. σ *satisfies* weak monotonicity *at* (x, y) *if, as long as* $\sigma(x) = 0$ *in* $< \mathcal{A}', \mathcal{R}^{-'}, \mathcal{R}^{+'}, \mathcal{R}^{0'} >$ *and* $\forall z \in [(\mathcal{R}^-(y) \cup \mathcal{R}^+(y) \cup \mathcal{R}^0)(y) \cap (\mathcal{R}^{-'}(y) \cup \mathcal{R}^{+'}(y) \cup \mathcal{R}^{0'})(y)] \backslash \{x\}$ *and for* $\sigma(z) = s$ *in* $< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+, \mathcal{R}^0 >$ *and* $\sigma(z) = s'$ *in* $< \mathcal{A}', \mathcal{R}^{-'}, \mathcal{R}^{+'}, \mathcal{R}^{0'} >$, $s = s'$. *Then the following holds for* $\sigma(y) = v$ *in* $< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+, \mathcal{R}^0 >$ *and* $\sigma(y) = v'$ *in* $< \mathcal{A}', \mathcal{R}^{-'}, \mathcal{R}^{+'}, \mathcal{R}^{0'} >$

- *if* $x \in \mathcal{R}^-(y) \cap \mathcal{R}^{-'}(y)$, *then* $v' > v$;
- *if* $x \in \mathcal{R}^+(y) \cap \mathcal{R}^{+'}(y)$, *then* $v' < v$;
- *if* $x \in \mathcal{R}^0(y) \cap \mathcal{R}^{0'}(y)$, *then* $v' = v$.

To summarize, *weak balance* (Rago et al., 2018) and *weak monotonicity* (Rago et al., 2021) characterize how arguments influence the decision-making process, which will be applied to explain recommendations in Chapters 6 and 7.

5.4 LIME and SHAP

As we presented in Section 2.5, to understand the increasingly inscrutable AI models, one popular way is to train a simpler model to approximate the original complex model. LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017) are two state-of-art model-agnostic methods that adopt this strategy. In this section, we compare the two methods and show why we have chosen SHAP.

LIME (Ribeiro et al., 2016) belongs to local interpretation methods. It is model-agnostic, which means that LIME can be applied to any model. It explains the result of a single instance x by approximating the original black-box model λ with a linear model l. The problem can be solved by the following optimization problem:

$$\arg\min_{l \in \mathcal{L}} L(\lambda, l, \delta_x) + \Omega(l) \tag{5.2}$$

where $\Omega(l)$ is the number of coefficients in l that is not zero and the loss function:

$$L(\lambda, l, \delta_x) = \sum_{x' \in X'} (\lambda(x') - l(x'))^2 \delta_x(x') \tag{5.3}$$

where $\delta_x(x')$ is usually cosine similarity or $l2$ distance (Ribeiro et al., 2016).

Figure 5.1 is a demo of LIME, showing the general idea of LIME. The blue/pink is the original black-box model λ. The bold red cross is the instance that we are interested in (to be explained). The dashed line is the human-understandable model l to explain the instance x. The main steps include:

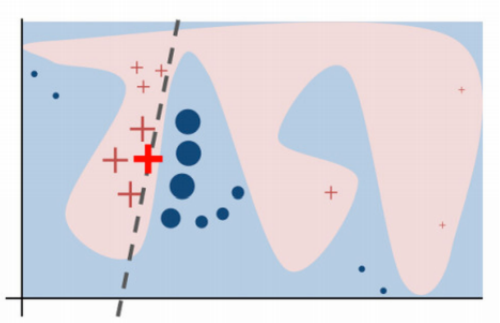


Figure 5.1: A demo of LIME (Ribeiro et al., 2016)

- (1) Generate new data points, according to the distribution of p features in the original dataset. The features are assumed to: follow a normal distribution; be independent of each other. Then get the predictions of new data points using the black-box model λ.
- (2) Assign weights to each new data point using a Gaussian Kernel (Ribeiro et al., 2016). The weights are decided by the distance of each new data point to the instance x to be explained. Like in Figure 5.1, the weights are represented by their size, the larger the size the larger the weight.
- (3) A linear model (e.g. Ridge Regression (Ribeiro et al., 2016)) l which uses a limited number of features is built for these newly created data points including the red cross. The linear model l is used to approximate the black-box model λ to explain the behavior of complex model λ around instance x, it is locally faithful, however, it is not globally faithful.

SHAP (Lundberg and Lee, 2017) basically originated from shapley value (Hart, 1989) in game theory. This method can provide global and local interpretations based on game theory. It considers each feature as a "player", and the "game" itself is the task of prediction. Therefore, the problem becomes calculating the contribution of each "player". SHAP calculates the average of the marginal contributions across all permutations for each feature.

Lundberg and Lee (2017) proved that Equation 5.2 can be solved by properly selecting $\Omega(l)$, $\delta_x(x^{'})$ and $L(\lambda, l, \delta_x)$ to guarantee desired properties such as local accuracy and consistency. After calculating the contribution of each feature for each instance, the global contribution can be calculated as the average of each feature over all instances, offering a global insight into each feature (Lundberg and Lee, 2017). Besides, the values of $\lambda(x)$ can be written as Equation 5.4, where ϕ_0 is the average prediction of the complex model λ and $\phi_1, \phi_2, \phi_3, \ldots$, can be calculated using the SHAP package implemented in Python (Lundberg and Lee, 2017). For more details, we refer to Lundberg and Lee (2017).

$$\lambda(x) = \phi_0 + \phi_1 + \phi_2 + \cdots + \phi_p \tag{5.4}$$

Comparison With the introduction above, we now compare LIME and SHAP. Table 2 summarizes the similarities and differences between the two methods.

The similarities between LIME and SHAP: (1) LIME and SHAP can both generate post-hoc explanations for a single instance to be explained without the need to elucidate the internal mechanism of models; (2) LIME and SHAP can be used to interpret any models. Although SHAP has

Table 5.2: Comparison between LIME and SHAP

	Local or Global	Stability concerns	Visualization
SHAP	Local and Global	No	Yes
LIME	Local	Yes	Yes

different explainers (Lundberg and Lee, 2017) such as TreeExplainers and DeepExplainers. Basically, the KernelShap can be applied to interpret any model; (3) Both LIME and SHAP provide intuitive visualization which shows the contribution of each factor, helping developers understand how complex models work. It should be noted that for plain users, it is non-trivial to understand the outputs returned by LIME and SHAP since background mathematical knowledge is required.

The differences between LIME and SHAP: (1) LIME can only provide local interpretations, meaning that it can only explain a single instance at a time. However, SHAP can not only zoom in on a single instance but also interpret the complex model from a global point of view; (2) Every time LIME is applied, it selects a limited number of features and generates new data points, therefore, the linear models (e.g. Ridge Regression) generated can be different (aka. the features used and the coefficients returned), raising concerns about the stability of LIME (Visani et al., 2020, Zafar and Khan, 2021). SHAP is basically grounded on game theory, which allows a fair computation of the contribution of each feature. Besides, desired properties such as local accuracy and consistency (Lundberg and Lee, 2017) are guaranteed. Table 5.2 summarizes the above comparison between LIME and SHAP. To summarize, the advantages of SHAP compared to LIME include (1) the global interpretation it provides, which allows developers to understand the behaviors of models from a global point of view; (2) since desired properties such as consistency are satisfied, there is no concern about stability issues. Therefore, we have chosen SHAP to generate post-hoc explanations.

5.5 Relation to our work

In this chapter, we have established the foundational knowledge required to develop A^3R (discussed in Chapter 6), CA-FATA (covered in Chapter 7), and SECE (elaborated in Chapter 8). The core of A^3R, CA-FATA, and SECE is the idea of explainable recommendations, conceptualized as a feature attribution problem. This involves discerning the influence of individual features or variables on the total output of a model. Given the difficulties surrounding conventional feature attribution methods, our first solution is to harness the potential of AFs, wherein item features are perceived as arguments. Specifically, A^3R (Zhong and Negre, 2022a) serves as a context-free instantiation of the *general framework for generating explanations* that we outline in Section 4.3.4. CA-FATA (Zhong and Negre, 2023) is a context-aware extension of A^3R that considers users' contexts. Our second solution is to expand on existing feature attribution methods that build surrogate models and validate the important features by counterfactual reasoning (Zhong and Negre, 2022c). This idea will be developed in Chapter 8.

A^3R: The primary task of RSs is to recommend relevant items to users u. To realize this goal, we make the following intuitive claims: (1) It is reasonable to assert that the features of an item, such as the actors or directors of movies, can influence users' preferences towards items. The influences can be positive, negative, or neutral; (2) In this regard, the features (at_i) can be viewed as arguments, along with another argument stating "the item can be recommended to the target user", rec^i; (3) Such a scenario can be seamlessly integrated into a TAF (Gabbay, 2016). For a

user-item interaction (u, i), the TAF tailored to this interaction is a quadruple: $< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+, \mathcal{R}^0 >$, where $\mathcal{A}$ contains at_i and rec^i. Intuitively, $\mathcal{R}^+(at, rec^i)$ denotes that feature at has a positive effect on the recommendation of item i to user u; $\mathcal{R}^-(at, rec^i)$ denotes that feature at has a negative effect on the recommendation of item i to user u; $\mathcal{R}^0(at, rec^i)$ denotes that feature at does not influence the recommendation of item i to user u. Hence, the objectives are as follows and will be further detailed in Chapter 6

- Predicting the rating assigned by a target user u to a particular item i.
- Determining the contribution of each item feature to a prediction.
- Assessing the polarity of each argument (feature) in the TAF.
- Designing the strength function of arguments that comply with the conditions of *weak balance* and *weak monotonicity* as defined in Section 5.3.2.

CA-FATA: A^3R does not consider users' contexts, which can influence users' preferences (Adomavicius et al., 2021). Considering the following recommendation scenario: for a target user u under a contextual situation cs. Like the context-free case, we make the three claims. As a result, for a user-item interaction (u, i) under cs, the TAF tailored to this interaction is a quadruple: $< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+, \mathcal{R}^0 >$, where $\mathcal{A}$ contains at_i and rec^i. The only difference with the context-free case is that the polarity of arguments is adapted to users' contexts. Hence, the objectives are as follows and will be further discussed in Chapter 7:

- Predicting the rating assigned by a target user u to a particular item i in a given contextual situation cs.
- Determining the contribution of each item feature to a prediction under this contextual situation cs.
- Assessing the polarity of each argument (feature) in the TAF.
- Designing the strength function of arguments that comply with the conditions of *weak balance* and *weak monotonicity* as defined in Section 5.3.2.

SECE: direct application of SHAP raises substantial concerns and limitations: (1) explanations returned by SHAP may not accurately represent the original RSs, and in some cases, these explanations can even be incorrect (Rudin, 2019); (2) for lay users, comprehending the significance of features as determined by these methods is challenging, given the prerequisite of intricate mathematical knowledge. For instance, the output of SHAP includes a "force_plot" figure (used to visualize how a prediction is achieved by showing the contribution of each feature please refer to Lundberg and Lee (2017)), necessitating a fundamental understanding of game theory on the part of the users. While SHAP serves as a useful tool for model developers, enabling them to visualize model behaviors and potentially aiding in model debugging, such graphical representations may prove difficult for lay users to interpret. Our solution is to validate the important features returned by SHAP through counterfactual reasoning. Under this setting, we apply existing recommender models, which are treated as black boxes. Therefore, the accuracy of prediction is out of our scope. That said, the objectives are as follows and will be further developed in Chapter 8:

- Identify the features that are potentially important to a recommender model
- Validate features through counterfactual reasoning.

Chapter 6

Model-intrinsic instantiation: Attribute-aware argumentative recommender

Contents

6.1 Introduction . . . 92
6.2 Recommendations through argumentation . . . 93
6.3 Our model . . . 94
6.3.1 Problem setting . . . 94
6.3.2 Main steps of A^3R . . . 96
6.4 Explaining recommendations using A^3R . . . 98
6.4.1 Argumentation scaffolding . . . 98
6.4.2 Explanation applications . . . 100
6.4.3 Relations with other models . . . 104
6.5 Generating recommendations using A^3R . . . 105
6.5.1 Experiments setting . . . 106
6.5.2 Experiment results . . . 107
6.5.3 Abalation study . . . 109
6.5.4 Case study . . . 110
6.6 Conclusions . . . 112

This chapter provides an instantiation of the *General Framework for Generating Explanation* (GFGE) highlighted by Algorithm 2 in Section 4.3.4 on Page 73, focusing on model-intrinsic explanations. We introduce the **A**ttribute-**A**ware **A**rgumentative **R**ecommender (A^3R) (Zhong and Negre, 2022a), which reframes explainable recommendations as a feature attribution problem. Essentially, this means discerning the impact of each feature on a given prediction. Specifically, A^3R conceives feature attribution as an argumentation process, treating features as arguments that can support, attack, or remain neutral to a prediction. Experimental results on four datasets spanning two distinct domains—movies and books—validate A^3R's ability to generate accurate predictions. The structure of argumentation underscores A^3R's capacity to explain recommendations in an argumentative manner. Simultaneously, we are aware that users' context can significantly influence

their preferences (Adomavicius et al., 2021), and that certain arguments deemed compelling in one context might lose their persuasive power in another. This chapter begins with an implementation that does not take into account users' context. A context-aware application will be presented in the next chapter.

The publication related to this chapter is:

- Zhong, J. and Negre, E. (2022). A^3R: Argumentative explanations for recommendations. In 2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA), pages 1–9. IEEE.

6.1 Introduction

Latent factor models have gained popularity since the Netflix challenge (2007) (Bennett et al., 2007). From a chronological point of view, after the Netflix challenge, various factorization-based models have been proposed. Matrix factorization (MF) (Koren et al., 2009) models users' preferences by simply adopting the dot product of user embedding and item embedding. More precisely, each user is represented by a vector $\mathbf{p}_u$, each user is represented by a vector ${\mathbf{q}_i}^T$. u' rating toward i is predicted as $\hat{r}_{(u,i)} = \mathbf{p}_u {\mathbf{q}_i}^T$, see Equation 3.12 in Section 3.4.2.1. Factorization machine (FM) (Rendle, 2010) extends this idea by capturing the second-order interactions of other features such as features of items (e.g. actors of a movie). Neural Factorization Machine (NFM) (He and Chua, 2017) proposes a *Bilinear Interaction* pooling operation to increase the expressivity of FM. Attentional Factorization Machines (AFM) (Xiao et al., 2017) further add an attention mechanism to model the importance of feature interactions. Neural Matrix Factorization (NeuMF) (He et al., 2017) proposes generalizing MF by multi-layer perceptron (MLP). These factorization-based methods have been proven to be effective in rating prediction and ranking prediction. NeuMF (He et al., 2017) has become a strong baseline in many papers (Mattson et al., 2020, Zamani and Croft, 2020, Al Jawarneh et al., 2020, Qin et al., 2020, Hu et al., 2018, Zhao et al., 2020). Nevertheless, as we introduce in Section 3.4.2, the explicit meanings of the learned latent factor remain unclear and the use of MLP makes factorization-based even more opaque, which makes it difficult to know users' preference toward the features of an item. Explaining recommendations generated by such techniques is non-trivial. Finding "relevant-user" is the common strategy applied to explain recommendations generated by factorization-based recommender techniques. Indeed, users usually have little knowledge (e.g. the items other users have interacted with before) about "relevant-user", which could hinder the trustworthiness of explanations (Zhang et al., 2020). At the same time, using other users' interactions may lead to privacy concerns.

In another line of research, argumentation frameworks (AFs) have been applied to enhance the explainability of results generated by AI systems (Vassiliades et al., 2021, Čyras et al., 2021). AF can represent the decision-making process in a graphical way, the predefined properties help to reason how to reach the best decisions (Vassiliades et al., 2021). AF also includes ways to define weighted arguments and dialectical relations between arguments, which contain semantics to decide how strong and acceptable these arguments are (Čyras et al., 2021). Argumentation-based methods have been applied in laws (Karafili et al., 2020), medical treatment (Sassoon et al., 2019) and cyber-security (Bench-Capon, 2020). In the domains of RSs, there have been several works (Teze et al., 2015, Rago et al., 2018, 2021, 2020) that build explainable RSs through AFs.

In our work (Zhong and Negre, 2022a), we aim to build a latent factor model that can provide model-intrinsic explanations. To this end, A^3R, which reframes explainable recommendations as

a feature attribution problem. Essentially, this means discerning the impact of each feature on a given prediction. Specifically, A^3R conceives feature attribution as an argumentation process, treating features as arguments that can support, attack, or remain neutral to a prediction. The key idea of A^3R is that users' ratings towards items depend on items' features. For example, in movie recommendations, movies' features mainly include actors, genres, and directors. The importance of feature types varies across users. Some users may especially prefer certain genres of movies, while some users may prefer movies starred by certain actors. Therefore it is important to capture this difference to accurately model users' preferences. A^3R computes the importance of different feature types (e.g. the feature types of a movie include the actors, the directors and the genres of the movie.). The importance is further applied to compute users' ratings towards item features (e.g. the value of the feature type "actors" of a movie can be Tom Hanks). Users' ratings towards different item features are weighted by the importance of feature types and are aggregated to compute the final ratings towards items. We define an interaction-tailored AF where features of items are arguments and users' ratings towards features are the strength of these arguments, these arguments may support, attack, or remain neutral users' ratings towards items depending on users' ratings. This means that the computation of A^3R is endowed with explicit semantics, therefore, the prediction of ratings can be traced, which leads to model-intrinsic explanations.

We show that with the help of AFs, the rating prediction of A^3R can be mapped into an argumentation procedure, and each step is endowed with explicit and intuitive semantics. More precisely, A^3R has the following applications in generating explanations: (1) generating toy templates for explaining recommended items and those not recommended; (2) generating interactive explanations that take into account immediate user feedback; (3) generating contrastive explanations that explain recommendations by comparing the differences between the recommended item and the one not recommended. Besides, A^3R can achieve competitive performances compared with some strong baselines such as NeuMF (He et al., 2017), etc. A^3R performs encouragingly better than existing argumentation-based model *Aspect-Item* (A-I) (Rago et al., 2018, 2020). To summarize, A^3R can generate intuitive explanations in an argumentative way while not sacrificing the accuracy of prediction.

The rest of this chapter is organized as follows: Section 6.2 introduces existing works that generate recommendations through argumentation. Section 6.3 details A^3R: how it predicts ratings. Section 6.4 details how it explains recommendations. Section 6.5 presents the datasets, the experiment setup, and the results of experiments on four datasets from two domains and the analyses related to the results. Finally, Section 6.6 concludes this chapter and bridges it to the next chapter.

6.2 Recommendations through argumentation

Owing to the inherent interpretability of argumentation, the recommendations produced by models based on argumentation naturally come with built-in explanations. For instance, Toniolo et al. (2020) use argumentation to create a decision-support system capable of advising medical treatments to patients grappling with multiple chronic health conditions. Recommendations are substantiated via *Satisfiability Modulo Theories* (Barrett and Tinelli, 2018), and patients receive justifications for these recommendations through argumentation dialogues, enhancing patient understanding and trust in the suggested treatments. In another application, Teze et al. (2015) employ Defeasible Logic Programming (DeLP) (García and Simari, 2004) to generate context-adapted recommendations, highlighting the importance and influence of context in the decision-making process.

Table 6.1: The key notions in Chapter 6

Notion	Description
π_u^t	The importance of type t to user u
t_i	The feature types of item i
at_i	The features of item i
at_i^t	The features of item i of type t (e.g. A movie may have several genres)
$\mathcal{P}_u^{at}$	The predicted rating of user u towards feature at
$\mathcal{R}^+, \mathcal{R}^-, \mathcal{R}^0$	Support, attack and neutral relations among arguments
$\mathcal{R}^-$(a,b)	Argument a attacks argument b, the same for "+" and "0"
$\mathcal{A}$	A set of arguments
rec^i	An argument stating "the item can be recommended to the target user"
$\mathcal{R}^+(rec^i) = \{at \vert (at, rec^i) \in \mathcal{R}^+\}$	The arguments (features) that support rec^i
$\mathcal{R}^-(rec^i) = \{at \vert (at, rec^i) \in \mathcal{R}^+\}$	The arguments (features) that attack rec^i
$\mathcal{R}^0(rec^i) = \{at \vert (at, rec^i) \in \mathcal{R}^+\}$	The arguments (features) that neutralize rec^i
$< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+, \mathcal{R}^0 >$	A tripolar argumentation framework
$\sigma(a)$	The strength of argument a

Furthermore, Briguez et al. (2014) put forth a movie RS driven by a set of pre-established recommendation rules, offering an instance of rule-based argumentation in the recommendation space. In a more recent work, Rago et al. (2018, 2020) introduce the explainable *Aspect-Item (A-I) framework*, which computes users' ratings toward items argumentatively. Specifically, this framework predicts users' ratings for the features of items (e.g., actors of a movie, genres of a movie, etc.), which are then consolidated to compute users' overall ratings toward the items. Our proposed A^3R model draws inspiration from the A-I framework and further expands upon it. We delve into a detailed comparison of these two models in Section 6.4.3, following a comprehensive introduction of the A^3R model.

6.3 Our model

In this section, we detail our model A^3R, the key notions used in this chapter are listed in Table 6.1.

6.3.1 Problem setting

To predict the rating that user u may give to item i, we make the following intuitive assumptions: (1) features of an item, such as the actors or directors of movies, can influence users' preferences towards items. The influences can be positive, negative, or neutral; (2) fn this regard, the features can be viewed as arguments, along with another argument stating "the item can be recommended to the target user", rec^i; (3) such a process can be seamlessly integrated into a TAF (Gabbay, 2016). For a user-item interaction (u, i), the TAF tailored to this interaction is a quadruple: $< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+, \mathcal{R}^0 >$, where $\mathcal{A}$ contains at_i and rec^i. To illustrate this idea, consider the following example where $\mathcal{R}^+(at, rec^i)$ denotes that feature at is an argument that supports the recommendation of item i to user u; $\mathcal{R}^-(at, rec^i)$ denotes that feature at is an argument that attacks the recommendation of item i to user u; $\mathcal{R}^0(at, rec^i)$ denotes that feature at is an argument that does not influence the recommendation of item i to user u. This idea is visually demonstrated in Figure 6.1.

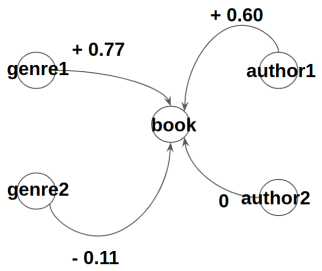


Figure 6.1: A graphical representation of the TAF corresponding to A^3R presented in Figure 6.2. The value on the arc corresponds to the user's rating toward attributes (the argument strength) calculated from Equation 6.3. "+" denotes supports, "−" denotes attacks, "0" denotes neutralizes.

Figure 6.2 is a toy example that shows how A^3R predicts the rating users may give to items in a book recommendation scenario. The features can be regarded as arguments that support/attack/neutralize (see Section 5.3 for the three relations under a TAF) the recommendation of the item. Intuitively, users' like/dislike/neutralize on features influences their like/dislike/neutralize on items. The essential idea of A^3R basically originates from content-based approaches in RSs, which assumes that users' ratings toward items depend on items' features. Users' ratings toward items can be computed by aggregating their ratings toward features of items. To summarize, the objectives of A^3R are the following:

- Estimating the rating that a specified user u would assign to a given item i.
- Determining how each item feature contributes to the predicted rating.
- Assessing the polarity (the positive, negative or neutral influence of each argument (feature)) within the TAF tailored to each user-item interaction.
- Developing the strength function for arguments that adhere to the principles of *weak balance* and *weak monotonicity* as outlined in Section 5.3.2.

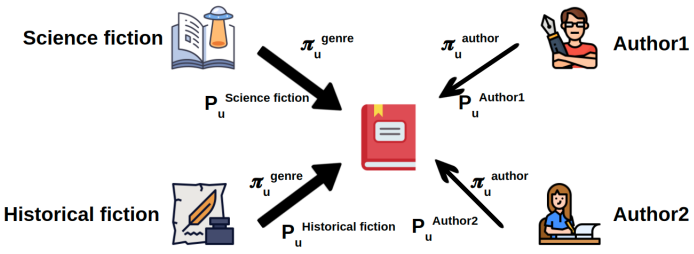


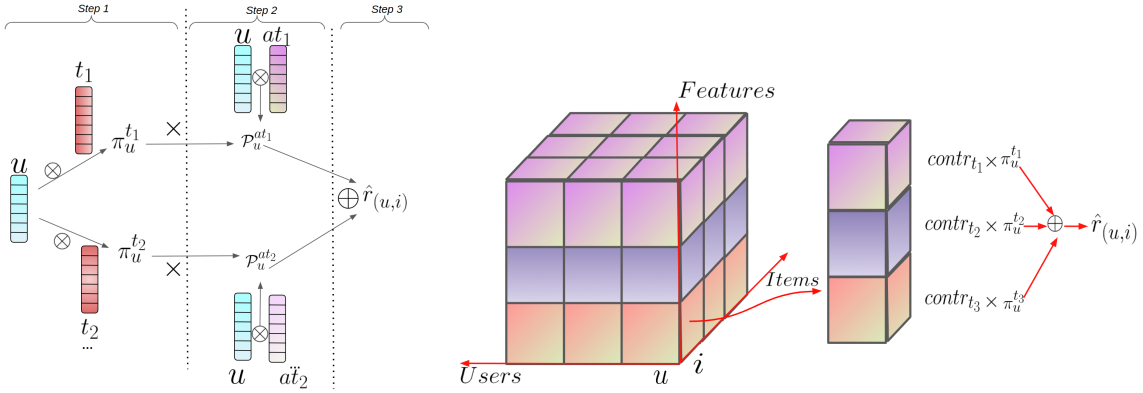
Figure 6.2: A toy example in book recommendation showing how A^3R works. The thickness of the arrow reflects the importance of each feature type. In this toy example, the user accords more importance to the genre of the book.

6.3.2 Main steps of A^3R

Before introducing the steps of our model we would like to clarify features types and features that will be frequently mentioned in the following of this chapter:

- **Feature type**: semantically, it is a high-level concept, which represents the aspects of item features. For example, the feature types of a movie include actors of the movie, genres of the movie, and directors of the movie. Under A^3R we only consider items that are from the same category, they share the same set of feature types.
- **Features**: an item can have several values for a feature type. For example, the genres of a movie can be "action" and "comedy" at the same time. "action" and "comedy" are the features of this movie under the feature type "genre"

Figure 6.3(a) depicts the three steps of A^3R: A^3R first computes the importance of each feature type to each user: the inner product of the vectors (as in Vanilla MF (Koren et al., 2009)) that represent users and the vectors that represent feature types; then users' ratings to each feature are computed: the inner product of the vectors (as in Vanilla MF (Koren et al., 2009)) that represent users and the vectors that represent feature; in sequence, the ratings toward each feature are aggregated by taking into consideration the importance of each feature type to get the final predicted ratings of user u toward item i. Figure 6.3(b) is a graphical illustration of the operations that compute the predicted rating. For each user-item interaction (u, i), each cube represents the contribution of the features of the same type, which will be computed in Equation 6.4. For example, a movie may have several actors, and the contribution of each actor will be aggregated. In what follows, we present in detail the three steps of A^3R. The explanation step will be discussed in Section 6.4.1.



(a) The three steps of A^3R

(b) An graphical illustration of A^3R's computing steps.

Figure 6.3: The computation steps of A^3R, **u**, **t**, **at** are vectors that represent user, feature type, and feature respectively; π_u^t is the importance of feature type t to user u; $\otimes$ denotes the inner product operation; $\times$ denote multiplication; $\oplus$ denotes addition.

Feature type importance (Step 1): the feature type in this work is similar to the relations in the knowledge graphs related to items. Knowledge graphs are composed of *entity-relation-entity*

triplets (Hogan et al., 2021). As a concrete example, the triplet $(HarryPotter, hasAuthor, J.K.Rowling)$ states the fact that the movie *Harry Potter* is written by *J. K. Rowling*. Here, *hasAuthor* is a relation in the knowledge graph related to movies. In this chapter, *hasAuthor* corresponds to the feature type *author*. To quantify the importance of feature type t to user u, we first compute the score between a feature type t and a user u by an inner product operation (exactly the same operation as Vanilla MF (Koren et al., 2009)) between $\mathbf{u}$ and $\mathbf{t}$. This operation has also been adopted to define the importance of relations in knowledge graphs related to items (Wang et al., 2019b,a) [11]

$$\beta_u^t = \mathbf{u} \cdot \mathbf{t} \tag{6.1}$$

where $\mathbf{u} \in \mathbb{R}^d$ and $\mathbf{t} \in \mathbb{R}^d$ are the vectors that represent user u and feature type t respectively; d denotes the dimension of the vectors.

Next, we normalize the score of each feature type to quantify its importance to each user, we follow the convention suggested by Velickovic et al. (2017):

$$\pi_u^t = \frac{exp(LeakyReLU(\beta_u^t))}{\sum_{t \in t_i} exp(LeakyReLU(\beta_u^t))} \tag{6.2}$$

Rating of each feature (Step 2): latent factor models (Koren et al., 2009, Rendle, 2010, Xiao et al., 2017, Rendle et al., 2020) have been proven to be efficient in modeling user-item interactions. To compute the rating toward each feature, we adopt the inner product as in the Vanilla MF (Koren et al., 2009):

$$\mathcal{P}_u^{at} = \mathbf{u} \cdot \mathbf{at} \tag{6.3}$$

where $\mathbf{at}$ denotes the vector that represents feature at.

Aggregating ratings towards features (Step 3): after calculating the importance of each feature type and users' ratings towards each feature, the next step is to compute the rating of user u towards item i by computing the contribution of each feature type t using the following equation:

$$contr_t = \frac{\sum_{at \in at_i^t} \mathcal{P}_u^{at}}{|at_i^t|} \tag{6.4}$$

where at_i^t denotes the features belonging to type t of item i, $|at_i^t|$ denotes the total number of features types of item i. Finally, u's rating towards i is:

$$\hat{r}_{(u,i)} = \sum_{t \in t_i} \pi_u^t * contr_t \tag{6.5}$$

where t_i denotes all the feature types of item i. It should be noted that the actual value of the user u's rating for item i is a real number between -1 and 1, as defined by Rago et al. (2018, 2021). With Equations 6.1 and 6.2, A^3R first computes the importance of each feature type to users; with Equation 6.3, A^3R computes the users' ratings toward features of items; with Equation 6.5, A^3R aggregates these ratings to compute users' ratings toward items. It is noteworthy that Equation 6.5

[11] Note that other functions may be adopted, but for simplicity, we adopt the inner product.

is remarkably similar to Equation 5.1, indicating that our model belongs to the family of generalized additive models. This similarity allows for easy identification of the contribution of each feature, fulfilling the first two goals: **computing users' ratings towards items and determining the contribution of each feature**.

In this section, we have presented how A^3R computes ratings. In the next section, we demonstrate how this process can be seamlessly integrated into an argumentation procedure.

6.4 Explaining recommendations using A^3R

In this section, we will show how A^3R can explain recommendations.

6.4.1 Argumentation scaffolding

In this subsection, we will show that the computation steps in Section 6.3.2 can actually be mapped into an argumentation procedure. We start by defining the TAF corresponding to each user-item interaction following the steps of A^3R, then we define the strength function of arguments that satisfies *weak balance* (Rago et al., 2018) and *weak monotonicity* (Rago et al., 2021), allowing to derive argumentative explanations.

6.4.1.1 Argumentation setting

A^3R first computes users' ratings towards features and then aggregates these ratings to predict ratings toward items. As a result, the features of items can be considered as arguments, and users' ratings towards features can be seen as the strength of these arguments. If A^3R predicts that a user's rating towards a feature of an item is high, then this feature can be viewed as an argument that supports the recommendation of the item. Conversely, if the predicted rating is low, the feature can be considered as an argument against the recommendation of the item. In cases where features do not attack or support, a neutralizing relation is added, represented by $\mathcal{P}_u^{at} = 0$. Therefore, we set $\sigma(at) = \mathcal{P}_u^{at}$ and adopt TAF that contains support, attack, and neutralizing. Moreover, we set $\sigma(rec^i) = \hat{r}(u,i)$. When $\hat{r}(u,i) > 0$, the argument rec^i is stronger if $\hat{r}_{(u,i)}$ is larger[12].

Recall that the true rating $r_{(u,i)}$ is a real number between -1 and 1, then the co-domain of $\mathcal{P}_u^{at}$ is also expected to be between -1 and 1. Therefore, when $\mathcal{P}_u^{at} > 0$, then $\sigma(at) > 0$, indicating that at is an argument that supports rec^i[13]; when $\mathcal{P}_u^{at} = 0$, then $\sigma(at) = 0$, indicating that at is an argument that neutralizes rec^i; when $\mathcal{P}_u^{at} < 0$, then $\sigma(at) < 0$, indicating that at is an argument that attacks rec^i. Therefore, the TAF corresponds to a user-item interaction (u,i) can be defined as follows:

Definition 6.1 *The TAF corresponding to (u,i) is a 4-tuple: $< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+, \mathcal{R}^0 >$ such that:*

- $\mathcal{R}^- = \{(at, rec^i) | \mathcal{P}_u^{at} < 0\}$;
- $\mathcal{R}^+ = \{(at, rec^i) | \mathcal{P}_u^{at} > 0\}$;
- $\mathcal{R}^0 = \{(at, rec^i) | \mathcal{P}_u^{at} = 0\}$.

According to the definition, $\mathcal{P}_u^{at}$ determines the polarity of arguments: if $\mathcal{P}_u^{at}$ is positive then the argument (feature) supports the recommendation of item i to user u; if $\mathcal{P}_u^{at}$ is negative then the

[12]Note that if $\hat{r}_{(u,i)} < 0$ then the smaller $\hat{r}_{(u,i)}$ is, the stronger argument "not recommending item i" is

[13]Semantically, users u prefers feature at

argument (feature) attacks the recommendation of item i to user u; if $\mathcal{P}_u^{at}$ is 0 then the argument neutralizes the recommendation. Therefore, the third goal: **determining the polarity of each argument (feature)**, is fulfilled.

6.4.1.2 Proofs

We will now show that by setting $\sigma(at) = \mathcal{P}_u^{at}$ and $\sigma(rec^i) = \hat{r}(u,i)$, TAF corresponding to (u,i) satisfies *weak balance* formally defined in Definition 5.1 and *weak monotonicity* formally defined in Definition 5.2. Recall that *weak balance* states that attacks (or supports) can be characterized as links between affecters and affectees in a way such that if one affecter is isolated as the only argument that affects the affectee, then the former reduces (increases, resp.) the latter's predicted rating with respect to the neutral point.

Proposition 6.1 *Given the TAF corresponding to* (u,i), $\sigma(at) = \mathcal{P}_u^{at}$ *and* $\sigma(i)$ *satisfy* weak balance.

Proof 6.1 *By inspecting Equations 6.4, 6.5, it can be observed that since users' ratings are transformed into* $[-1,1]$, *the co-domain of* σ *is also* $[-1,1]$. *Case (i): if* $\mathcal{R}^+(rec^i) = \{at\}$, $\mathcal{R}^-(rec^i) = \emptyset$ *and* $\mathcal{R}^0(rec^i) = \emptyset$ *then* $\sigma(at) > 0$, *therefore,* $\mathcal{P}_u^{at} > 0$, *according to Equations 6.4 and 6.5,* $\hat{r}(u,i) > 0$, *indicating that* $\sigma(rec^i) > 0$. *Case (ii): if* $\mathcal{R}^-(rec^i) = \{at\}$, $\mathcal{R}^+(rec^i) = \emptyset$ *and* $\mathcal{R}^0(rec^i) = \emptyset$ *then* $\sigma(at) < 0$, *therefore,* $\mathcal{P}_u^{at} < 0$, *according to Equations 6.4 and 6.5,* $\hat{r}(u,i) < 0$, *indicating that* $\sigma(rec^i) < 0$. *Case (iii): if* $\mathcal{R}^0(rec^i) = \{at\}$, $\mathcal{R}^+(rec^i) = \emptyset$ *and* $\mathcal{R}^-(rec^i) = \emptyset$ *then* $\sigma(at) = 0$, *therefore,* $\mathcal{P}_u^{at} = 0$, *according to Equations 6.4 and 6.5,* $\hat{r}(u,i) = 0$, *indicating that* $\sigma(rec^i) = 0$.

Similar to *weak balance*, *weak monotonicity* characterizes attacks, supports, and neutralizes as links between arguments such that if the strength of one affecter is muted then the strength of its affectees increases, decreases, and remains unchanged, respectively. Therefore, *weak monotonicity* highlights the positive/negative/neutral effect between arguments. In this way, *weak monotonicity* reveals the positive, negative, or neutral effect of an argument.

Proposition 6.2 *Given the TAF corresponding to* (u,i), $\sigma(at) = \mathcal{P}_u^{at}$ *and* $\sigma(i)$ *satisfy the* weak monotonicity.

Proof 6.2 Weak monotonicity *is formulated for two TAFs: from* $< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+, \mathcal{R}^0 >$ *to* $< \mathcal{A}', \mathcal{R}^{-'}, \mathcal{R}^{+'}, \mathcal{R}^{0'} >$, *after modifying certain arguments (e.g. muting certain features). Case (i): if* $at \in \mathcal{R}^-(rec^i)$, *then according to Definition 6.1,* $\mathcal{P}_u^{at} < 0$, *when at is muted then* $\mathcal{P}_u^{at'} = 0$. *According to Equations 6.4 and 6.5,* $\hat{r}'_{(u,i)} > \hat{r}_{(u,i)}$, *indicating that* $\sigma(rec^i)' > \sigma(rec^i)$. *Case (ii): if* $at \in \mathcal{R}^+(rec^i)$, *then according to Definition 6.1,* $\mathcal{P}_u^{at} > 0$, *when at is muted then* $\mathcal{P}_u^{at'} = 0$. *According to Equations 6.4 and 6.5,* $\hat{r}'_{(u,i)} < \hat{r}_{(u,i)}$, *indicating that* $\sigma(rec^i)' < \sigma(rec^i)$. *Case (iii): if* $at \in \mathcal{R}^0(rec^i)$, *then according to Definition 6.1,* $\mathcal{P}_u^{at} = 0$, *when at is muted then* $\mathcal{P}_u^{at'} = 0$. *According to Equations 6.4 and 6.5,* $\hat{r}'_{(u,i)} = \hat{r}_{(u,i)}$, *indicating that* $\sigma(rec^i)' = \sigma(rec^i)$.

Based on Proposition 6.2, the following corollary holds.

Corollary 6.1 *If the predicted rating of on* $at \in at_i$ *is increased:* $\mathcal{P}_u^{at'} > \mathcal{P}_u^{at}$, *then the predicted rating of item* i *also increases:* $\hat{r}'_{(u,i)} > \hat{r}_{(u,i)}$. *Accordingly, if the predicted rating one* $at \in at_i$ *is decreased:* $\mathcal{P}_u^{at'} < \mathcal{P}_u^{at}$, *then the predicted rating of item* i *also decreases:* $\hat{r}'_{(u,i)} < \hat{r}_{(u,i)}$.

Proof 6.3 *By inspecting Equations 6.4, 6.5, Proposition 6.2, it is trivial to see that the corollary holds.*

Propositions 6.1 and 6.2 ensure that the fourth objective: **designing strength function that satisfies *weak balance* and *weak monotonicity***, is fulfilled. The TAF corresponding to the A^3R shows how each feature influences the final rating prediction (recommendation) in an argumentative manner: each feature may support, attack, or neutralize the recommendation of the item, and the user's rating towards each feature determines its polarity. That being said, A^3R is decomposable (Lipton, 2018) (see Section 2.5 for more details), as each computation has explicit meanings that can also be articulated as intelligible (Lou et al., 2012). In the next section, we will detail A^3R's potential applications of generating explanations

6.4.2 Explanation applications

Having defined the steps of A^3R, we will show that, under A^3R, Algorithm 2 is instantiated as $GFGE(A^3R, (u,i), DataProcessing, QueryFunction1, ArgumentationProcedure, QueryFuction2, \emptyset)$:

- $\mathcal{M}$ becomes A^3R detailed in Section 6.3.2.
- $\mathcal{D}$ becomes user-item interactions: (u,i), which includes users' ratings towards items and items' features.
- $\mathcal{P}$, the pre-treatment function: (1) transforms users' ratings into real numbers between -1 and 1 to ensure the polarity of arguments[14]; (2) maps users, feature types, and features into the same vector space. More details are depicted in Figure 6.3.
- $InterpretModel$ becomes the argumentation procedure depicted in Section 6.4.1, where features of items are arguments and users' ratings towards items represent the strength and polarity of arguments. The rating predicting process is mapped into an argumentation procedure.
- $InterpretData$ becomes $QueryFunction1$, $InterpretOutput$ becomes $QueryFunction2$. They are simple query functions: $QueryFunction1(u,i)$ returns $r_{(u,i)}$ and at_i; $QueryFunction2(u,i)$ returns the predicted ratings $\hat{r}_{(u,i)}$
- $PostHocInterpretation$ is empty since each step possesses explicit semantics, there is no need for conducting post-hoc interpretation.
- $Aggregation$ can have different forms according to the application scenarios.

To illustrate the potential application of argumentative explanations generated by A^3R, we now introduce the toy explanation templates, interactive explanations, and contrastive explanations, which are actually different forms of $Aggregation$.

[14]The codes for data pre-processing are available on https://github.com/JinfengZh/a3r

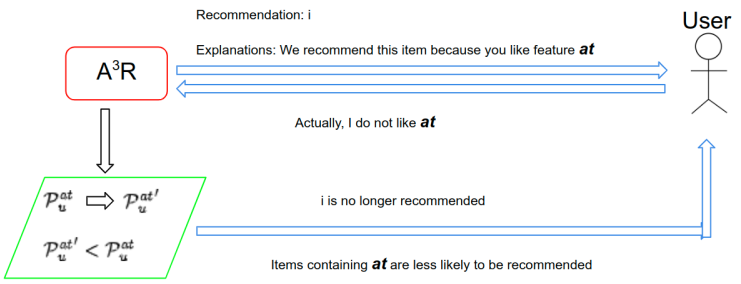
Table 6.2: Several possible explanation scenarios of A^3R, SR denotes "strong recommendation", WR denotes "weak recommendation", NR "not recommended".

Scenario	Content	Example
SR	$at_1 = \arg\max_{at \in at_i} \pi_u^t * \mathcal{P}_u^{at}$ $at_2 = \arg\max_{at \in at_i \setminus at_1} \pi_u^t * \mathcal{P}_u^{at}$	We recommend you this item because you like at_1 and at_2
WR	$at_1 = \arg\max_{at \in at_i} \pi_u^t * \mathcal{P}_u^{at}$ $at_2 = \arg\min_{at \in at_i \setminus at_1} \pi_u^t * \mathcal{P}_u^{at}$	We recommend you this item because you like at_1 although you dislike at_2
NR	$at_1 = \arg\min_{at \in at_i} \pi_u^t * \mathcal{P}_u^{at}$ $at_2 = \arg\min_{at \in at_i \setminus at_1} \pi_u^t * \mathcal{P}_u^{at}$	We do not recommend you this item because you dislike at_1 and at_2

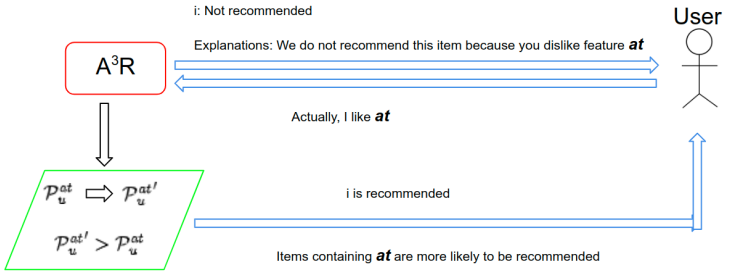
6.4.2.1 Toy explanation templates

In Figure 6.1, we utilize the Tripolar Argumentation Framework (TAF) as a demonstration of user-item interaction, originating from the context illustrated in Figure 6.2. Within this TAF, each feature of the book, as seen in Figure 6.2, is designated as an argument. The strength of each argument directly reflects the user's preference for the corresponding feature. For instance, the user's preference towards the genre of science fiction ($genre1$) is indicated by a rating of $+0.77$, leading to a corresponding argument strength of $+0.77$. This strong preference indicates that $genre1$ functions as a supporting argument for recommending this particular book. Conversely, the user's rating towards historical fiction ($genre2$) is -0.11, making the strength of $genre2$ equivalent to -0.11. This negative rating implies a dislike for historical fiction, suggesting that $genre2$ serves as a counter-argument against the book's recommendation. Moreover, the user's rating for $author1$ stands at $+0.60$, assigning an equal strength to the $author1$ argument. This positive rating implies a preference for books written by $author1$, hence $author1$ strengthens the argument for the book's recommendation. Lastly, the user's rating for $author2$ is neutral, at 0, indicating a neutral argument strength for $author2$. This suggests that $author2$ does not influence the user's decision. In practical terms, should this book be recommended, an explanation may be: "The system recommends this book because it falls under the genre of science fiction ($genre1$) and is authored by $author1$." If the book is not recommended, an explanation could be: "The system does not recommend this book due to its genre of historical fiction, which may not align with your preferences."

After conducting the above analyses, we propose three explanation templates in Table 6.2, similar to the three explanation templates suggested by Rago et al. (2021). For "strong recommendation", we propose selecting the two strongest arguments (aka. features) that support the recommendation of the item. In "weak recommendation", we propose selecting the strongest argument that supports the recommendation of the item and the strongest one that attacks the recommendation of the item. In "not recommended", we propose selecting the two strongest arguments (aka. features) that attack the recommendation of the item. Note that the number of features applied in each scenario can reach up to 3 to avoid too much information for users, this is suggested by Pu and Chen (2007).



(a) A^3R recommends an item and explains this recommendation while the user does not like the feature.



(b) A^3R does not recommend an item and explains this behavior while the user likes the feature.

Figure 6.4: Interaction process between users and A^3R

6.4.2.2 Interactive RSs

Another potential application of the A^3R model lies in its integration with interactive recommender systems (RSs), which take into account immediate user feedback to improve and adapt recommendations on the go (He et al., 2016, Rago et al., 2020, 2021). This idea allows for a more dynamic and customized recommendation process, as compared to traditional, static RSs (He et al., 2016). Following the optimization of A^3R parameters, the calculation of the importance of feature type (as per Equation 6.1) and users' respective ratings towards each feature (see Equation 6.3) can proceed. When users receive a recommendation i from the RS, it comes with the explanations generated by A^3R, such as "We recommend this item because you like feature at." If users indicate a disinterest in feature at, the system can subsequently downgrade the rating of this argument. To illustrate, the system could modify $\mathcal{P}_u^{at}$ to $\mathcal{P}_u^{at'}$, where $\mathcal{P}_u^{at'} < \mathcal{P}_u^{at}$. Consequently, Corallary 6.1 ensures that $\hat{r}(u,i)' < \hat{r}(u,i)$, thus effectively preventing further recommendations of item i. This process is illustrated in Figure 6.4(a). Conversely, in scenarios where users appreciate item i despite it not being recommended (refer to the NR scenario in Table 6.2), the system explains, "We do not recommend this item because you dislike feature at". If users express that they do indeed favor feature at, the system can then increase the rating of this feature. The system could, for example, alter $\mathcal{P}_u^{at}$ to $\mathcal{P}_u^{at'}$, under the condition that $\mathcal{P}_u^{at'} > \mathcal{P}_u^{at}$. Consequently, according to Corallary 6.1, $\hat{r}(u,i)' > \hat{r}(u,i)$. This adjustment ensures that item i is recommended and items possessing feature at are more likely to be favored in future recommendations, which is illustrated in Figure 6.4(b). This mechanism demonstrates that A^3R allows users to provide feedback when recommendations

Algorithm 3: Generating contrastive explanations using A^3R

Input : A^3R: The recommender model detailed in Section 6.3.2
u: a target user
$\mathcal{I}\backslash\mathcal{I}_u$: a set of items that u has not interacted before

Output: Explanations for recommending item

1 $i_{rec} \leftarrow \arg\max_{i \in \mathcal{I}\backslash\mathcal{I}_u} \hat{r}_{(u,i)}$, computed by A^3R; ▷ A recommended item to be explained

2 $at_{pro} \leftarrow \arg\max_{at \in at_{i_{rec}}} \pi_u^t * \mathcal{P}_u^{at}$; ▷ The feature that contributes the most to recommending i_{rec}

3 $t_{pro} \leftarrow$ type of at_{pro}

4 $i_{con} \leftarrow \arg\min_{i \in \mathcal{I}\backslash\{\mathcal{I}_u, i_{rec}\}} \hat{r}_{(u,i)}$

5 $at_{con} \leftarrow \arg\min_{at \in at_{i_{con}}^{t_{pro}}} \pi_u^t * \mathcal{P}_u^{at}$

6 *Explanation* $\leftarrow$ "We recommend i_{rec} instead of i_{con} because you prefer at_{pro} and i_{rec} is at_{pro} while i_{con} is at_{pro}.

are unsuitable, thereby ensuring the scrutability of the explanations as articulated by Tintarev and Masthoff (2015).

6.4.2.3 Contrastive explanations

As we detail in Section 5.3.2, at the core of *weak balance* and *weak monotonicity* is the counterfactual reasoning to analyze the influence of arguments. Therefore, it would be natural to include contrastive explanations (Miller, 2019), which provide information not just about the features of the recommended item that led to it being recommended, but also how these features differ from those of other similar items that were not recommended (see the NR scenario in Table 6.2).

Formally, Algorithm 3 illustrates the process of generating contrastive explanations using A^3R. Initially, the algorithm identifies a recommendation, denoted as i_{rec} (line 1). Subsequently, it computes the feature contributing the most to i_{rec}'s recommendation (line 2), and identifies this feature type as t_{pro} (line 3). The algorithm then identifies the lowest-rated item by the target user, denoted as i_{con} (line 4). For the comparison to be logical, the algorithm specifies the feature of i_{con} that belongs to t_{pro} and receives the lowest rating from the target user (line 5). The recommendation for i_{rec} can then be contrastively explained in relation to i_{con}: "We recommend i_{rec} over i_{con} because you have a preference for at_{pro}. i_{rec} exhibits at_{pro}, whereas i_{con} exhibits at_{con}." An example of a contrastive explanation for a movie recommendation could be:

Example 6.1 *We recommend Movie A over Movie B due to your preference for action movies. While Movie A fits this genre, Movie B is a drama.*

In conclusion, A^3R is a highly adaptable model that provides explanations for both recommended and non-recommended items. Furthermore, it offers users the flexibility to define their own templates according to their specific requirements. We have also demonstrated that A^3R can be incorporated into interactive RSs to generate recommendations that align closely with users' preferences. Moreover, the model facilitates the generation of contrastive explanations by leveraging the inherent counterfactual reasoning within A^3R.

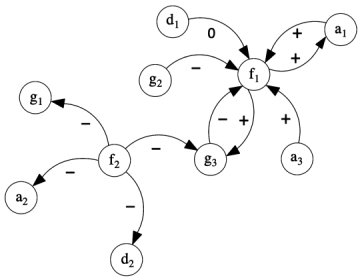


Figure 6.5: An example of the TAF taken from (Rago et al., 2018, 2021), where f_1 and f_2 are movies, other nodes denote feature (actors, directors, and genres) of movies. The user has rated f_2 and the target is to predict this user's rating toward f_1. In this case, the user has only rated one movie, however, as the number increases the user-tailored TAF grows quickly.

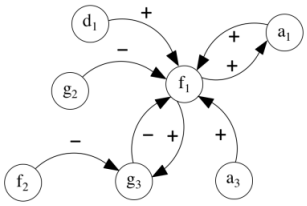


Figure 6.6: An example of the TAF taken from (Rago et al., 2018, 2021), this is a sub-graph of Figure 6.5. Several possible sub-graphs can be extracted to explain the user's rating toward f_1, which takes additional effort to explain recommendations.

6.4.3 Relations with other models

A-I (Rago et al., 2018, 2021) is a hybrid approach: it combines collaborative filtering and content-based method. Rago et al. (2018, 2021) define users' profiles as "collaborative filtering" and "type importance". "Collaborative filtering" defines how users are influenced by "similar users". The "similar users" are defined as follows: $w_{u,v} = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|}$ where $\mathbf{u}$ and $\mathbf{v}$ are vectors that represent users' preferences toward each genre (in their work, the genre of movies). "type importance" is the same as π_u^t in our work, which defines how important each feature type is to users. A-I computes users' ratings toward each feature of an item and these ratings are regarded as the strength of arguments that lead to the final ratings of items, which makes recommendations generated by A-I explainable. In this work, we also compute users' ratings toward features of items, see Section 6.3.

We point out the traits of our A^3R compared with the A-I:

- We do not explicitly explore users' "similar users", which is defined by users' preferences on genres of movies. We argue that this is too limited because users' preferences for actors and directors should be considered, which are two other important feature types.
- Unlike the A-I framework that directly sets the importance to all users (the importance of feature type to all users is the same), A^3R learns the importance for each user supervised by user-item interactions by leveraging the prediction power of Vanilla MF, which is, in fact,

Table 6.3: Comparison of $MF, FM, NFM, AFM, A-I, A^3R$, indicating Matrix Factorization, Factorization Machine, Neural Factorization Machine, Attentional Factorization Machines, Neural Matrix Factorization, Aspect-Item and Attribute-Aware Argumentative Recommender, respectively.

	User-item interactions	Item features	Intrinsic explanations
MF	Yes	No	No
FM	Yes	Yes	No
NFM	Yes	Yes	No
AFM	Yes	Yes	No
$NeuMF$	Yes	Yes	No
$A-I$	Yes	Yes	Yes
A^3R	Yes	Yes	Yes

another form of collaborative filtering. In Section 6.5.4, we will show that A^3R can cluster users according to their preferences (aka, the importance of feature type to users).

- A^3R maps users' ratings towards items and features on an interaction-tailored AF, while in A-I (Rago et al., 2018, 2020, 2021) the authors map users' ratings towards items and features on a user-tailored AF (see Figure 6.5), the graphical representation of a user-tailored AF could be rather large. Therefore, a sub-graph (see Figure 6.6) has to be extracted to explain recommendations. In A^3R, AFs are interaction-tailored (see Figure 6.1). Therefore, all the arguments (features) are directly linked to items and can directly influence users' ratings towards items. As a result, there is no need for extracting sub-graphs of AFs to explain recommendations.

Table 6.3 presents the comparison of A^3R and other related models. It can be concluded that A^3R considers both user-item interactions and features of items like FM, NFM, and AFM. For more details about these latent factor models, please refer to Section 3.4.2. Besides, A^3R can provide model-intrinsic explanations. Compared with state-of-art argumentative explanation model $A-I$, A^3R can model the importance of feature types for each user, allowing finer modeling of users' preferences (detailed in the next section).

6.5 Generating recommendations using A^3R

In the last section, we have shown that for each user-item interaction, the steps of A^3R can actually be mapped into a TAF, where features are regarded as arguments. By setting users' ratings towards these features as the strength of these arguments, *weak balance* and *weak monotocity* are satisfied. With this, we have proposed three explanation applications. Since the problem of explainable recommendations also involves generating high-quality predictions. In this section, we will conduct experiments on four real-world datasets to address the following research questions: (1) can A^3R achieve competitive performance compared to baseline methods? What are the advantages of A^3R compared to baseline methods? (2) how does the importance of feature type affect the performance of A^3R?

Table 6.4: Detailed statistics of the datasets applied in this chapter, $|\mathcal{U}|$, $|\mathcal{I}|$, $|Features|$, $|\mathcal{T}|$, $|Ratings|$ indicates the number of user, items, features, feature types and ratings, respectively.

	$\|\mathcal{U}\|$	$\|\mathcal{I}\|$	$\|Features\|$	$\|\mathcal{T}\|$	$\|Ratings\|$	Sparsity (%)
Netflix	5128	528	1154	3	193081	99.26
MovieLens (Dev)	610	9533	22428	3	100378	98.27
MovieLens 100k	943	1239	4328	3	86572	92.59
DBook2014	5576	2680	1712	2	65961	94.35

6.5.1 Experiments setting

We have conducted experiments on the following real-world datasets:

Netflix, a subset of the Netflix challenge[15]; **MovieLens Development (Dev.)** dataset and **MovieLens 100K** benchmark dataset (Harper and Konstan, 2015); **DBook2014** dataset is the dataset for the *Linked Open Data-enabled Recommender Systems Challenge*[16], to get the types of features (aka. authors and genres) of items, we crawl the DBpedia links[17] provided in the original dataset. For the books whose information is missing, we manually complete them according to *Goodreads*[18], we release the completed dataset and the access is provided[19]. Note that for the **Netflix**, **MovieLens Development (Dev.)** and the **MovieLens 100K** we reuse the version reported by Rago et al. (2021), each of them contains three types of features, namely actors, directors and genres of movies. We filter users who have rated at least 10 items, however, we do not set constraints on actors and directors. For **DBook2014** we follow the same processing. In all the datasets, the ratings given by users are between 1 and 5. The datasets we have worked on include movie and book recommendations, which show the generality of A^3R. Table 6.4 shows the detailed statistics of the four datasets applied in this chapter.

The A^3R we present in Section 6.3 can be applied to miscellaneous predictions such as ranking prediction and explicit rating prediction. This depends on how the objective function tailors to A^3R. In this chapter, the user-item interactions in the four datasets are explicit ratings given by users, therefore, the datasets are adapted to rating prediction tasks. We use the mean squared loss (Gunawardana et al., 2021) to optimize parameters of models:

$$\mathcal{LOSS} = \sum_{(u,i)\in\mathcal{TRA}} (\hat{r}_{(u,i)} - r_{(u,i)})^2 + \lambda \|\mathcal{W}\|_2^2 \tag{6.6}$$

where $\mathcal{TRA}$ is the training set, $\hat{r}_{(u,i)}$ is the predicted rating of user u to item i, λ is the regularization strength to reduce over-fitting, $\mathcal{W}$ denotes the parameters to be learned. Traditionally, error metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) (Silveira et al., 2019) are adopted to evaluate the quality of rating predictions. In many cases (Balog et al., 2019, Kim and Suh, 2019, Rago et al., 2021) where users give explicit ratings, datasets are randomly divided into a training set and a test set, with a ratio 4 : 1 (80% for training and 20% for test). A^3R is implemented using Pytorch (Paszke et al., 2019) and all the parameters are optimized by

[15] https://www.netflixprize.com/
[16] https://2014.eswc-conferences.org/important-dates/call-RecSys.html
[17] https://www.dbpedia.org/
[18] https://www.goodreads.com/
[19] https://github.com/JinfengZh/a3r

mini-batch Adam (Kingma and Ba, 2014). The hyperparameters are optimized through a grid search: the learning rate is tuned on $[10^{-5}, 5*10^{-4}, 10^{-3}, 5*10^{-2}, 10^{-1}]$; the batch size is tuned on $[128, 256, 512, 1024]$; regularization strength is tuned on range $[10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}]$, the size of vectors is tuned on $[4, 8, 16, 32, 64]$. To further reduce the effects of over-fitting, we adopt *Dropout* (Srivastava et al., 2014), and the dropout rate is tuned on range $[0.0, 0.1, 0.2, 0.3, 0.4, 0.5]$.

We compare the following baselines designed for rating prediction:

- Vanilla Matrix factorization (MF) (Koren et al., 2009): this algorithm wins the Netflix challenge. However, it only models user and item interactions.
- Non-negative Matrix Factorization (NMF) (Luo et al., 2014): is a variant of MF.
- K Nearest Neighbours (KNN) (Peterson, 2009): this is a collaborative filtering approach that predicts ratings by finding the most similar users.
- Factorization machine (FM) (Rendle, 2010): this is a strong baseline. Unlike MF, FM considers features related to items and users by capturing the second-order interactions of all information.
- Neural factorization machine (NFM (He and Chua, 2017)): generalizes FM by adding a *Bilinear Interaction* operation.
- Attentional Factorization Machines (AFM) (Xiao et al., 2017): this is another strong baseline, which is a variant of FM. It further computes the importance of each interaction captured by FM.
- Neural matrix factorization (NeuMF) (He et al., 2017): generalizes MF by Multi-Layer Perceptron (MLP). We empirically set the MLP factor size as 8, the sizes of the hidden layer as $(64, 32, 16, 8)$, the GMF (Generalized Matrix Factorization) factor size as 32.
- Aspect-item (A-I) (Rago et al., 2021): is a collaborative filtering approach that explores users' similar items and aspects of items. Note that in A-I, Rago et al. (2021) simply set the importance of actors, directors, genres, and collaborative factor as 0.1, 0.1, 0.1, 0.7, we thus use the same setting for **Netflix**, **MovieLens Development (Dev.)** and **MovieLens 100K**. For **DBook2014** we set the importance of authors, genres, and the collaborative factor as 0.25, 0.25, 0.5, respectively. Note that for **DBook2014**, we just empirically set the values of each factor, which gives more weight to the collaborative factor like Rago et al. (2021).

6.5.2 Experiment results

Table 6.5 presents the experiment results (aka. RMSE and MAE, note that the lower the values the better the performances) of A^3R and other baselines. The following are the observations from results:

- Compared with KNN, the traditional collaborative filtering method, A^3R improves the rating prediction accuracy. This is because KNN only considers user-item interactions, ignoring items' features that can influence users' preferences. Besides, the sparsity of user-item interactions (see Table 6.4) worsens the situation. When user-item interactions are sparse, traditional models cannot accurately predict users' ratings.

Table 6.5: Rating prediction on four datasets. NB indicates neighborhood-based methods, FB indicates factorization-based methods, AB indicates argumentation-based methods.

	Model	Netflix		MovieLens (Dev)		MovieLens 100k		DBook2014	
		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
NB	KNN	1.07	0.79	1.01	0.76	1.15	0.84	1.05	0.86
FB	Vanilla MF	1.01	0.78	0.99	0.75	1.07	0.78	1.02	0.81
	NMF	1.06	0.76	1.01	0.75	1.10	0.81	1.03	0.80
	FM	0.98	0.76	0.93	0.74	1.01	0.80	1.02	0.81
	NFM	0.95	0.73	0.91	0.73	0.98	0.79	0.98	0.81
	AFM	**0.93**	**0.72**	0.90	0.73	0.96	0.78	0.95	0.79
	$NeuMF$	**0.93**	0.73	**0.89**	**0.67**	**0.96**	**0.76**	**0.94**	**0.73**
AB	$A-I$	1.19	0.90	1.19	0.90	1.16	0.91	1.28	1.04
Ours	A^3R	0.95	0.74	0.92	0.71	1.01	0.79	1.01	0.80
	Over $A-I$(%)	20.17	17.78	22.68	21.11	12.93	13.18	21.09	23.07

Table 6.6: Comparison between A^3R and baselines on precision, recall and f1 score. NB indicates neighborhood-based methods, FB indicates factorization-based methods, AB indicates argumentation-based methods.

	Model	Netflix			MovieLens (Dev)			MovieLens 100k			DBook2014		
		Precision	Recall	f1	Precision	Recall	f1	Precision	Recall	f1	Precision	Recall	f1
NB	KNN	0.82	**0.99**	**0.91**	0.84	0.94	0.89	0.85	0.97	0.91	0.92	0.98	**0.95**
FB	Vanilla MF	0.83	0.98	0.90	0.86	**0.94**	**0.90**	0.85	**0.99**	**0.92**	0.92	**0.99**	0.95
	NMF	0.85	0.93	0.89	0.86	0.90	0.88	0.86	0.95	0.90	0.93	0.96	0.94
	FM	0.86	0.80	0.83	0.85	0.88	0.86	0.84	0.90	0.87	0.91	0.93	0.92
	NFM	0.88	0.83	0.86	0.87	0.90	0.88	0.82	0.91	0.86	0.93	0.90	0.92
	AFM	**0.89**	0. 85	0.87	0.88	0.90	0.89	0.90	0.86	0.87	0.92	0.94	0.93
	$NeuMF$	0.88	0.79	0.83	0.87	0.90	0.88	**0.91**	0.86	0.88	0.93	0.94	0.93
AB	$A-I$	0.84	0.87	0.85	0.84	0.85	0.85	0.86	0.93	0.89	0.92	0.91	0.91
Ours	A^3R	0.87	0.80	0.84	**0.90**	0.81	0.86	0.88	0.86	0.87	**0.94**	0.87	0.90

- A^3R outperforms Vanilla MF and NMF, our interpretation is similar to that for KNN: Vanilla MF and NMF considers only user-item interactions. It is encouraging to see that A^3R can achieve competitive performances against some strong baselines such as AFM and $NeuMF$. For example, on **MovieLens 100k**, A^3R performs worse than $NeuMF$ and gains competitive performances compared with NFM and AFM. Under A^3R, only the interactions of vectors representing users and features are captured. Nevertheless, each interaction is endowed with explicit semantics: features are mapped into TAF as in Definition 6.1. The strength of arguments in the TAF satisfies *weak balance* and *weak monotonicity*, which helps to explain how a prediction has been achieved. Compared with A^3R, FM, NFM, and AFM add the following interactions: interactions between items and features, interactions among features, etc. We argue that although A^3R captures fewer interactions, each interaction is endowed with explicit semantics, which reflects users' preferences toward features of items. Besides, the inner product of user vectors and feature vectors are weighted by the importance of the feature type, see Equation 6.5. As opposed to these factorization-based models, A^3R can generate model-intrinsic explanations (see Section 6.4.2 for more details on explaining recommendations in an argumentative way) while keeping competitive rating prediction performances.

- Compared with $A-I$, a recent model that can generate model-intrinsic explanations with

the help of argumentation frameworks, A^3R significantly improves rating prediction on the four datasets. Our interpretation is two-fold: (1) A^3R leverages the power of Vanilla MF to compute users' rating towards features; (2) Besides, A^3R can learn the importance of feature types guided in a data-driven way, while in $A-I$, the importance of feature type is set empirically. Therefore, A^3R can model each user's preference more accurately. Further analyses show that A^3R can cluster users according to the importance they accord to each feature type, see Section 6.5.4 and Figure 6.7; (3) A^3R does not explicitly explore "similar users" like $A-I$. Choosing the most similar k users is a problem that remains unresolved. When k is too small, some "similar users" may be ignored while a too-large k brings noise from less similar users. A^3R explores features of items, which are directly linked to items. Therefore, correctness can be assured and can avoid noise simultaneously.

Due to the subjective nature of ratings, different users might assign different ratings to the same item, even if they both enjoyed it. For instance, one user might give a rating of 5 stars while another might give it 4 or 4.5 stars. To account for such variations in subjective ratings, we follow the approach proposed by Rago et al. (2021) and convert ratings to a binary scale. We report global precision, recall, and F1 score based on the binary scale, and present the results in Table 6.6. The results in Table 6.6 confirm the trends observed in Table 6.5, showing that A^3R in most cases achieves similar performances compared with the baselines. Furthermore, the advantage of using A^3R over other methods extends beyond its superior performance. A^3R provides transparency in the prediction process, allowing for the tracing of the feature attribution procedure. This added interpretability is a crucial advantage compared to other baseline models.

To summarize, the advantages of A^3R are as follows: (1) it achieves competitive performance compared to both context-free and context-aware baselines. These baselines use factorization-based methods such as MF, FM, and some combine neural networks like NeuMF, which makes them difficult to interpret. On the other hand, A^3R provides explicit semantics for each computation and generates argumentative explanations (see Table 6.2 for some examples); (2) compared to the argumentation-based method A-I, A^3R significantly improves prediction accuracy predictions.

6.5.3 Abalation study

Table 6.7: Rating prediction on four datasets.

Model	Netflix		MovieLens (Dev)		MovieLens 100k		DBook2014	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
$AVG-A^3R$	0.99	0.76	0.97	0.72	1.09	0.81	1.06	0.85
A^3R	0.95	0.74	0.92	0.71	1.01	0.79	1.01	0.80

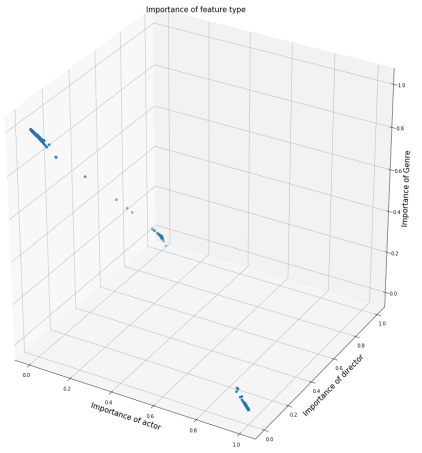
We also study how the importance of feature type computed in Equation 6.2 may influence the performance of A^3R. We, therefore, set the importance of each feature type equal for each user in all datasets, keep steps 2 and step 3 the same and do the same experiments. In **Netflix**, **MovieLens Development (Dev.)** and the **MovieLens 100K**, there are three feature types in each dataset, we, therefore, set the importance of each feature as 0.33 for all users; in **DBook2014**, there are two feature types, we, therefore, set the importance of feature type as 0.5 for all users. We denote the A^3R after this modification as $AVG-A^3R$. Table 6.7 presents the comparison between A^3R and

$AVG - A^3R$. It can be observed that A^3R consistently outperforms $AVG - A^3R$, this is because users' preferences toward each feature usually are not the same. A^3R captures this difference and can make more accurate predictions. Our further analysis in Section 6.5.4 verifies this assertion.

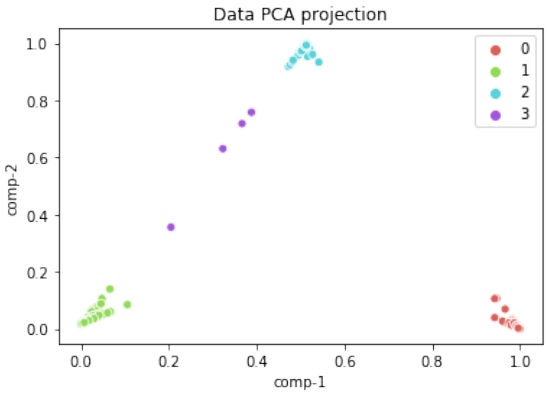
6.5.4 Case study

To further analyze the impact of feature importance, we represent each user by the importance computed by Equation 6.2. We take the dataset **Netflix** as an example. Therefore, in this dataset, each user is represented by a vector of 3 dimensions: $(\pi_u^{actor}, \pi_u^{director}, \pi_u^{genre})$. That said, we put the vectors under a 3-D coordinate system, see Figure 6.7(a). It seems that users from **Netflix** roughly belong to four clusters. To get more insights on users' behaviors in different clusters, we utilized K-means, which is often applied for clustering for its simplicity and effectiveness (Velmurugan and Santhanam, 2010). We, therefore, run K-means and it turns out that 4 clusters fit the dataset best. Then we visualize the results of clustering through PCA (Principal Component Analysis), note that other methods can be applied to reduce dimension, we have chosen PCA for its simplicity, see Figure 6.7. Figure 6.7(b) depicts the four clusters in the **Netflix** dataset. Users from cluster 0 accord almost all of the importance to the actors of movies (see Figure 6.7(c)) , indicating that users in this cluster consider that when choosing movies, movie genre is the most important factor; users from cluster 1 accord almost all of the importance to the genres of movies (see Figure 6.7(d)); users from cluster 2 accord almost all of the importance to the directors of movies (see Figure 6.7(e)); users from cluster 3 are less polarized, they accord almost all of the importance to the directors and genres of movies (see Figure 6.7(f)) It is clear that users' preferences in the three clusters are different, which further verifies our assertion in Section 6.5.3 that A^3R can differentiate users' preferences toward different feature types, thus giving more accurate predictions. Note that for other datasets, **MovieLens Development (Dev.)**, **MovieLens 100K** and **DBook2014**, users can also be clustered according to the feature type importance as depicted in Figure 6.7. More results on the clustering of users on other datasets can be found through `https://github.com/JinfengZh/a3r`.

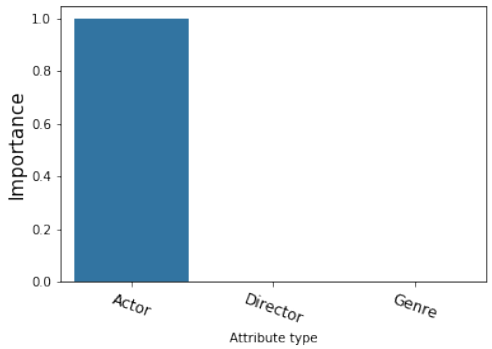
Figure 6.7 provides a comprehensive, global perspective on the operation of A^3R. Rather than focusing on a singular instance, it embraces the entire user set within its scope. Thus, it demonstrates the dual capacity of A^3R as a tool for interpretation—it is proficient not only in elucidating individual instances, as discussed in Sections 6.4.2.1, 6.4.2.2, and 6.4.2.3, but also adept at rendering insights into the overarching behavior of the model on a global scale. This dual function significantly enhances the utility of A^3R in explainable recommendations, as it allows both granular, instance-based insights and a holistic understanding of the model behavior. These global insights could prove instrumental in uncovering broader patterns, trends, or biases in the AI system (Arrieta et al., 2020), thereby improving transparency and aiding in the refinement of system operations.



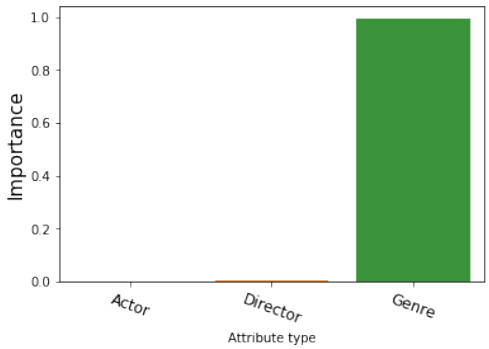
(a) Original projection of users.



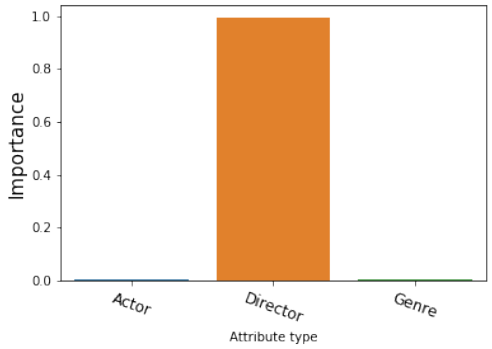
(b) PCA projection of users.



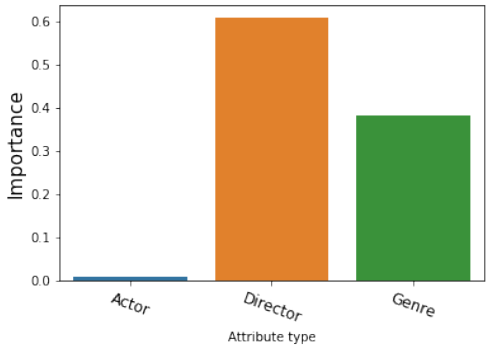
(c) Average importance of each feature type of users from cluster 0.



(d) Average importance of each feature type of users from cluster 1.



(e) Average importance of each feature type of users from cluster 2.



(f) Average importance of each feature type of users from cluster 3.

Figure 6.7: Case study on **Netflix** that shows the clustering of users according to the feature importance learned by A^3R. The histogram shows the average importance of each feature in the cluster.

6.6 Conclusions

In this chapter, we have expanded upon GFGE, as discussed in Algorithm 2 (refer Section 4.3.4), with our focus primarily on model-intrinsic explanations. Our proposition, the Attribute-Aware Argumentative Recommender (A^3R) (Zhong and Negre, 2022a), merges factorization-based methods with argumentation frameworks. We have formulated a TAF for each user-item interaction, treating features as arguments and defining user ratings of these features as the strength of the corresponding arguments.

Our comprehensive theoretical analyses (see Section 6.4.1) underscore that the argument strength function within the TAF complies with the principles of *weak balance* (Rago et al., 2018) and *weak monotonicity* (Rago et al., 2021). This compliance introduces explicit semantics at each progression step, thereby assisting in deciphering the pathway leading to a specific prediction. Consequently, aided by the TAF, we can generate model-intrinsic explanations for every user-item interaction.

To underscore the practical utility of A^3R, we have proposed three diverse explanation scenarios: "strong recommendation", "weak recommendation", and "not recommended". It is also possible to incorporate A^3R into interactive RSs and generate contrastive explanations.

Our empirical testing across four diverse datasets from two different domains shows that A^3R surpasses traditional factorization methods such as KNN and Vanilla MF in terms of performance. Moreover, it is encouraging to see that A^3R can match the performance of advanced models like $NeuMF$. Significantly, A^3R outstrips the state-of-the-art argumentative recommender model $A-I$ in terms of rating prediction accuracy.

However, the A^3R model presents certain limitations. Notably, it does not consider users' contexts, even though such contexts can dramatically influence user preferences (Adomavicius et al., 2021). The development of context-aware explanations, which tailor recommendations to users' contexts, remains an interesting avenue for future research (Zhang et al., 2020, Zhong and Negre, 2022e). Moreover, we must recognize that the persuasive power of arguments can fluctuate depending on the user's context. Thus, future enhancements to the A^3R model should incorporate these valuable context-based insights. In the next chapter, we will extend the A^3R model to incorporate and leverage users' contexts.

Chapter 7

Model-intrinsic instantiation: Context-aware feature attribution through argumentation

Contents

7.1 Introduction . . . 114
7.2 Our model . . . 115
7.2.1 Problem Setting . . . 115
7.2.2 Main steps of CA-FATA . . . 116
7.3 Explaining recommendations using CA-FATA . . . 120
7.3.1 Argumentation scaffolding . . . 120
7.3.2 Explanation applications . . . 122
7.3.3 Relations with other models . . . 124
7.4 Generating recommendations using CA-FATA . . . 124
7.4.1 Experiments setting . . . 124
7.4.2 Experiment results . . . 126
7.4.3 Abalation study . . . 128
7.4.4 Case study . . . 129
7.5 Conclusions . . . 130

In this chapter, we extend the A^3R presented in Chapter 6 by leveraging users' contexts. Like A^3R, it is an instantiation of the *General Framework for Generating Explanation* (GEGE) highlighted by Algorithm 2 in Section 4.3.4 on Page 73, focusing on model-intrinsic explanations. Following the same idea as A^3R, we conceive feature attribution as an argumentation process, treating features as arguments that can support, attack, or remain neutral to a prediction. As we introduce in Section 5.2, the history of feature attribution methods can be traced back to General Additive Models (GAMs), which extend linear regression models by incorporating non-linear relationships between dependent and independent variables. In recent years, gradient-based methods and surrogate models have been applied to unravel complex Artificial Intelligence (AI) systems, but these methods have limitations. GAMs tend to achieve lower accuracy, gradient-based methods

can be difficult to interpret, and surrogate models often suffer from stability and fidelity issues. Furthermore, most existing methods do not consider users' contexts, which can significantly influence their preferences. To address these limitations and advance the current state-of-the-art, we define a novel feature attribution framework called **C**ontext-**A**ware **F**eature **A**ttribution **T**hrough **A**rgumentation (CA-FATA). Our framework harnesses the power of argumentation by treating each feature as an argument that can either support, attack, or neutralize a prediction. Additionally, CA-FATA formulates feature attribution as an argumentation procedure, and each computation has explicit semantics, which makes it inherently interpretable. Therefore, CA-FATA is an instantiation of GFGE highlighted by Algorithm 2 in Section 4.3.4, focusing on model-intrinsic explanations. CA-FATA also easily integrates side information, such as users' contexts, resulting in more accurate predictions (Adomavicius et al., 2021). Our experiments on two real-world datasets demonstrate that CA-FATA, or one of its variants, outperforms existing argumentation-based methods and achieves competitive performances compared to existing context-free and context-aware methods. CA-FATA also ensures the explainability of recommendations, making it a promising framework for practical applications.

The publications related to this chapter are:

- Zhong, J. and Negre, E. (2023). Context-aware feature attribution through argumentation. Accepted by the CARS Workshop at Recsys 2023.
- Zhong, J. and Negre, E. (2022e. Towards improving user-recommender systems interactions. In 2022 IEEE/SICE International Symposium on System Integration (SII), pages 816–820. IEEE.

7.1 Introduction

The Attribute-Aware Argumentative Recommender (A^3R) that we present in Chapter 6 is closely linked to the recently introduced Aspect-Item framework (A-I) (Rago et al., 2018, 2021). Both A-I and A^3R employ argumentation to predict user ratings for items, viewing items and features as arguments that can either support or attack each other, providing argumentative explanations for recommendations. Despite its capacity to substantially enhance recommendation performance in prediction accuracy, A^3R, like A-I, overlooks users' contexts, which can significantly affect their preferences (Adomavicius et al., 2021). Indeed, context holds equal importance within argumentation frameworks (AFs), as the potency of specific arguments can vary in different settings. Hence, it is crucial to consider the context when applying argumentation, as this context-sensitive approach can significantly impact the argument's effectiveness (Teze et al., 2018, García and Simari, 2004).

Recognizing this, several research studies have attempted to integrate context into argumentation-based methods, yielding some promising yet somewhat limited results. For instance, Teze et al. (2018) made notable strides by proposing a model that leverages Defeasible Logic Programming (DeLP) (García and Simari, 2004) to generate context-aware recommendations. Here, context is encapsulated as a set of conditions that steers concluding expressions. However, this work only provided a preliminary demonstration of the system's capabilities through a prototype in a mobile robotic environment (Luis-Argentina, 2008), suggesting a potential for further development and broader application. In the same vein, Zeng et al. (2018) put forth a context-based decision-making framework, also leveraging DeLP, but with an added nuance. In their model, contexts are cast as rules within an assumption-based AF—an extension of the conventional AF. This structural

adaptation allows contexts to moderate the accessibility of goals stemming from decisions, thereby influencing the decision-making procedure. As an illustration, they offered a practical example in a medical scenario, where choosing the right treatment for a patient at risk of blood clotting was at stake. Though both Teze et al. (2018) and Zeng et al. (2018) have paved the way for decision-making frameworks that accommodate contextual factors via argumentation theory, they failed to address a seminal issue in Recommender Systems (RSs)—rating prediction. This gap is vital as it directly impacts the accuracy and usefulness of recommendations. Turning to a slightly different direction, Budán et al. (2020a) considered the impact of context in argumentation, devising methods to compute the similarity of arguments in different contexts. While this approach is novel, it leaves out a comprehensive examination of how context influences the decision-making process.

In light of these considerations, our proposed model, Context-Aware Feature Attribution Through Argumentation (CA-FATA) (Zhong and Negre, 2023), diverges from the approach by Budán et al. (2020a). It centers on a more profound exploration of how contextual elements might steer decision-making processes in an argumentative manner. Like in A^3R we design the strength function of arguments that satisfies the generalized concepts of *weak balance* (Rago et al., 2018) and *weak monotonicity* (Rago et al., 2021), which characterize how arguments influence the decision-making process (detailed in Section 5.3.2). Further discussions on this innovative framework will follow in the subsequent sections.

The rest of this chapter is organized as follows: Section 7.2 details CA-FATA: how it predicts ratings. Section 7.3 details how it explains recommendations. Section 7.4 presents the datasets, the experiment setup, and the results of experiments on real-world datasets and the analyses related to the results. Finally, Section 7.5 concludes this chapter and bridges it to the next chapter.

7.2 Our model

In this section, we detail our model A^3R, the key notions used in this chapter are listed in Table 7.1.

7.2.1 Problem Setting

The concept of "context" has been a focal point of extensive investigation across myriad disciplines, despite the absence of a universally accepted definition. In our exploration, we embrace the definition put forth by Abowd et al. (1999), which considers "context as any information that can be used to characterize the situation of an entity." A context-sensitive situation, symbolized as cs, is typically an amalgamation of numerous contextual conditions. For example, various factors such as "Companion", "Day of the week", and "Location" can shape the context of a user's situation. Each of these contextual factors is comprised of several potential values or contextual conditions. For example, the contextual factor "Companion" could incorporate possible values such as "With family" or "With friends". Consequently, a contextual situation is essentially a confluence of various contextual conditions, such as a user's situation might be described as $(Sunday, at_home, with_family)$. In this chapter, we use the terms "context" and "contextual situation" interchangeably. Context-Aware Recommender Systems (CARSs) utilize this contextual information to refine the modeling of users' preferences, enabling the provision of recommendations that are custom-tailored to their specific contextual situations. Numerous context-aware models, including but not limited to CAMF-C (Baltrunas et al., 2011b) and FM (Rendle, 2010), which are built upon factorization-based methods, have been proposed. Recent advancements have seen the application of neural networks to capture users' preferences in a variety of contextual situations (Unger et al., 2020). However, the

Table 7.1: The key notions in Chapter 7

Notion	Description
cf	A contextual factor
C	The contextual factors to characterize the situation of users
cd	A contextual condition
$\pi^{cf}_{u_{cs}}$	The importance of contextual factor cf to user u under cs
$cs = (cd_1, cd_2, cd_3, \dots)$	A contextual situation
$\pi^{t}_{u_{cs}}$	The importance of type t to user u under cs
t_i	The feature types of item i
at_i	The features of item i
at^t_i	The features of item i of type t (e.g. A movie may have several genres)
$\mathcal{P}^{at}_{u_{cs}}$	The predicted rating of user u towards feature at under cs
$\mathcal{R}^+, \mathcal{R}^-, \mathcal{R}^0$	Support, attack and neutral relations among arguments
$\mathcal{R}^-$(a,b)	Argument a attacks argument b, the same for "+" and "0"
$\mathcal{A}$	A set of arguments
rec^i	An argument stating "the item can be recommended to the target user"
$\mathcal{R}^+(rec^i) = \{at \vert (at, rec^i) \in \mathcal{R}^+\}$	The arguments (features) that support rec^i
$\mathcal{R}^-(rec^i) = \{at \vert (at, rec^i) \in \mathcal{R}^+\}$	The arguments (features) that attack rec^i
$\mathcal{R}^0(rec^i) = \{at \vert (at, rec^i) \in \mathcal{R}^+\}$	The arguments (features) that neutralize rec^i
$< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+, \mathcal{R}^0 >$	A tripolar argumentation framework
$\sigma(a)$	The strength of argument a

challenge arises in explicating the recommendations generated by these methods, as the semantics of the learned latent factors are typically opaque. As such, we gravitate towards the use of argumentation, given its inherent intuitiveness and applicability.

Considering the following recommendation scenario: for a target user u under a contextual situation cs. The features of an item can have positive, negative, or neutral impacts on the prediction. In this regard, the features of items (at_i) can be viewed as arguments, along with another argument stating "the item can be recommended to the target user", rec^i. Such a scenario can be seamlessly integrated into a Tripolar Argumentation Framework (TAF) (Gabbay, 2016). For a user-item interaction (u, i) under cs, the TAF tailored to this interaction is a quadruple: $< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+, \mathcal{R}^0 >$, where $\mathcal{A}$ contains at_i and rec^i. To illustrate this idea, consider the following example where $\mathcal{R}^+(at, rec^i)$ denotes that the feature at has a positive effect on the recommendation of item i. Figure 7.1 visually demonstrates this idea. Hence, the objectives are as follows: (1) to predict the rating assigned by a target user u to a particular item i in a given contextual situation cs; (2) to determine the contribution of each item feature to a prediction under this contextual situation cs; (3) to assess the polarity of each argument (feature) in the TAF; (4) to design the strength function of arguments that comply with the conditions of *weak balance* and *weak monotonicity* as defined above.

7.2.2 Main steps of CA-FATA

The core ideas of CA-FATA are as follows: (1) users' ratings towards items depend on the features of items, which is also a fundamental concept of content-based RSs; (2) the importance of feature

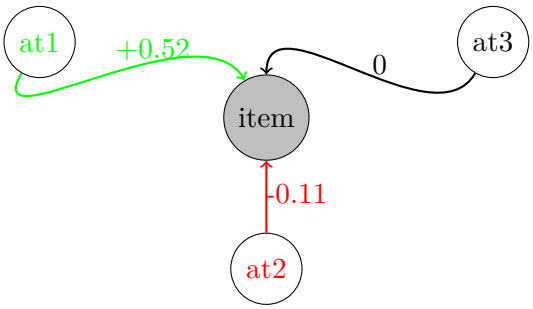


Figure 7.1: A graphical representation of an argumentation procedure in a recommendation scenario. Each node represents an argument, The value on the arc denotes the strength and polarity of the argument, "+" denotes supports, "-" denotes attacks, and "0" denotes neutralizes.

types varies across users. For instance, when choosing a movie, some users may have specific preferences for certain actors while others may prefer movies of particular directors. In the former case, users pay more attention to the feature type ***stars_in***, while in the latter case, they accord more importance to the feature type ***directs***; (3) CA-FATA further extends this idea by noting that users' preferences also vary across contexts (Adomavicius and Tuzhilin, 2011). In the context of movie recommendations, some users are more likely to be influenced by ***companion*** (e.g., with a lover or children) while others may be more likely to be influenced by ***location*** (e.g., home or public place); (4) we map item-features graphs into TAFs, where features of items are regarded as arguments that may attack, support, or neutralize the recommendation of the item; (5) the dialectical relations are learned in a data-driven way and supervised by users' past interactions with items (aka. ratings towards items).

The structure of the Context-Aware Factorization Argumentation Tripolar Analysis (CA-FATA), represented in Figure 7.2, comprises four successive steps. Compared visually with the Attribute-Aware Argumentative Recommender (A^3R) structure, depicted in Figure 6.3(a) in Section 6.3.2, CA-FATA introduces an additional "Step 0". (0) First, we calculate the representation of target users within the given contextual situation. This initial step is crucial to ensure the tailoring of users' preferences and argument dialectical relations to the contextual circumstances. This step's objective is to procure a distinct user representation for each target contextual situation, ensuring that feature importance for each user is also adapted to the specific context. Additionally, it assures the polarity of the arguments (item features) aligns with the target contextual situation. (1) Next, we discern the significance of feature types relative to the aforementioned contextual situation. (2) The third step involves computing the users' ratings of item features within the contextual situation under consideration. This data then serves to delineate the dialectical relations. (3) The final step aggregates the ratings derived in the preceding step to generate users' overall item ratings.

Computing user representation (Step 0): As illustrated in Figure 7.2, the ultimate representation of a user is determined by the user's contextual situation. In this step, our goal is to compute the representation of users that is adapted to the target contextual situation. To achieve this, we begin by computing the score of the contextual factor cf for a user u:

$$\beta_u^{cf} = g(\mathbf{u}, \mathbf{cf}) \tag{7.1}$$

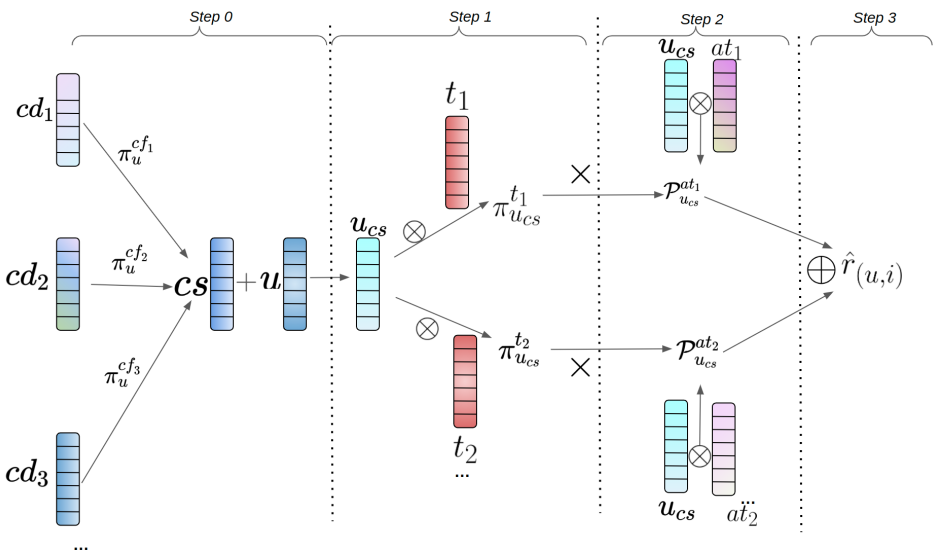


Figure 7.2: Illustration of the steps of CA-FATA. **u**, **t**, **cf**, **at** are vectors that represent user, feature type, contextual factor, and feature respectively; π_u^t is the importance of feature type t to user u; π_u^{cf} is the importance of contextual factor cf to user u; $\otimes$ denotes the inner product operation; $\times$ denote multiplication; $\oplus$ denotes addition.

where g is the inner product[20]. After calculating the score β_u^{cf} of all contextual factors to this user, we normalize the score using Equation 7.2 as described by Velickovic et al. (2017) to obtain the importance of each contextual factor. The importance computed here is similar to the relevance weight introduced by Budán et al. (2020a,b). However, unlike in these two works, where the relevance weight of the context is set empirically, in our work, the importance of the context is learned in a data-driven way. Intuitively, π_u^{cf} characterizes the extent to which user u wants to take contextual factor cf into account. Similar to Equation 6.2 where we compute the importance of feature type, which is proposed by Velickovic et al. (2017). We use the following to compute the importance of contextual factors:

$$\pi_u^{cf} = \frac{exp(LeakyReLU(\beta_u^{cf}))}{\sum_{cf \in C} exp(LeakyReLU(\beta_u^{cf}))} \tag{7.2}$$

In sequence, we compute the representation of the contextual situation cs by summing up all the vectors representing contextual conditions multiplied by π_u^{cf}.

$$\mathbf{cs} = \sum_{cd \in cs} \pi_u^{cf} \mathbf{cd} \tag{7.3}$$

where **cs** is the vector that denotes contextual situation cs. The next step is to aggregate the representation of contextual situation cs with the representation of user u to obtain a specific representation of user u under the contextual situation cs. To avoid having an excessive number of parameters[21], we sum **u** and **cs**. By aggregating information from a contextual situation cs and a

[20]Note that other functions (e.g. MLP) may be adopted, but for simplicity, we adopt the inner product.

[21]Note that other aggregation methods such as concatenation are also possible but more parameters are induced. We leave this exploration for future work.

user u, each user u gets a specific representation $\boldsymbol{u_{cs}}$ under a contextual situation cs.

$$\boldsymbol{u_{cs}} = \mathbf{u} + \mathbf{cs} \tag{7.4}$$

Having obtained the specific representation for user u under the target contextual situation cs, the other steps follow A^3R: computing feature type importance, computing users' ratings towards features, and aggregating ratings towards features.

Computing feature type importance (Step 1):
The feature types in this chapter are similar to the relations in knowledge graphs, which are directed graphs consisting of *entity-relation-entity* triplets (Hogan et al., 2021). For instance, the triplet $(HarryPotter, hasDirector, MikeNewell)$ indicates that the movie *Harry Potter* is directed by *Mike Newell*. Here, *hasDirector* is a relation in the knowledge graph that pertains to movies, and in this chapter, it corresponds to the feature type *director*. We quantify the score between a feature type t and a user u as proposed by Wang et al. (2019a):

$$\beta^t_{u_{cs}} = g(\boldsymbol{u_{cs}}, \mathbf{t}) \tag{7.5}$$

where $\boldsymbol{u_{cs}}$ is computed in the previous step, $\mathbf{t}$ denotes the vector that represents feature type t. After calculating the score $\beta^t_{u_{cs}}$ of all feature importance to this user, we normalize the score using Equation 7.6 as described by Velickovic et al. (2017) to obtain the importance of each feature type.

$$\pi^t_{u_{cs}} = \frac{exp(LeakyReLU(\beta^t_{u_{cs}}))}{\sum_{t \in t_i} exp(LeakyReLU(\beta^t_{u_{cs}}))} \tag{7.6}$$

Computing users' ratings towards features (Step 2):
To compute users' ratings towards features, we adopt the inner product again:

$$\mathcal{P}^{at}_{u_{cs}} = g(\boldsymbol{u_{cs}}, \mathbf{at}) \tag{7.7}$$

According to Equations 7.2, 7.3, and 7.4, the representation of a user under one context differs from that under another context. As a result, the representation of user u_{cs} is specific to each context, and the importance of feature type and user's rating towards features is also context-aware.

Aggregating ratings towards features (Step 3): after calculating the importance of each feature type and users' ratings towards each feature, the next step is to compute the rating of user u towards item i by computing the contribution of each feature type t using the following equation:

$$contr_t = \frac{\sum_{at \in at^t_i} \mathcal{P}^{at}_{u_{cs}}}{|at^t_i|} \tag{7.8}$$

where at^t_i denotes the features belong to type t of item i. Finally, u's rating towards i under cs is:

$$\hat{r}_{(u,i)} = \sum_{t \in t_i} \pi^t_{u_{cs}} * contr_t \tag{7.9}$$

where t_i denotes all the feature types of item i. We would like to highlight that the actual user u's rating for item i is a real number ranging between -1 and 1, in accordance with the definitions

provided in earlier works (Rago et al., 2018, 2021). Furthermore, the striking resemblance between Equation 7.9 and Equation 5.1 is worth noting, signifying that our model belongs to the extended family of generalized additive models. This correlation facilitates straightforward identification of each feature's contribution, thereby achieving the first two objectives: **calculating users' ratings for items and determining each feature's contribution defined in Section 7.2.1**.

7.3 Explaining recommendations using CA-FATA

In this section, we depict the details for explaining recommendations generated by CA-FATA

7.3.1 Argumentation scaffolding

In the following subsection, we aim to demonstrate how the computational steps outlined in Section 7.2.2 can be accurately mapped onto an argumentation procedure. Our process begins by establishing a Targeted Argumentation Framework (TAF) that corresponds to each user-item interaction. This is accomplished by adhering to the steps outlined in CA-FATA. Subsequently, we delineate a strength function for arguments that aligns with both the principles of *weak balance* (Rago et al., 2018) and *weak monotonicity* (Rago et al., 2021). This approach paves the way for determining the contribution of individual features.

7.3.1.1 Argumentation setting

CA-FATA first computes users' ratings towards features and then aggregates these ratings to predict ratings towards items. As a result, the features of items can be considered as arguments, and users' ratings towards features can be seen as the strength of these arguments. If CA-FATA predicts that a user's rating towards a feature of an item is high (under a target contextual situation cs), then this feature can be viewed as an argument that supports the recommendation of the item under cs. Conversely, if the predicted rating is low, the feature can be considered as an argument against the recommendation of the item under cs. In cases where features do not attack or support, a neutralizing relation is added, represented by $\mathcal{P}^{at}_{u_{cs}} = 0$. Therefore, we set $\sigma(at) = \mathcal{P}^{at}_{u_{cs}}$ and adopt TAF that contains support, attack, and neutralizing. Moreover, we set $\sigma(rec^i) = \hat{r}(u,i)$. When $\hat{r}(u,i) > 0$, the argument rec^i is stronger if $\hat{r}_{(u,i)}$ is larger[22].

Recall that the true rating $r_{(u,i)}$ is a real number between -1 and 1, then the co-domain of $\mathcal{P}^{at}_{u_{cs}}$ is also expected to be between -1 and 1. Therefore, when $\mathcal{P}^{at}_{u_{cs}} > 0$, then $\sigma(at) > 0$, indicating that at is an argument that supports rec^i[23]; when $\mathcal{P}^{at}_{u_{cs}} = 0$, then $\sigma(at) = 0$, indicating that at is an argument that neutralizes rec^i; when $\mathcal{P}^{at}_{u_{cs}} < 0$, then $\sigma(at) < 0$, indicating that at is an argument that attacks rec^i. Therefore, the TAF corresponds to a user-item interaction (u,i) under cs can be defined as follows:

Definition 7.1 *The TAF corresponding to (u,i) under cs is a 4-tuple: $< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+, \mathcal{R}^0 >$ such that:*

- $\mathcal{R}^- = \{(at, rec^i) | \mathcal{P}^{at}_{u_{cs}} < 0\}$;
- $\mathcal{R}^+ = \{(at, rec^i) | \mathcal{P}^{at}_{u_{cs}} > 0\}$;
- $\mathcal{R}^0 = \{(at, rec^i) | \mathcal{P}^{at}_{u_{cs}} = 0\}$.

[22] Note that if $\hat{r}_{(u,i)} < 0$ then the smaller $\hat{r}_{(u,i)}$ is, the stronger argument "not recommending item i" is

[23] Semantically, users u prefers feature at

According to the definition, $\mathcal{P}^{at}_{u_{cs}}$ determines the polarity of arguments: if $\mathcal{P}^{at}_{u_{cs}}$ is positive then the argument (feature) supports the recommendation of item i to user u; if $\mathcal{P}^{at}_{u_{cs}}$ is negative then the argument (feature) attacks the recommendation of item i to user u; if $\mathcal{P}^{at}_{u_{cs}}$ is 0 then the argument neutralizes the recommendation. Therefore, the third goal: **determining the polarity of each argument (feature) under the target contextual situation defined in Section 7.2.1**, is fulfilled.

7.3.1.2 Proofs

We will now show that by setting $\sigma(at) = \mathcal{P}^{at}_{u_{cs}}$ and $\sigma(rec^i) = \hat{r}(u,i)$, TAF corresponding to (u,i) under *cs* satisfies *weak balance* and *weak monotonicity*. Recall that *weak balance* states that attacks (or supports) can be characterized as links between affecters and affectees in a way such that if one affecter is isolated as the only argument that affects the affectee, then the former reduces (increases, resp.) the latter's predicted rating with respect to the neutral point.

Proposition 7.1 *Given the TAF corresponding to (u,i) under cs, $\sigma(at) = \mathcal{P}^{at}_{u_{cs}}$ and $\sigma(i)$ satisfy* weak balance.

Proof 7.1 *By inspecting Equations 7.8, 7.9, it can be observed that since users' ratings are transformed into $[-1,1]$, the co-domain of σ is also $[-1,1]$. Case (i): if $\mathcal{R}^+(rec^i) = \{at\}$, $\mathcal{R}^-(rec^i) = \emptyset$ and $\mathcal{R}^0(rec^i) = \emptyset$ then $\sigma(at) > 0$, therefore, $\mathcal{P}^{at}_{u_{cs}} > 0$, according to Equations 7.8 and 7.9, $\hat{r}(u,i) > 0$, indicating that $\sigma(rec^i) > 0$. Case (ii): if $\mathcal{R}^-(rec^i) = \{at\}$, $\mathcal{R}^+(rec^i) = \emptyset$ and $\mathcal{R}^0(rec^i) = \emptyset$ then $\sigma(at) < 0$, therefore, $\mathcal{P}^{at}_{u_{cs}} < 0$, according to Equations 7.8 and 7.9, $\hat{r}(u,i) < 0$, indicating that $\sigma(rec^i) < 0$. Case (iii): if $\mathcal{R}^0(rec^i) = \{at\}$, $\mathcal{R}^+(rec^i) = \emptyset$ and $\mathcal{R}^-(rec^i) = \emptyset$ then $\sigma(at) = 0$, therefore, $\mathcal{P}^{at}_{u_{cs}} = 0$, according to Equations 7.8 and 7.9, $\hat{r}(u,i) = 0$, indicating that $\sigma(rec^i) = 0$.*

Similar to *weak balance*, *weak monotonicity* characterizes attacks, supports, and neutralizes as links between arguments such that if the strength of one affecter is muted then the strength of its affectees increases, decreases, and remains unchanged, respectively. Therefore, *weak monotonicity* highlights the positive/negative/neutral effect between arguments. In this way, *weak monotonicity* reveals the positive, negative, or neutral effect of an argument.

Proposition 7.2 *Given the TAF corresponding to (u,i) under cs, $\sigma(at) = \mathcal{P}^{at}_{u_{cs}}$ and $\sigma(i)$ satisfy* weak monotonicity.

Proof 7.2 Weak monotonicity *is formulated for two TAFs: from $< \mathcal{A}, \mathcal{R}^-, \mathcal{R}^+, \mathcal{R}^0 >$ to $< \mathcal{A}', \mathcal{R}^{-'}, \mathcal{R}^{+'}, \mathcal{R}^{0'} >$, after modifying certain arguments (e.g. muting certain features). Case (i): if $at \in \mathcal{R}^-(rec^i)$, then according to Definition 6.1, $\mathcal{P}^{at}_{u_{cs}} < 0$, when at is muted then $\mathcal{P}^{at\,\prime}_{u_{cs}} = 0$. According to Equations 7.8 and 7.9, $\hat{r}'_{(u,i)} > \hat{r}_{(u,i)}$, indicating that $\sigma(rec^i)' > \sigma(rec^i)$. Case (ii): if $at \in \mathcal{R}^+(rec^i)$, then according to Definition 6.1, $\mathcal{P}^{at}_{u_{cs}} > 0$, when at is muted then $\mathcal{P}^{at\,\prime}_{u_{cs}} = 0$. According to Equations 7.8 and 7.9, $\hat{r}'_{(u,i)} < \hat{r}_{(u,i)}$, indicating that $\sigma(rec^i)' < \sigma(rec^i)$. Case (iii): if $at \in \mathcal{R}^0(rec^i)$, then according to Definition 6.1, $\mathcal{P}^{at}_{u_{cs}} = 0$, when at is muted then $\mathcal{P}^{at\,\prime}_{u_{cs}} = 0$. According to Equations 7.8 and 7.9, $\hat{r}'_{(u,i)} = \hat{r}_{(u,i)}$, indicating that $\sigma(rec^i)' = \sigma(rec^i)$.*

Based on Proposition 7.2, the following corollary holds.

Corollary 7.1 *If the predicted rating of on $at \in at_i$ is increased: $\mathcal{P}^{at\ \prime}_{u_{cs}} > \mathcal{P}^{at}_{u_{cs}}$, then the predicted rating of item i also increases: $\hat{r}'_{(u,i)} > \hat{r}_{(u,i)}$. Accordingly, if the predicted rating one at $\in at_i$ is decreased: $\mathcal{P}^{at\ \prime}_{u_{cs}} < \mathcal{P}^{at}_{u_{cs}}$, then the predicted rating of item i also decreases: $\hat{r}'_{(u,i)} < \hat{r}_{(u,i)}$.*

Proof 7.3 *By inspecting Equations 7.8, 7.9, Proposition 7.2, it is trivial to see that Corallary 6.1 holds.*

Propositions 7.1 and 7.2 guarantee that the fourth objective—**the formulation of a strength function that complies with *weak balance* and *weak monotonicity* defined in Section 7.2.1**—is accomplished. In an argumentative fashion, the TAF related to CA-FATA elucidates how each feature sways the final rating prediction (recommendation): each feature may support, attack, or have no impact on the item's recommendation, with its polarity determined by the user's rating towards that feature. Therefore, CA-TAFA can be considered decomposable (Lipton, 2018), as each computation bears explicit meanings, which can be expressed in an intelligent manner (Lou et al., 2012).

As a running example, Figure 7.1 presents the TAF for a user-item interaction under the contextual situation *cs* (extracted from Figure 7.2). In this TAF, each feature of the item represents an argument. The user's rating towards each feature determines the strength and polarity of the argument, thereby reflecting the user's preference. The strength of argument 1 is $+0.52$, indicating that the user likes feature 1 (e.g., a movie director or actor), and this feature supports the recommendation of the item to the user. The strength of argument 2 is -0.11, indicating that the user does not like feature 2 and that this feature attacks the recommendation of the item to the user. Finally, the strength of argument 3 is 0, indicating that this feature does not influence the user's rating. Note that, according to the four steps in Section 7.2.2, the prediction score could differ under different contexts, even for the same user and item. Therefore, the corresponding TAFs would also differ.

7.3.2 Explanation applications

Having defined the steps of CA-FATA, we will show that, under CA-FATA, Algorithm 2 is instantiated as $GFGE(CA{-}FATA, (u, i, cs), DataProcessing, QueryFunction3, TAF, QueryFuction4, \emptyset)$:

- $\mathcal{M}$ becomes CA-FATA detailed in Section 7.2.2.
- $\mathcal{D}$ becomes user-item interactions under target contextual situation *cs*: (u, i, cs), which includes users' ratings towards items under target contexts and items' features.
- $\mathcal{P}$, the pre-treatment function: (1) transforms users' ratings into real numbers between -1 and 1 to ensure the polarity of arguments[24]; (2) maps users, feature types, and features into the same vector space. More details are depicted in Figure 7.2.
- $InterpretModel$ becomes the argumentation procedure depicted in Section 7.3.1, where features of items are arguments and users' ratings towards items represent the strength and polarity of arguments. The rating predicting process is mapped into an argumentation procedure.

[24]The codes for data pre-processing are available on https://github.com/JinfengZh/ca-fata

Table 7.2: Three explanation templates for user-item interaction under contextual situation $cs = (cd_1, cd_2, cd_3, \dots)$, SR denotes "strong recommendation", WR denotes "weak recommendation", NR "not recommended".

Scenario	Content	Example
SR	$at_1 = \arg\max_{at \in at_i} \mathcal{P}^{at}_{u_{cs}}$ $at_2 = \arg\max_{at \in at_i \setminus at_1} \mathcal{P}^{at}_{u_{cs}}$ $cd = \arg\max_{cd \in cs} \pi^{cf}_{u}$	When cd, we recommend you this item because you like at_1 and at_2.
WR	$at_1 = \arg\max_{at \in at_i} \mathcal{P}^{at}_{u_{cs}}$ $at_2 = \arg\min_{at \in at_i \setminus at_1} \mathcal{P}^{at}_{u_{cs}}$ $cd = \arg\max_{cd \in cs} \pi^{cf}_{u}$	When cd, we recommend you this item because you like at_1 although you dislike at_2.
NR	$at_1 = \arg\min_{at \in at_i} \mathcal{P}^{at}_{u_{cs}}$ $at_2 = \arg\min_{at \in at_i \setminus at_1} \mathcal{P}^{at}_{u_{cs}}$ $cd = \arg\max_{cd \in cs} \pi^{cf}_{u}$	When cd, we do not recommend you this item because you dislike at_1 and at_2.

- *InterpretData* becomes $QueryFunction3$, *InterpretOutput* becomes $QueryFunction4$. They are simple query functions: $QueryFunction3(u, i, cs)$ returns $r_{(u,i,cs)}$ and at_i; $QueryFunction4(u, i)$ returns the predicted ratings $\hat{r}_{(u,i,cs)}$.

- *PostHocInterpretation* is empty since each step possesses explicit semantics, there is no need for conducting post-hoc interpretation.

- *Aggregation* can have different forms according to the application scenarios, which is similar to the case of A^3R

Upon conducting the aforementioned analyses, we introduce three explanation templates encapsulated in Table 7.2. These mirror the three explanation types established for A^3R in Chapter 6, albeit with the inclusion of user-specific contexts. In every scenario, the most influential contextual condition is selected (as determined by Equation 7.1). Under the banner of "strong recommendation", our approach proposes choosing the two most potent arguments (i.e., features) that support the recommendation for the item in question. Within the "weak recommendation" scenario, we propose the selection of the most influential argument in support of the item's recommendation, paired with the strongest argument against it. For situations where an item is "not recommended", our model suggests identifying the two most compelling arguments (i.e., features) that undermine the item's recommendation. Each of these templates weaves together contextual information and corresponding arguments that either substantiate or contest the item's recommendation. To encapsulate, CA-FATA emerges as a versatile model capable of elucidating not only the reasons behind recommending certain items but also the rationale for discouraging others. In addition, CA-FATA offers the flexibility to create their own templates tailored to real needs. Like A^3R, CA-FATA can also be integrated into interactive RSs (see Section 6.4.2.2) and employed to generate contrastive explanations (see Section 6.4.2.3).

Table 7.3: Comparison of $A-I$, A^3R and Ca-FATA

	Features type importance	AFs	Context-aware
$A-I$	Set empirically	User-tailored	No
A^3R	Data-driven	Interaction-tailored	No
CA-FATA	Data-driven	Interaction-tailored	Yes

7.3.3 Relations with other models

Both the A^3R and CA-FATA models have drawn substantial inspiration from the $A-I$ model, as delineated by Rago et al. (2018, 2021). The $A-I$ model cleverly merges elements of collaborative filtering and content-based methods, producing a hybrid approach to item recommendation. It introduces the notion of viewing item features as arguments, an idea that is adopted by both A^3R and CA-FATA. Given that CA-FATA builds upon A^3R as an extension, as outlined in Chapter 6, it naturally inherits the traits of the A^3R model. However, beyond the three distinctions outlined in Section 6.4.3, CA-FATA diverges from $A-I$ and A^3R by leveraging on user contexts. To be specific, CA-FATA computes the representation of each user under specific contextual situations. This means that the importance attributed to each feature type is also adapted to match the target context. At the same time, the TAF designed for user-item interaction is also cognizant of the context. Consequently, CA-FATA has the capability to generate context-aware recommendations and provide explanations that are both aware of the context and based on argumentation. A detailed comparison of A-I, A^3R, and CA-FATA models can be found in Table 7.3. The A^3R model improves upon A-I by leveraging the strength of latent factor models to learn the importance of feature types in a data-driven manner and by constructing interaction-tailored AFs to minimize the effort of extracting subgraphs (see Figures 6.5 and 6.6). CA-FATA further enhances A^3R by harnessing user contexts.

7.4 Generating recommendations using CA-FATA

In the last section, we have shown how to explain recommendations generated by CA-FATA and the explanations applications. In this section, we conduct experiments on two real-world datasets to address the following research questions: (1) Can CA-FATA achieve competitive performance compared to baseline methods? What are the advantages of CA-FATA compared to baseline methods? (2) How does context influence the performance of CA-FATA? (3) How does the importance of feature type affect the performance of CA-FATA?

7.4.1 Experiments setting

We have conducted experiments on the following real-world datasets:

Frappé: this dataset is collected by Baltrunas et al. (2015). This dataset originated from Frappé, a context-aware app recommender. There are 96 303 logs of usage from 957 users under different contextual situations, 4 082 apps are included in the dataset. Following Unger et al. (2020), we apply log transformation to the number of interactions. As a result, the number of interactions is scaled to $0-4.46$. Each contextual situation is composed of 7 contextual conditions: "daytime", "weekday", "isweekend", "homework", "weather", "country", "city" and each of them corresponds

Table 7.4: Basic statistics of the two datasets

Dataset	Frappé	Yelp
#users	585	9 976
#items	3 923	52 298
#interactions	94 716	904 648
sparsity (users × items)	94.47%	99.84%
#Contextual factors	7	8
#Feature types	5	3
Scale	0-4.46	1-5

to a contextual factor. The features of each app include: "category", "number of downloads", "language", "price", "average rating (given by other users)".

Yelp[25]**:** this dataset contains users' reviews on bars and restaurants in metropolitan areas in the USA and Canada. Consistent with previous studies by Zhou et al. (2020), Geng et al. (2022), we use the records between January 1, 2019 to December 31, 2019, which contains 904 648 observations. As contextual factors, we have derived "month", "day of week", "isweekend", "time of the day" from the timestamp of the records; "alone_or_companion" is extracted from the reviews of users, "state" and "city" are provided. The features available include: "stars" (average ratings given by other users), "review count " (the number of reviews received), and "item type".

For the two datasets, we have adopted the 10-core setting, following Wang et al. (2019d), to ensure data quality. This means that only users with at least 10 interactions are kept. Detailed statistics on the pre-processed datasets are presented in Table 7.4.

We compare the following baselines: (1) **Vanilla MF** (Koren et al., 2009): this classic collaborative filtering method only considers user-item pairs and computes the inner product of the vectors representing users and items to make predictions, without taking into account additional information such as users' contexts and item features. (2) **CAMF-C** (Baltrunas et al., 2011b): an extension of **Vanilla MF** that incorporates the global influence of contexts on ratings. (3) **FM** (Rendle, 2010): a strong baseline that models the second-order interactions between all information related to user-item interactions, including users' characteristics, users' context, and item features. (4) **NeuMF** (He et al., 2017): a method that combines matrix factorization and MLP (Multi-Layer Perceptron) to model the latent features of users and items. (5) **ECAM-NeuMF** (Unger et al., 2020): an extension of **NeuMF** that integrates contextual information. Note that the authors in (Unger et al., 2020) do not release the implementation detail, for the NeuMF part, we empirically set the MLP factor size as 8, the sizes of the hidden layer as $(16, 8, 4)$, the GMF (Generalized Matrix Factorization) factor size as 16. This setting also applies to pure NeuMF. (6) **A-I** (Rago et al., 2018, 2021): an argumentation-based framework that computes users' ratings towards features, which are then aggregated to obtain the ratings towards items. Following Rago et al. (2018)[26], we set the "collaborative factor" as 0.8, 20 most similar users are selected, and all the feature importance is set as 0.1.

In these two datasets, users give explicit ratings towards items, therefore the squared loss is

[25] https://www.yelp.com/dataset

[26] For detail please refer to https://github.com/CLArg-group/KR2020-Aspect-Item-Recommender-System.

Table 7.5: Comparison between CA-FATA and baselines on RMSE and MAE, the best are in bold, and the second best are underlined.

Model		Yelp		Frappé	
		RMSE	MAE	RMSE	MAE
Context-free	MF	1.1809	0.9446	0.8761	0.6470
	NeuMF	1.1710	0.8815	0.6841	0.5207
Context-aware	FM	1.1703	0.9412	0.7067	0.5796
	CAMF-C	1.1693	0.9241	0.7283	0.5727
	LCM	1.1687	0.9294	0.6952	0.5396
	ECAM-NeuMF	1.1098	0.8636	0.5599	0.4273
Argumentation-based	A-I	1.3978	1.1205	1.1711	0.9848
Our propositions	**CA-FATA**	**1.1033**	**0.8519**	**0.5154**	**0.3910**

utilized to optimize parameters of CA-FATA:

$$L = \sum_{(u,i,cs)\in\mathcal{T}} (\hat{r}_{(u,i,cs)} - r_{(u,i,cs)})^2 + \lambda\|\Theta\|_2^2 \tag{7.10}$$

where $\mathcal{T}$ is the training set, $\hat{r}_{(u,i,cs)}$ is the predicted rating and $r_{(u,i,cs)}$ denotes the actual rating, λ denotes the regularization parameter to reduce over-fitting, Θ denotes the parameters of CA-FATA. We implement CA-FATA using Pytorch[27] and we optimize the parameters using *mini-batch Adam*. The testing platform is Tesla P100-PCIE, 16GB memory in CPU. The hyper-parameters are tuned through a grid search: the learning rate is tuned on $[10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}]$; the batch size is tuned on $[128, 256, 512, 1024, 2048, 4096]$; regularization parameter is tuned on range $[5 * 10^{-5}, 10^{-4}, 5*10^{-3}, 10^{-3}, 10^{-2}]$. The embedding size is tuned on $[8, 16, 32, 64, 128, 256]$. Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) are selected as the primary evaluation metrics. We follow the convention established by Balog et al. (2019), Kim and Suh (2019), Rago et al. (2021) by splitting the datasets into a training set and a test set with a ratio of $4:1$.

7.4.2 Experiment results

Table 7.5 presents the results of the rating prediction experiments. We observe that CA-FATA performs better than all baselines on both the Yelp dataset and the Frappé dataset, indicating its superiority in handling complex contextual information. The following are some specific observations:

On the Yelp data set:

- CA-FATA surpasses MF in recommendation accuracy due to its consideration of users' contexts and nuanced modelling of their preferences towards features. In comparison, MF solely scrutinizes user-item interactions. NeuMF, despite its combination of matrix factorization and MLP to capture users' and items' latent features, falls short of the performance of CA-FATA. Additionally, CA-FATA's computation steps each possess clear semantics, contributing to the superior explainability of its recommendations. The model further examines the impact of contexts and item features, an aspect that will receive more detailed attention in Section 7.4.3.

[27] Access to source code is provided in https://github.com/JinfengZh/ca-fata

- Among context-aware models, CA-FATA notably outshines CAMF-C, FM, LCM, and ECAM-NeuMF. Beyond its performance, CA-FATA also boasts the ability to explain recommendations in an argumentative fashion, as deliberated in Section 7.3.1.
- Compared to the A-I model, CA-FATA significantly enhances prediction accuracy, showing an RMSE improvement of 21.06% and an MAE improvement of 23.97%. This notable enhancement can be attributed to three facets:
  - Firstly, CA-FATA leverages the strengths of Vanilla MF, leading to more precise predictions.
  - Secondly, CA-FATA learns the significance of different feature types to users under varying contexts, whereas A-I assigns uniform importance to feature types for all users, and does not factor in the users' contexts.
  - Thirdly, the selection of the number of the k most similar users in A-I is not straightforward—too few may overlook influential users, while too many can introduce noise from less similar users. In contrast, CA-FATA manages to sidestep this problem.

On the dataset Frappé:

- CA-FATA performs exceptionally well on this dataset, outperforming all baselines. Compared to context-free methods, CA-FATA improves upon MF by 40.86% and 39.56% on RMSE and MAE, respectively. Compared to NeuMF, CA-FATA achieves reductions in RMSE and MAE by 24.28% and 24.03%, respectively.
- Among the context-aware baselines, CA-FATA outperforms FM, CAMF-C, LCM, and ECAM-NeuMF, demonstrating its ability to model users' preferences under different contexts. Another advantage of CA-FATA is the ability to provide argumentative explanations, which is not possible for these baselines.
- Compared to A-I, like on Yelp, CA-FATA achieves a significant reduction in RMSE and MAE by 55.99% and 60.29%, respectively.

A horizontal comparison of Frappé and Yelp datasets: this comparison shows that CA-FATA performs better on Frappé than on Yelp. We attribute this difference to the sparsity of the dataset, as Yelp is still highly sparse even after applying the 10-core setting, with a sparsity of 99.84% (see Table 7.4), while Frappé has a sparsity of 94.47%.

Due to the subjective nature of ratings, different users might assign different ratings to the same item, even if they both enjoyed it. For instance, one user might give a rating of 5 stars while another might give it 4 or 4.5 stars. To account for such variations in subjective ratings, we follow the approach proposed by Rago et al. (2021) and convert ratings to a binary scale. In our case, ratings greater than or equal to 3 in the Yelp dataset are considered positive, while ratings less than 3 are considered negative. For the Frappé dataset, we set the threshold to 0.9030, which is the average rating across all users. We report global precision, recall, and F1 score based on the binary scale, and present the results in Table 7.6. The results in Table 7.6 confirm the trends observed in Table 7.5, showing that CA-FATA and its variants in most cases outperform the baselines. Furthermore, the advantage of using CA-FATA over other methods extends beyond its superior performance. CA-FATA provides transparency in the prediction process, allowing for the tracing

Table 7.6: Comparison between CA-FATA and baselines on precision, recall and f1 score.

Model	Yelp			Frappé		
	Precision	Recall	f1	Precision	Recall	f1
MF	0.8346	0.8907	0.8617	0.6889	0.7713	0.7278
NeuMF	0.8962	0.9522	0.9234	0.6882	0.8151	0.7463
FM	0.8654	0.9028	0.8837	0.6996	0.7896	0.7337
CAMF-C	0.8701	0.9176	0.8932	0.6913	0.7804	0.7331
LCM	0.8693	0.9207	0.8942	0.6894	0.7769	0.7305
ECAM-NeuMF	0.8849	**0.9827**	**0.9313**	0.7774	0.8124	0.7945
A-I	0.8249	0.8828	0.8528	0.6731	0.7658	0.7164
CA-FATA	**0.8888**	0.9754	0.9303	**0.8043**	**0.8336**	**0.8182**

of the feature attribution procedure. This added interpretability is a crucial advantage compared to other baseline models.

In conclusion, CA-FATA offers significant advantages that set it apart in the realm of explainable recommendations. Firstly, CA-FATA demonstrates competitive performance, effectively holding its own against both context-free and context-aware baseline models. These baseline models primarily employ latent factor models (MF, FM, and CAMF-C) or integrate neural networks (NeuMF and ECAM-NeuMF) to generate recommendations. However, these methods often obscure interpretability, leaving the reasons behind recommendations opaque to the end-users. CA-FATA, conversely, affords clarity and transparency. It is designed to provide explicit semantics for each computational step, leading to argumentative explanations that users can intuitively comprehend (examples are provided in Table 7.2). Secondly, when contrasted with other argumentation-based methods, namely A-I and A^3R, CA-FATA excels not only in delivering superior prediction accuracy but also in generating explanations that incorporate context. Such context-aware explanations are more tailored to users' current situation, resulting in more relevant recommendations. Lastly, an essential characteristic of CA-FATA (shared with A^3R) is the way its AFs are constructed. While in A-I, AFs are built with a user-oriented approach, CA-FATA instead creates these frameworks based on each interaction. This strategic differentiation eliminates the need to extract sub-graphs when generating explanations, making the recommendation process more efficient and the output more understandable.

7.4.3 Abalation study

Table 7.7: Performances of the variants of CA-FATA

Model	Yelp		Frappé	
	RMSE	MAE	RMSE	MAE
FATA	1.1434	0.9059	0.6950	0.5439
AVG-FATA	1.1611	0.9314	0.6970	0.5461
CA-FATA	**1.1033**	**0.8519**	**0.5154**	**0.3910**
AVG-CA-FATA	1.1035	0.8637	0.5254	0.4025

In order to investigate the impact of contextual factors on the performance of CA-FATA, we

propose an alternative approach called FATA, which neglects user contexts and computes ratings using Equations 7.5,7.6, and 7.9, identical to the A^3R (Zhong and Negre, 2022a) we presented in Chapter 6. Results presented in Table 7.7 and 7.8 (**refer to rows 3 and 5**) demonstrate that CA-FATA outperforms FATA, indicating that incorporating user contexts enables more nuanced modeling of user preferences and improves prediction accuracy. Note that FATA is basically a variant of CA-FATA, the difference between CA-FATA and FATA is that FATA does not consider users' contexts and is actually the A^3R (Zhong and Negre, 2022a) model. The version with "AVG" means that the importance of each feature type is set the same for all users. This conclusion is reinforced by the superior performance of CAMF-C over MF and ECAM-NeuMF over NeuMF.

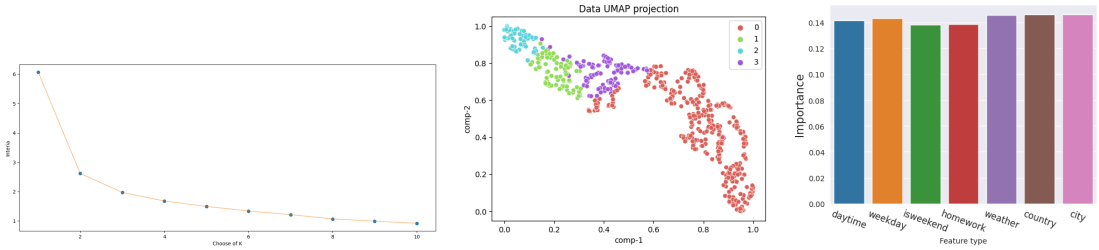
Table 7.8: Performances of the variants of CA-FATA

Model	Yelp			Frappé		
	Precision	Recall	f1	Precision	Recall	f1
FATA	0.8982	0.9361	0.9167	0.6752	0.8139	0.7381
AVG-FATA	**0.8998**	0.9277	0.9135	0.6748	0.8141	0.7380
CA-FATA	0.8888	**0.9754**	**0.9303**	**0.8043**	**0.8336**	**0.8182**
AVG-CA-FATA	0.8894	0.9748	0.9302	0.7982	0.8308	0.8142

To investigate the influence of feature type importance on our proposed model's performance, we introduce AVG-CA-FATA and AVG-FATA for CA-FATA and FATA, respectively (**refer to rows 4 and 6 in Tables 7.7 and 7.8**). In these models, the importance of each feature type is uniformly set for all users. For instance, in Frappé, where there are five feature types, the importance is set to 0.2 for all users, while in Yelp, where there are three feature types, the importance is set to 0.33. Results demonstrate that AVG-CA-FATA performs worse than CA-FATA, as does AVG-FATA when compared to FATA. Furthermore, comparisons between FATA, AVG-FATA, CA-FATA, and AVG-CA-FATA confirm the advantages of incorporating user contexts and modeling feature type importance across users.

7.4.4 Case study

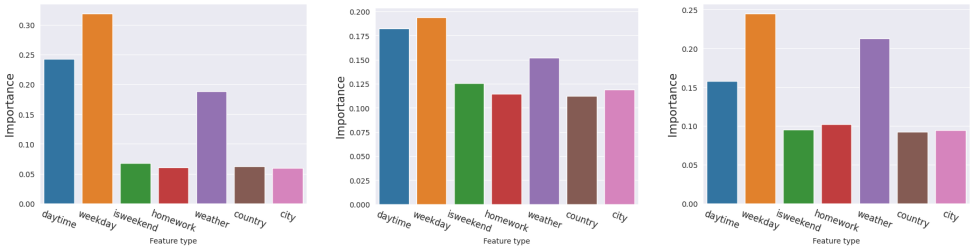
To further visualize the impact of context, we represent each user by their contextual factor importance, computed using Equation 7.2. We use the Frappé dataset as an example, where a vector of seven dimensions represents each user: $(\pi_u^{daytime}, \pi_u^{weekday}, \pi_u^{isweekend}, \pi_u^{homework}, \pi_u^{weather}, \pi_u^{country}, \pi_u^{city})$. Since the vectors are of 7 dimensions, it would not be possible to visualize the projection of users as we did for the **Netflix** dataset in Chapter 6. We therefore first apply K-means clustering for its simplicity and effectiveness (Velmurugan and Santhanam, 2010), and find that four clusters best fit the dataset, as illustrated in Figures 7.3(a) and 7.3(b). We then use UMAP (McInnes et al., 2018) to visualize the clustering results. Note that other dimension reduction techniques (such as PCA and t-SNE (Van der Maaten and Hinton, 2008), for the results of these methods please refer to `https://github.com/JinfengZh/ca-fata`) could also be used, but we choose UMAP because it can preserve the underlying information and general structure of the data. The average importance of each contextual factor for the users in the four clusters is shown in Figures 7.3(c), 7.3(d), 7.3(e), 7.3(f), revealing that users pay different levels of attention to contextual factors in the different clusters. Note that the same visualization applies to the Yelp data, due to limited space, we have omitted the visualization on the Yelp dataset.



(a) Interia with regard to the number of clusters

(b) Cluster of users according to importance of each contextual factor

(c) Average importance of each contextual factor of users from cluster 0



(d) Average importance of each contextual factor of users from cluster 1

(e) Average importance of each contextual factor of users from cluster 2

(f) Average importance of each contextual factor of users from cluster 3

Figure 7.3: A case study on Frappé dataset that shows the clustering of users according to the contextual factor importance learned by $CA-FATA$. The histogram shows the average importance of each contextual factor in the cluster.

In this section, we conducted experiments on two real-world datasets to evaluate the performance of our proposed CA-FATA model. The results demonstrate that CA-FATA outperforms some neural network-based models in terms of recommendation accuracy. Additionally, CA-FATA has the advantage of providing explanations for recommendations through arguments, which is not possible with these neural network-based models. Furthermore, our model demonstrates significant improvement compared to existing argumentation-based models. The ablation study conducted on our proposed model highlights the benefits of leveraging users' contexts and modeling the importance of feature types.

7.5 Conclusions

In light of the interpretability challenges associated with existing feature attribution methods, we present a novel feature attribution framework called **C**ontext-**A**ware **F**eature **A**ttribution **T**hrough **A**rgumentation (CA-FATA). CA-FATA is a feature attribution framework that treats features as arguments that can either support, attack, or neutralize a prediction using argumentation procedures. This approach provides explicit semantics to each step and allows for easy incorporation of user context to generate context-aware recommendations and explanations. The strength function of arguments in CA-FATA is designed to satisfy two important properties: *weak balance* and *weak monotonicity*, which highlights how features influence a prediction. These properties help identify important features and study how they influence the prediction task. Like the A^3R presented in

Chapter 6, we also introduce three explanation scenarios - strong recommendation, weak recommendation, and not recommended, which can be used to explain why items have been recommended or not recommended. Our experimental results show that CA-FATA outperforms several strong baselines regarding RMSE, MAE, precision, recall, and f1 score, highlighting its ability to provide both accuracy and explainability.

In Chapters 6 and 7, we have introduced two recommender models that are intrinsically explainable: A^3R and CA-FATA. The two models instantiate the GFGE (Lines 3 to 6 in Algorithm 2). A^3R and CA-FATA are decomposable (Lipton, 2018) (see Section 2.5 for more details), as each computation has explicit meanings that can also be articulated as intelligible (Lou et al., 2012), which helps to identify the most important feature for the prediction. Therefore, the two models fall into the model-intrinsic explanation scenario.

A^3R and CA-FATA, both draw inspiration from the A-I model as detailed by Rago et al. (2018, 2021). $A-I$ incorporates aspects of collaborative filtering and content-based methods, thus creating a hybrid framework for item recommendation. It propounds the concept of treating item features as arguments, a perspective embraced by both A^3R and CA-FATA. A^3R model varies from the A-I model in several ways. First, A^3R does not merely analyze similar users based on movie genre preferences like A-I, which is restrictive and overlooks important features like preferences for specific actors and directors. Second, instead of assigning the same feature importance to all users as A-I does, A^3R learns each user's feature importance through interactions with items using Vanilla MF, a form of collaborative filtering. Lastly, A^3R employs interaction-tailored AFs to model the prediction process, unlike the user-tailored AFs used by A-I. This approach removes the need for extracting sub-graphs to explain recommendations as all features directly influence user ratings. Given that CA-FATA builds on the foundations laid by A^3R, it quite naturally absorbs the characteristics of the A^3R model. Yet, apart from the three distinctions outlined in Section 6.4.3, CA-FATA separates itself from $A-I$ and A^3R by taking advantage of user contexts. More specifically, CA-FATA calculates the representation of each user within specific contextual situations. This implies that the importance accorded to each feature type is also adapted to conform to the target context. Concurrently, the TAF designed for user-item interaction remains sensitive to the context. Consequently, CA-FATA is equipped to generate context-aware recommendations and provide explanations that are both context-aware and argumentative. A thorough comparison of A-I, A^3R, and CA-FATA models is illustrated in Table 7.3.

To summarize, A^3R model advances over A-I by harnessing the power of latent factor models: instantiating GFGE under the model-intrinsic explanation setting. A^3R can learn the importance of feature types in a data-driven manner. It further eases the explanation process by constructing interaction-tailored Argumentation Frameworks (AFs), which reduces the effort required to extract subgraphs (as demonstrated in Figures 6.5 and 6.6). Expanding on these strengths, CA-FATA introduces an even more refined approach by effectively utilizing user contexts. This increases the accuracy of recommendations and enhances the adaptability of the explanations generated.

GFGE includes another setting: the post-hoc explanation setting, where the inner working mechanism of models is not elucidated. As we introduce in Chapter 1, retaining existing complex RSs in production and adopting post-hoc explanation techniques (Lipton, 2018) to explain recommendations is practical. It allows for the reuse of existing RSs, reducing engineering costs and these RSs usually gain higher performances than those that adopt simple models. In the next chapter, we will contribute to this topic by instantiating the GFGE (depicted in Algorithm 2) in the post-hoc scenario.

Chapter 8

Post-hoc instantiation: Shap-enhanced counterfactual explanations for recommendations

Contents

8.1 Introduction . . . 134
8.2 Motivations . . . 135
8.2.1 Desired properties of explanations . . . 135
8.2.2 Counterfactual explanations . . . 136
8.3 Shap-Enhanced Counterfactual Explanations . . . 138
8.3.1 Problem formulation . . . 138
8.3.2 Main steps of the framework . . . 138
8.3.3 Complexity analysis . . . 141
8.4 Experiments . . . 141
8.4.1 Dataset and recommender model . . . 141
8.4.2 Evaluation settings . . . 142
8.4.3 Experiment results . . . 143
8.5 Conclusions . . . 145

In this chapter, we answer our research question: **"How to explain recommendations?"** from another perspective: **RQ 3.2 "How to explain existing complex RSs using post-hoc explanation techniques while mitigating the limitations of such techniques?"**. Concretely, we instantiate the General Framework for Generating Explanations (GFGE), as defined in Algorithm 2 in Section 4.3.4 on Page 73, for the post-hoc setting. Explaining recommendations has become important for enhancing users' trust and satisfaction. However, explanation methods vary across different recommender models, increasing engineering costs. As recommender systems (RSs) become ever more inscrutable, directly explaining RSs sometimes becomes impossible. Post-hoc explanation methods that do not elucidate the internal mechanisms of RSs are popular approaches. State-of-the-art post-hoc explanation methods like LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017) generate explanations by constructing simpler surrogate models

that approximate the original ones. These methods actually return the features that have contributed the most to a prediction, which is among the feature attribution scope (see Section 5.2 for a comparison with other feature attribution methods). Despite their popularity, the direct application of such methods brings about particular challenges. Firstly, post-hoc explanations may lack fidelity to the original RS due to the absence of a clear elucidation of the system's internal mechanisms. Secondly, the outputs returned by methods like LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017) require a foundational understanding of mathematical concepts, making them less accessible to average users. The pivotal question then emerges: how can we harness post-hoc explanation methods to elucidate any recommender model while sidestepping these constraints? In response to this question, we propose to validate the important features returned by post-hoc explanation methods through counterfactual reasoning. We turn to pertinent research in psychology, philosophy, and cognitive science and propose generating counterfactual explanations utilizing SHAP (Lundberg and Lee, 2017). We have compared LIME and SHAP in Section 5.4, the reason why we have chosen SHAP is its stability and that SHAP can provide both local and global insights into the behaviors of models.

The publication related to this chapter is:

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

8.1 Introduction

Explaining recommendations has attracted much more attention in recent years (Zhang et al., 2020). Based on the models adopted in systems, a variety of methods for generating explanations in RSs have been developed. Similar user/item style explanations (Herlocker et al., 2000), feature-based explanations (Ferwerda et al., 2018), social explanations (Quijano-Sanchez et al., 2017), context-aware explanations (Misztal and Indurkhya, 2015, Li et al., 2020a). However, these methods can be applied to only specific recommender techniques, limiting their adaptability, reusability, and generality. This means that every time a new recommender technique is developed, a corresponding explanation method has to be developed. For more discussions on model-specific and model-agnostic explanations, please refer to Section 2.5.2. A major research topic is: can we find a generic method to explain why recommendations are generated?

Recent advances in eXplainable Artificial Intelligence (XAI) shed light on this topic. In order to explain the outputs of an AI system, two strategies can be applied (Lipton, 2018): (1) adopt transparent models which means that how models work can be easily explained, for example, simple linear models and association rules; (2) provide post-hoc explanations without elucidating precisely how models work. Simple models are easy to understand however they often suffer from lower prediction accuracy, which is known as the trade-off between complexity and accuracy (Doshi-Velez and Kim, 2017). SHAP (Lundberg and Lee, 2017) and LIME (Ribeiro et al., 2016) are among the most prominent tools that can provide post-hoc explanations in literature. They can explain the output of a given instance by estimating the importance of each feature. We note that in RSs, features refer to the variables used to determine a recommendation, for example, it can be users' demographic information or items' characteristics such as prices. There are some works (Guo et al., 2021, Singh and Anand, 2019, Nóbrega and Marinho, 2019) that explain recommendations by directly applying the importance of each feature calculated by LIME and SHAP. The major

limits and concerns of applying directly such methods are: (1) post-hoc explanations may not be faithful to the original RSs and sometimes such explanations may even be wrong (Rudin, 2019); (2) for plain users, understanding the importance of features returned by such methods is non-trivial since complex background mathematical knowledge is required. For example, the outputs of SHAP include the "force_plot" (used to visualize how a prediction is achieved by showing the contribution of each feature please refer to Lundberg and Lee (2017)) figure, this requires that users have basic knowledge of game theory. Therefore, SHAP and LIME are good tools for model developers to visualize the behaviors of models and may help them debug models, however, for plain users, such figures may not be easy for them to understand.

How to make use of post-hoc explanation methods to explain any recommender model while avoiding these limits? Validating the important features returned by these post-hoc explanation methods through counterfactual reasoning is our proposition. More specifically, generating counterfactual explanations that are easily understandable (Wachter et al., 2017) and are of high fidelity (Kaffes et al., 2021), is a potential solution. Unlike LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017) that approximate the original models, counterfactual explanations seek the minimum changes necessary to alter the outputs of the original models. Therefore, counterfactual explanations have high fidelity (Kaffes et al., 2021). A typical counterfactual explanation describes a causal situation " If A did not occur, then B would not have happened" (Molnar, 2020). In this chapter, we show that SHAP can be adapted to generate such explanations. More details about SHAP and LIME have been presented in Section 5.4. The contributions of this chapter include the following: (1) by summarizing state-of-art works, we show that explanations in RSs should be contrastive and selected; (2) we propose **S**hap-**E**nhanced **C**ounterfactual **E**xplanation (SECE) in RSs to generate explanations that are contrastive and selected, to the best of our knowledge, this is the first paper that combines Shapley value method and counterfactual method to address explanations in RSs; (3) we apply our explanation method to interpret a recently proposed context-aware recommender model as a case study to show its effectiveness.

The rest of this chapter is organized as follows: in Section 8.2, we introduce the motivations of our solution; in Section 8.3 we formulate explanations in RSs as counterfactual explanations: list-level explanations and instance-level explanations; in Section 8.4, we present the result of experiments in which we evaluate our method in terms of success rate, time, and length of explanations. In Section 8.5, we conclude and propose perspectives.

8.2 Motivations

8.2.1 Desired properties of explanations

Seeking for explanations to support decisions is the nature of human beings, which is also the case for decision-support systems driven by AI. What constitutes good explanations has been extensively studied in social science, cognitive science, and psychology science (Lipton, 1990, Lombrozo, 2006, Nozick, 1983). Miller (2019) investigated explanations in AI from a social science point of view and concluded that explanations should be contrastive and selected (in a biased manner), for more detailed discussions on the desired properties, please refer to Section 2.3.

- Explanations are contrastive: instead of asking why an event P happened, people prefer asking why event P rather than event Q happened. P and Q refer to the fact and the foil respectively. Accordingly, "why P" type questions and "why P not Q" type questions belong

Table 8.1: The key notions in this chapter

Symbol	Meaning
λ	A black-box model to be explained
l	An interpretable linear model
$x = \{x_1, x_2, x_3 \dots x_p\}$	An instance to be explained
p	The number of features used in λ
$\delta_x(x^{'})$	Proximity between $x^{'}$ and instance x
$\Omega(l)$	The complexity of l
$X^{'}$	A set of instance x's neighbors
ϕ_j	Margin contribution of feature x_j
$C = \{c_1, c_2 \dots\}$	A set of contextual conditions
$CS = (c_1, \dots c_p)$	A contextual situation

to plain-fact questions and contrastive questions. Plain-fact questions should be answered by a "complete" explanation showing details across a "non-interrupted" causal chain across time (Van Bouwel and Weber, 2002). Explanations that show the differences between fact P and foil Q are easier to derive (Lipton, 1990). This means that desired explanations usually do not show full causality analysis but the differences between facts and selected foils, which leads to the next finding: explanations are selected.

- Explanations are selected: people rarely seek the entire and actual cause of an event, rather, they select a limited number of causes as an explanation, potential reason could be that complete causality analyses are usually too large to understand (Hilton, 2017). And contrasts (differences) between facts and foils are the primary way that people select explanations, which means that explanations should be selected to meet up with explainees' demands.

The model we develop in this chapter satisfies the two properties, which will be discussed in Section 8.3

8.2.2 Counterfactual explanations

In this section, we review counterfactual explanations in AI and show that counterfactual explanations are contrastive and selected. The symbols used in this chapter are presented in Table 8.1.

Counterfactual explanations are defined as the minimum change of feature values of an instance to be made in order to alter the output of a model (Wachter et al., 2017). The upsides of counterfactual explanation include the following (also the motivations that we have adopted counterfactual explanations):

- Counterfactual reasoning, which involves constructing alternate versions of reality, is a natural cognitive mechanism that appears early in childhood (O'Connor et al., 2014). This process facilitates learning from experiences, modulates emotional responses, aids decision-making, and improves social interactions. It also becomes an almost instinctive reaction in the face of adverse situations (McCrae, 1987). When transposed to the field of AI, counterfactual reasoning bolsters decision-making, especially when the model's predictions are different

from the expected outcomes (Doshi-Velez and Kim, 2017), offering a systematic and natural methodology for generating explanations.

- Counterfactual explanations focus on individual instances, elucidating the model's behavior for these specific cases, and hence, are classified under local explanations (refer to Section 2.5.2 for a more detailed discussion). Such explanations, presented through particular instances, have been found to streamline the learning process, particularly when the subject matter is intricate (Watson and Shipman, 2008).

- Counterfactual explanations exhibit a contrastive nature, addressing "why P not Q" type question. They discern the critical modifications necessary for occurrence P and identify the distinguishing elements between fact P and foil Q. Owing to these characteristics, counterfactual explanations demonstrate high fidelity (Kaffes et al., 2021) and are in compliance with the General Data Protection Regulation (GDPR) (Wachter et al., 2017).

- Moreover, counterfactual explanations are selective, altering only a minor subset of feature values to transition from fact P to foil Q. When compared to other explanation methods, such as LIME (Ribeiro et al., 2016), counterfactual explanations are generally more accessible and comprehensible for non-expert users (Molnar, 2020).

To the best of our knowledge, counterfactual explanations were initially employed for document classification (Martens and Provost, 2014). The adoption of this explanatory methodology has grown in academic literature due to its user-friendliness, and practical usefulness in deciphering decision rationales (Wachter et al., 2017). A naive approach to finding counterfactual explanations is to enumerate all subsets of features within an instance $x = x_1, x_2, x_3, \ldots, x_p$ (where p denotes the number of features utilized in a black-box model). However, a significant drawback is that the complexity grows exponentially with the increase in the number of features used in the original model.

In the RSs domain, there exist some works that aim to provide counterfactual explanations to explain recommendations. Ghazimatin et al. (2020) and Kaffes et al. (2021) defined counterfactual explanations as the user-item interactions to be removed to result in a different top-ranked recommendation. Ghazimatin et al. (2020) modeled users and items on a heterogeneous information graph. Therefore, the explanation method applies only to specific recommender models. Kaffes et al. (2021) used normalized length and the importance of a candidate to guide the search for counterfactual explanations. However, candidates are not filtered and the search space is still too large, resulting in a longer search time. We argue that, Ghazimatin et al. (2020) and Kaffes et al. (2021) only consider user-item interactions. As a matter of fact, other information such as items' characteristics (Ferwerda et al., 2018), social relationships (Quijano-Sanchez et al., 2017), and users' context (Misztal and Indurkhya, 2015, Li et al., 2020a) can be used to generate explanations. In this chapter, we will provide a counterfactual explanation method that can be applied in any recommender model with the help of SHAP (Lundberg and Lee, 2017).

8.3 Shap-Enhanced Counterfactual Explanations

8.3.1 Problem formulation

In this section, we formulate explanation problems in RSs[28] as counterfactual explanations and present how SHAP can be used to generate contrastive and selected explanations in RSs.

When users require recommendations, RSs usually return a list of top-ranked items. Considering a recommender model λ, the features it uses to generate recommendation for users are $x = \{x_1, x_2, x_3 \dots x_p\}$. The features vary across RSs. For example, in CARSs, the features are users' contextual situations (companion, location, time of the day, etc.). A target user u submits a request for recommendations, by applying the recommender model λ, a list of items $\{i_1, i_2, i_3 \dots, i_{max}\}$ highly ranked are recommended. We note that the items in the list $\{i_1, i_2, i_3 \dots, i_{max}\}$ are ordered according to the predicted scores. Explanations are used to justify either a single recommendation or a list of recommendations. We call them instance-level explanations and list-level explanations respectively.

Instance-level explanations: if a specific item i is to be explained, this corresponds to an instance-level explanation. Counterfactual explanations consist in identifying the minimum set of feature values $x_{exp} \subseteq x$ to be changed in order that item i is no longer in the top-ranked recommendation list.

List-level explanations: similarly, a list-level explanation can be defined as follows: the minimum set of feature values $x_{exp} \subseteq x$ to be changed in order that another set of items $\{i'_1, i'_2, i'_3, \dots, i'_{max}\}$ is recommended.

8.3.2 Main steps of the framework

As we presented in Section 8.2.1, desired properties of explanations in AI include being contrastive and being selected (Miller, 2019). We now present how SHAP can be applied to generate counterfactual explanations in RSs that satisfy these two properties.

Following SHAP, the "game" is the prediction task, and the "players" are the features that the recommender model λ considers. Based on the analysis in Section 8.2, the contribution ϕ_j of each feature can be calculated using SHAP (Lundberg and Lee, 2017). However, interpreting these contributions as counterfactual statements may not be a sensible way to summarize information (Kumar et al., 2020) since ϕ_j is the average marginal contribution of feature j.

Although it is not appropriate to interpret ϕ_j as counterfactual statements, SHAP and the naive approach mentioned above can be combined to compute counterfactual explanations more efficiently. The intuition of this idea is that SHAP formulates explanation problems as computing the SHAP value of each feature. In other words, the SHAP value can represent the importance of features and how a decision has been influenced by these features. Therefore, features whose SHAP values are high may be an "intelligent" start-point for searching (Ramon et al., 2020). The authors adopted this strategy and showed its effectiveness. In their work, candidates being investigated (potential counterfactual explanations) are removed. Their approach is practical since the data operated on is behavioral and textual data. However, in some cases replacing values of these features with substitute values would be more appropriate. For example, in context-aware recommender systems (CARSs hereinafter), removing certain contextual information would make

[28] Since this chapter instantiates Algorithm 2 in Section 4.3.4 on Page 73 in the post-hoc setting, we do not make any assumptions on the recommender model.

the situation completely different, which violates the principle that a counterfactual should be as similar as possible to the instance regarding feature values (Molnar, 2020). Therefore, we believe that the removed contextual condition should be replaced by other possible values.

For instance-level explanations, we propose Algorithm 4[29] to compute a counterfactual explanation for a single recommendation i in the list $\{i_1, i_2, i_3, \ldots, i_{max}\}$. x contains the information used for generating the recommendation, for example, they can be characteristics of items, users, or contextual information. When considering a single item i, each feature j's contribution ϕ_j is computed using SHAP. Features with ϕ_j larger than a threshold are selected and are sorted according to their contributions in a descending manner. Since we wish to select more impacting factors, we only consider features whose ϕ_j are positive, which means that the threshold here is 0. We note that, in extreme cases where an instance's feature is close to the reference value in SHAP, the returned contribution of each feature becomes very close to 0, ordering features by their ϕ_j may not be meaningful, we, therefore, suggest using the naive method.

In sequence, the value of the feature that is ranked at the top is changed, in step 14, x_j is replaced by x_j^{new}. Another list of recommendations $\{i'_1, i'_2, i'_3, \ldots, i'_{max}\}$ is computed given $x_{new} = \{x_1, x_2, x_j^{new} \ldots x_p\}$. If i is no longer in $\{i'_1, i'_2, i'_3, \ldots, i'_{max}\}$ then x_j can serve as a counterfactual explanation. If i is still in $\{i'_1, i'_2, i'_3, \ldots, i'_{max}\}$ after all possible x_j^{new} has been tried, then the top-two[30] ranked features are investigated, and so on, until all features in I_p are explored. Lastly, if $E = \emptyset$, it means that this item can not be explained by Algorithm 4.

Following the same strategy, list-level explanations can be generated. The difference are: the input i becomes a list of recommendations $\{i_1, i_2, i_3, \ldots, i_{max}\}$; in step 16, the condition becomes: $\{i_1, i_2, i_3, \ldots, i_{max}\}$ differs $\{i'_1, i'_2, i'_3, \ldots, i'_{max}\}$. The difference between $\{i_1, i_2, i_3, \ldots, i_{max}\}$ and $\{i'_1, i'_2, i'_3, \ldots, i'_{max}\}$ can be quantified as the similarity between them. The similarity can be quantified by the overlap coefficient (Vijaymeena and Kavitha, 2016). Overlap coefficient is defined as in Equation 8.1. Different thresholds can be chosen to decide whether a counterfactual explanation has been identified. The threshold also indicates the percentage of recommendation explained, which is referred to as **Model Fidelity**[31] by Peake and Wang (2018) and can be calculated by Equation 8.2.

$$Overlap\ Coefficient = \frac{|\{i_1, i_2, i_3, \ldots, i_{max}\} \cap \{i'_1, i'_2, i'_3, \ldots, i'_{max}\}|}{|\{i_1, i_2, i_3, \ldots, i_{max}\}|} \tag{8.1}$$

$$Model\ Fidelity = \frac{|Explainable\ Items \cap Recommended\ Items|}{|Recommended\ Items|} \tag{8.2}$$

Having defined the steps of SECE, we will show that, under SECE, Algorithm 2 is instantiated as $GFGE(\mathcal{M}, (u, i), DataProcessing, QueryFunction5, TAF, QueryFuction6, SECE)$:

- $\mathcal{M}$ can be any recommender models.
- $\mathcal{D}$ becomes user-item interactions: (u, i), which includes users' ratings towards items and items' features.
- $\mathcal{P}$ depends on $\mathcal{M}$.

[29] We will first consider categorical features as it is much easier to determine the values after modifications.

[30] Only modifying x_2 may also lead to a counterfactual explanation, we leave this for future work.

[31] It seems that the success rate is more appropriate since model fidelity concerns whether explanations are aligned with the internal mechanisms of modes. In the following, we will use the term success rate.

Algorithm 4: Generate counterfactual explanations using SHAP

Input : $X = \{x_1, x_2, x_3, \ldots, x_p\}$, information used to generate a list of recommendations; λ, 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 ..., i_max} for user u
Compute the contribution of each features in X using SHAP: r̂_ui = φ_0 + φ_1 + φ_2 + ··· + φ_p
I_p ⊂ X : {I_1, I_2, ...}, indices of features whose φ_{I_p} >= 0
Sort I_p according to φ_{I_p} in a descending manner
for l ← 1 to length(I_p) do
    E_to_expand = I_p[: l] % The l first indices
    for j ∈ E_to_expand do
        Change x_j to x_j^new
        Generate a new list of recommendations {i'_1, i'_2, i'_3, ..., i'_max} given
          x_new = {x_1, x_2, x_j^new, ..., x_p}
        if i ∉ {i'_1, i'_2, i'_3, ..., i'_max} then
            E = E ∪ x_j
            E_counter = E_counter ∪ x_j^new
            Stop = True
            break
        end
    end
    if Stop = True then
        break
    end
end
if i ∈ {i'_1, i'_2, i'_3, ..., i'_max} then
    E = {}
end
return E, E_counter
```

Table 8.2: Details of selected contextual factors in CoMoDa

Context factors	Contextual conditions
Physical wellness	Healthy, Ill
Mood	Positive, Neutral, Negative
Social	Alone, Partner, Friends, Colleagues, Parents
	Public, Family

- *InterpretModel* return simply the output of $\mathcal{M}$ and the mechanism of $\mathcal{M}$ remains unclear.
- *InterpretData* becomes $QueryFunction5$, *InterpretOutput* becomes $QueryFunction6$. They are simple query functions: $QueryFunction5(u, i)$ returns $r_{(u,i)}$ and at_i; $QueryFunction6(u, i)$ returns the predicted ratings $\hat{r}_{(u,i)}$.
- *PostHocInterpretation* becomes SHAP that returns important features according to the marginal contribution of each feature.
- *Aggregation* becomes Algorithm 4, it returns the important features (according to SHAP) validated by counterfactual reasoning, thus in the form of counterfactual explanations.

8.3.3 Complexity analysis

As in Section 8.3.1, in a recommender model f, p features are considered to generate recommendations. Suppose that the number of possible values for feature x_k is n_k, then in naive method, the average maximum number of tries will be $\frac{(n_1+1)*(n_2+1)*(n_3+1)*\cdots*(n_p+1)-2^p}{p}$. By calculating and ordering the contribution of each feature, possible candidates can be selected. Suppose that after selection, the number of candidate features becomes s then the average number of tries will be $\frac{(n_1+1)*(n_2+1)*(n_3+1)*\cdots*(n_s+1)-2^s}{s}$, since s is smaller than p after filtering, the time needed for searching is reduced. Therefore, counterfactual explanations can be found more quickly.

8.4 Experiments

In this section, we present the experimental setup of a case study to show the effectiveness of our explanation method proposed in Section 8.3.

8.4.1 Dataset and recommender model

In this section, we present an application case of Algorithm 4 in which we explain a context-aware model proposed by us (Zhong and Negre, 2022d). The main reason why we choose to explain CARSs is that, contextual information that can significantly influence users' preferences has not been fully inspected to explain recommendations (Zhang et al., 2020, Musto et al., 2020), we wish to contribute to this topic by applying Algorithm 4.

The dataset we apply is **LDOS-CoMoDa**, a dataset for context-aware movie recommendation (Košir et al., 2011). In this dataset, there are 121 users, 1197 movies, 2296 ratings, and 12 context factors. Each contextual situation is composed of 12 contextual conditions: "time", "daytype", "season", "location", "weather", "social", "end emotion", "dominant emotion", "mood", "physical", "decision", "interaction". We note that we select the 3 most impacting contextual factors

(Ferdousi, 2020) to reduce the time required for the experiment. The contextual factors we selected are: "social", "mood", "physical". The possible values for each contextual factor are presented in Table 8.2.

In this dataset, there are some users who have given relatively fewer ratings, which makes it difficult to model their preferences under different contextual situations. Therefore, we only consider users who have given more than 10 ratings. At last, we have 2032 ratings from 33 users. We generate 10 recommendations (aka. $max = 10$) for the users in the corresponding contextual situations.

The recommendation problem in CARSs can be resumed as generating recommendations for a user u given a target contextual situation CS composed of several contextual conditions $\{c_1, c_2, c_3 \dots c_p\}$. For example, a contextual situation can be $CS = \{Weekend, At_home, With_child\}$ which shows that it is the weekend and this user is at home with his child.

The main steps of the black-box model proposed by Zhong and Negre (2022d)[32] mainly include: (1) select a relevant local dataset by users' contexts; (2) a context-free method is applied on the local dataset to generate a list of top-ranked recommendations. Following Algorithm 4, the recommendations can be explained by computing counterfactual explanations. We note that in step 14 since the values in x are all categorical, x_j is replaced by a randomly selected from other possible values for this contextual factor. For example, considering the contextual factor "mood", it has three possible values: "positive", "neutral" and "negative". If $x_j = positive$ then x_{new} will be randomly selected from $\{neutral, negative\}$. After a counterfactual explanation is identified, say, $\{mood_positive \rightarrow mood_negative\}$. This means that user's mood at the moment of recommendation is positive and it is identified to be the most impacting factor. Changing mood from *mood_positive* to *mood_negative* will result in the change of recommendation list from $\{i_1, i_2, i_3, \dots, i_{max}\}$ to $\{i'_1, i'_2, i'_3, \dots, i'_{max}\}$.

In an instance-level explanation (Top-1), "You are in good moods, we, therefore, recommend you the following movies $\{i_1, i_2, i_3, \dots, i_{max}\}$, however, when you are sad we will recommend you the following movies $\{i'_1, i'_2, i'_3, \dots, i'_{max}\}$, without movie i_1".

In a list-level explanation, recommendations are presented as follows: " You are in good moods, we, therefore, recommend you the following movies $\{i_1, i_2, i_3, \dots, i_{max}\}$, however, when you are sad we will recommend you the following movies $\{i'_1, i'_2, i'_3, \dots, i'_{max}\}$". The whole list of recommendations is changed in list-level explanations, which is not the case in instance-level explanations.

Apparently, in both cases, there is a difference between the two lists. Users can easily notice the differences between the two lists, which could help users better understand how the system works.

8.4.2 Evaluation settings

We assess the quality of our SHAP-enhanced explanation method (S-method hereinafter) from three aspects: (1) the average time consumed to compute a counterfactual explanation. The time needed to find the first explanation set E to measure the efficiency of our method; (2) the length of counterfactual explanation, in our case, it will be the size of E in Algorithm 4. An explanation with a smaller size is easier to understand. Therefore, the smaller the size, the better the explanation is (Kaffes et al., 2021); (3) the success rate (similar to the model fidelity defined by Peake and Wang (2018)) of our method defined in Equation 8.2. The larger it is, the more recommendations can be

[32]Note that this chapter focuses on the post-hoc explanation scenario, therefore, models are treated as a black box. More details about the model we propose (Zhong and Negre, 2022d) will be provided in Appendix A.

explained. This in turn indicates that the explanation method has higher fidelity to the original model.

We will compare the effects of list-level explanations and instance-level explanations as we presented in Section 8.3.1. The baselines that we have chosen are the following:

- The heuristic searching method proposed by Kaffes et al. (2021)(H-method hereinafter). For instance-level explanations, the search is guided by the importance of candidates (Kaffes et al., 2021); for list-level explanations, the search is guided by the overlap coefficient of $\{i_1, i_2, i_3, \ldots, i_{max}\}$ and $\{i'_1, i'_2, i'_3, \ldots, i'_{max}\}$.
- Like Kaffes et al. (2021), we also compare the naive method (N-method, hereinafter) that examines candidates in a length-rising order, it first examines candidates whose length is 1 (one contextual factor changed) and then 2 (two contextual factors changed), etc. Values of each contextual factor are randomly selected. Therefore, the difference with Algorithm 4 is that $I_p = x$, which means that candidates are not filtered using SHAP. For this baseline, we call it naive search.
- Another baseline is that adopted by Kaffes et al. (2021), called random search (R-method). This method randomly considers the length of candidates and randomly selects values of each contextual factor to replace the original values.

8.4.3 Experiment results

Following the methodology presented in Section 8.4, we compare our S-method with H-method, N-method, and R-method.

In Tables 8.3, 8.4 and 8.5, the percentage is the threshold to decide whether a counterfactual explanation has been identified. For example, the 100% means that a counterfactual explanation is identified when two lists of recommendations are totally different. The Top-k means the item to be explained. For example, the Top-1 indicates that the first recommendation in the list is explained.

8.4.3.1 Time required for searching

Table 8.3 presents the time required for searching. In terms of efficiency, R-method is not as efficient as S-method, N-method, and H-method except for explaining 100% recommendations in list-level explanations. As we presented in Section 8.2, computing the contribution of contextual factors using SHAP is not efficient. In our case, computing the contribution of each contextual factor in a recommendation is about 2.62 seconds, which is included in Table 8.3 for the S-method. Although the S-method can reduce the number of candidates, the time needed for calculating the contribution of each factor takes a longer time. An exception is for list-level explanations, if we want to explain 100% recommendations, counterfactual explanations are strictly defined. This usually means more changes to be made. By reducing the number of candidates, the S-method reaches higher efficiency.

8.4.3.2 Success rate

Table 8.4 presents the comparison of success rate. In list-level explanations, it can be observed that as the thresholds decrease, the percentage of recommendations explained also declines, which

Table 8.3: Time consumed to search for list-level and instance-level counterfactual explanations (in seconds)

	Instance-level					List-level			
	Top-1	Top-2	Top-3	Top-4	Top-5	100%	90%	70%	50%
S-method	4.82	4.68	4.60	4.56	4.35	**6.82**	4.97	4.29	4.20
N-method	3.69	3.51	3.42	2.84	1.99	7.07	2.86	1.68	1.54
R-method	6.49	5.77	4.54	4.01	3.91	12.21	9.32	5.34	4.67
H-method	3.73	3.64	3.56	2.79	2.20	7.35	3.13	1.94	1.81

Table 8.4: Percentage of recommendations explained

	Instance-level					List-level			
	Top-1	Top-2	Top-3	Top-4	Top-5	100%	90%	70%	50%
S-method	98%	98%	98%	98%	98%	99%	99%	97%	96%
N-method	95%	94%	94%	94%	94%	99%	97%	92%	92%
R-method	92%	91%	90%	90%	90%	94%	93%	92%	91%
H-method	95%	95%	95%	95%	95%	94%	93%	94%	94%

means that the similarity between the original recommendation list and the newly generated recommendation list is higher. As for instance-level explanations, the more the recommendation to be explained is at the back of the recommendation list, the lower the success rate. Since fewer changes are made to features, the original recommendation list changes less (fewer recommendations are explained). We note that the higher success rate here means larger differences between the original recommendation and the newly generated one. As a result, users can see a more evident comparison between the two lists, which may help them better understand how the system models their preferences.

8.4.3.3 Length of explanations

Table 8.5 presents the comparison of the length of explanations. N-method, S-method, and H-method are able to find the shortest counterfactual explanations (only one contextual condition changed). This in turn shows that users' contextual situations can greatly influence their preferences. However, in R-method, since the length of candidates is randomly selected, the final length of counterfactual explanations becomes larger.

8.4.3.4 Instance-level vs list-level

Comparing list-level explanations and instance-level explanations, it can be observed that instance-level explanations usually require less time for searching, however, less percentage of recommendations can be explained. Therefore, the advantage of instance-level explanations lies in efficiency. The advantage of list-level explanations is in success rate. Another potential advantage of list-level explanations is that the differences between $\{i_1, i_2, i_3, \ldots, i_{max}\}$ and $\{i'_1, i'_2, i'_3, \ldots, i'_{max}\}$ is larger, this may help users better identify the reason why items are recommended to them. Which one is preferred by users, instance-level or list-level? We leave this for future work.

Table 8.5: The average length of counterfactual explanations returned (number of contextual conditions).

	Instance-level					List-level			
	Top-1	Top-2	Top-3	Top-4	Top-5	100%	90%	70%	50%
S-method	1	1	1	1	1	1	1	1	1
N-method	1	1	1	1	1	1	1	1	1
R-method	1.91	1.80	1.82	1.73	1.69	2.13	1.95	1.99	1.77
H-method	1	1	1	1	1	1	1	1	1

8.4.3.5 Parameter comparison

In list-level explanations, as the threshold decreases, less time is needed for searching for counterfactual explanations. At the same time, counterfactual explanations are less strictly defined, and fewer changes are needed to be made to the original contextual situation. In instance-level explanations, the more the recommended item is at the back of the list the less time is needed to search for counterfactual explanations. Recommended items at the back of the recommendation lists have lower scores, which means that they are less adapted to users' contextual situations. Less time is needed to find counterfactual explanations.

In this section, we presented the results of our experiments. We show that our S-method is able to identify counterfactual explanations in two levels: instance-level and list-level. As we show in Section 8.2.2, the explanations generated are contrastive and are selected, which satisfies the desired properties defined by Miller (2019). In most cases, our method is able to explain more than 95% recommendations, showing high success rate. Besides, the explanations generated by our method are faithful to the original recommender model by identifying the key contextual conditions. Another advantage is that these explanations are adapted to users' contextual situations, which could be more convincing.

8.5 Conclusions

In this chapter, we introduce an instantiation of GFGE, as outlined in Algorithm 2, within a post-hoc context. We employ SHAP (SHapley Additive exPlanations) (Lundberg and Lee, 2017) to determine the individual contributions of each feature. However, instead of using the SHAP outputs as direct explanations, we apply counterfactual reasoning to validate the important features returned by SHAP. Specifically, we harness them to identify potential candidates for counterfactual explanations. Our approach enables the generation of counterfactual explanations that are both faithful to the original model and easily comprehensible. Our experimental results indicate that, under certain conditions, SHAP can expedite the process of locating suitable counterfactual explanations. Additionally, we propose a two-level explanation structure: list-level explanations, which explain a recommendation list as a whole, and instance-level explanations, focusing solely on a single instance within a recommendation list. Notably, while list-level explanations can cover a broader range of recommendations compared to instance-level explanations, this advantage comes at the cost of increased search time.

Several improvements regarding SECE can be envisaged. The present study exclusively focuses on categorical features. It would also be interesting to extend SECE to cases where numerical

features are applied. Comparing the *instance-level* and *list-level* explanations is another direction that is worth more exploration. Furthermore, applying SECE to explain other black-box models also raises interesting prospects.

Chapters 6 and 7 introduce Algorithm 2 in the model-intrinsic setting while this chapter focuses on the post-hoc setting. In the next chapter, we will compare the three instantiations. Furthermore, we are aware that evaluating the qualities of explanations (as we depicted in Definition 4.7) in RSs is as important as generating explanations. In the next chapter, we will present some primary results of a user study that compares different types of explanations.

Chapter 9

Comparison and evaluation of explanation methods

Contents

9.1 Comparison 148
9.1.1 Model-intrinsic vs Post-hoc 148
9.1.2 Local vs Global 150
9.1.3 Model-specific vs Model-agnostic 151
9.2 Evaluation of explanations 151
9.2.1 Steps and experiment setup 152
9.2.2 Results of user study 154
9.2.3 Discussions 158
9.3 Conclusions 159

In Chapters 6 and 7, we have instantiated the *General Framework for Generating Explanations* (GFGE) in Section 4.3.4 by introducing **A**ware **A**rgumentative **R**ecommender (A^3R) (Zhong and Negre, 2022a) and **C**ontext-**A**ware **F**eature **A**ttribution **T**hrough **A**rgumentation (CA-FATA) (Zhong and Negre, 2023) respectively. Both A^3R and CA-FATA are decomposable (Lipton, 2018), and each step is endowed with explicit semantics. Therefore, model-intrinsic explanations can be generated. In Chapter 8, we have introduced **S**hap-**E**nhanced **C**ounterfactual **E**xplanation (SECE), an instantiation of GFGE depicted in Algorithm 2 in the post-hoc setting. In this chapter, we will compare the three instantiations. Considering that explanations involve knowledge transfer through interactions, we have conducted a primary user study that evaluates the qualities of explanations in recommender systems (RSs) (Zhong and Negre, 2022b).

The following of this chapter is organized as follows, Section 9.1 compares the three instantiations, Section 9.2 presents the results of a primary user study that evaluates the qualities of explanations in RSs, Section 9.3 ends this chapter and bridges to the concluding part of this thesis.

The publications related to this chapter are:

- Zhong, J. and Negre, E. (2022). Context-aware explanations in recommender systems. In International Conference on Deep Learning, Artificial Intelligence and Robotics, pages 76–85. Springer.

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

9.1 Comparison

In Chapters 6, 7 and 8, we have specified the components of each instantiation. To facilitate the comparison between them, we will now show the whole instantiation of each method. Algorithm 5 shows how A^3R (in Chapter 6) instantiates Algorithm 2. In Algorithm 5, the model $\mathcal{M}$ is A^3R; the data $\mathcal{D}$ is the user-item interactions; $\mathcal{P}$ does data pre-processing; $InterpretData$ becomes $QueryFunction1$ that returns user's rating toward item and item's features; $InterpretModel$ becomes an argumentation procedure that returns a Tripolar Argumentation Framework (TAF) tailored to the user-item interaction; $InterpretOutput$ becomes $QueryFunction2$ that returns the predicted rating; $PostHocInterpretation$ is empty because there is no need for conducting post-hoc interpretation; $Aggregation$ can be flexible, which depends on the application scenarios (see Section 6.4.2). CA-FATA (in Chapter 7) is an extension of A^3R, therefore, it instantiates Algorithm 2 in a quite similar way, the difference is that the input data $\mathcal{D}$ also contains users' contexts.

Algorithm 6 depicts the idea of SECE (in Chapter 8), which instantiates Algorithm 2 in a post-hoc setting. In Algorithm 6, $\mathcal{M}$ can be any model; the data $\mathcal{D}$ is the user-item interactions; $\mathcal{P}$ can be designed adapted to $\mathcal{M}$; $InterpretData$, $InterpretModel$ and $InterpretOutput$ all becomes query functions that return users' ratings towards items; $PostHocInterpretation$ becomes SHAP (Lundberg and Lee, 2017) since Algorithm 6 is under the post-hoc explanation setting ($\mathcal{M}$ is regarded as black-box and thus not interpretable.); $Aggregation$ becomes Algorithm 4, it returns the important features (given by SHAP) validated by counterfactual reasoning.

In Chapter 2, we have introduced and compared existing explanation methods in the context of AI from the following aspects: model-intrinsic vs. post-hoc; local vs. global; model-specific vs. model-agnostic. Following the same spirit, we will now compare the three instantiations. Table 9.1 depicts a high-level comparison of the three instantiations.

Table 9.1: Comparison of A^3R, CA-FATA and SECE

	Model-intrinsic vs Post-hoc	Local vs Global	Model-specific vs Model-agnostic
A^3R	Model-intrinsic	Local and Global	Model-specific
CA-FATA	Model-intrinsic	Local and Global	Model-specific
SECE	Post-hoc	Local	Model-agnostic

9.1.1 Model-intrinsic vs Post-hoc

Algorithm 5 is a model-intrinsic instantiation of Algorithm 2. A^3R approaches explainable recommendations by reframing it as a feature attribution problem, a paradigm that seeks to quantify the contribution of individual features towards a given prediction. A unique aspect of A^3R is its conceptualization of feature attribution as an argumentation process. In this view, features assume the roles of arguments that can either support, attack or remain neutral towards a prediction. This vision of feature attribution allows for a more nuanced understanding of how different arguments (features of items) within the model influence the final outcome. This argumentation process effectively maps the prediction process into a serie of explicit, semantically meaningful steps, reflecting

Algorithm 5: An instantiation of Algorithm 2 by A^3R

Input : $\mathcal{M}$: A^3R
$\mathcal{D}$: (u, i)
$\mathcal{P}$: data processing
$InterpretData$: $QueryFunction1$
$InterpretModel$: argumentation procedure
$InterpretOutput$: $QueryFunction2$
$PostHocInterpretation$: $\emptyset$
$Aggregation$: a function that aggregates information from $QueryFunction1$, $QueryFunction2$ and interaction-tailored TAF

Output: Explanations for the output returned by the model given $\mathcal{D}$

$\mathcal{D} \leftarrow \mathcal{P}(D)$
$output \leftarrow \mathcal{M}(\mathcal{D}) \leftarrow \hat{r}_{(u,i)}$
$InfoData \leftarrow InterpretData(\mathcal{D}) \leftarrow r_{(u,i)}, at_i$
$InfoModel \leftarrow InterpretModel(A^3R) \leftarrow TAF$
$InfoOutput \leftarrow InterpretOutput(output) \leftarrow \hat{r}_{(u,i)}$
$Explanation \leftarrow Aggregation(InfoData, InfoModel, InfoOutput)$
Return $Explanation$

the decomposability property of A^3R as outlined by Lipton (2018) (see Section 2.5). In this context, decomposability refers to the capability of breaking down the contributions of individual features, making the model's internal decision-making process transparent: $InfoModel(A^3R)$ returns an interaction-tailored TAF. Additionally, the strength function of arguments within A^3R has been carefully designed to adhere to the principles of *weak balance* (Rago et al., 2018) and *weak monotonicity* (Rago et al., 2021)—the proofs of which are presented in Section 6.4.1.2. These properties show how arguments (features) influence the recommendation, allowing different application scenarios (see Section 6.4.2). Notably, the explanations generated follow the internal mechanism of A^3R. For a comprehensive examination of A^3R's application to explanation generation (the Aggregation function), please see Sections 6.4.2 and 6.5.4. Note that the above analyses also apply to CA-FATA since it is an extension of A^3R.

On the contrary, SECE instantiates Algorithm 2 in a post-hoc setting. This algorithm, too, can be viewed as a feature attribution method. Within the SECE framework, $InfoModel(\mathcal{M})$ solely provides the outcome for a given instance to be elucidated, leaving the internal mechanics **obscured**. The output of $InfoModel(\mathcal{M})$ is utilized to construct a surrogate model using SHAP (Lundberg and Lee, 2017). This surrogate model helps quantify the marginal contribution of each feature, thereby identifying critical features that shape a prediction. The importance and influence of these features are further validated through counterfactual reasoning. Put differently, Algorithm 6 generates counterfactual explanations under the guidance of SHAP. We have demonstrated that the explanations produced are selected and contrastive in Section 8.2.2, which are identified as desired properties of explanations according to Miller (2019) (refer to Section 8.2.2).

In sum, A^3R, CA-FATA, and SECE all fall into the broader spectrum of feature attribution methods. Notably, A^3R and CA-FATA exhibit intrinsic interpretability, offering transparent insights into their workings. In contrast, SECE leverages a post-hoc strategy to provide explanations for recommendations, thereby retaining the underlying model's opacity while still offering informa-

Algorithm 6: An instantiation of Algorithm 2 by SECE.

Input : $\mathcal{M}$: Any model
$\mathcal{D}$: (u, i) or a set of (u, i)
$\mathcal{P}$: data processing
$InterpretData$: $QueryFunction5$
$InterpretModel$: $\mathcal{M}$
$InterpretOutput$: $QueryFunction6$
$PostHocInterpretation$: SHAP
$Aggregation$: SECE

Output: Explanations for the output returned by the model given $\mathcal{D}$

$\mathcal{D} \leftarrow \mathcal{P}(D)$
$output \leftarrow \mathcal{M}(\mathcal{D}) \leftarrow \hat{r}_{(u,i)}$
$InfoData \leftarrow InterpretData(\mathcal{D}) \leftarrow r_{(u,i)}$
$InfoModel \leftarrow InterpretModel(\mathcal{M}) \leftarrow r_{(u,i)}$
$InfoOutput \leftarrow InterpretOutput(output) \leftarrow \hat{r}_{(u,i)}$
$InfoPostHoc \leftarrow SHAP(M) \leftarrow ImportantFeatures$
$Explanation \leftarrow Aggregation(InfoData, InfoModel, InfoOutput, ImportantFeatures)$
Return $Explanation$

tive and useful explanations.

9.1.2 Local vs Global

Another noteworthy distinction between Algorithm 5 and Algorithm 6 lies in the nature of their input data, denoted as $\mathcal{D}$. Significantly, for Algorithm 5, $\mathcal{D}$ could be a single instance (a specific user-item interaction), or a collection of such instances. Operating within the A^3R framework, each prediction can be explained, which corresponds to localized, or instance-based, explanations. The three specific applications discussed in Section 6.4.2 (also in Section 7.3.2) - toy templates, interactive explanations, and contrastive explanations - all adhere to this principle, each shedding light on individual instances or predictions. In addition, Sections 6.5.4 and 7.4.4 provide a global perspective on the model's behavior. Here, users are depicted as vectors, each composed of the importance weights of various feature types[33]. These vectors are subsequently subjected to a clustering process, and the resulting clusters are visualized (as seen in Figures 6.7 and 7.3). An underlying assumption of A^3R is that users' preferences can significantly differ across various types of features. The clustering patterns shown in Figures 6.7 and 7.3 confirm that A^3R can effectively capture these distinctions, thus facilitating a more precise prediction of user ratings (Zhong and Negre, 2022a). Therefore, A^3R and CA-FATA combine local and global techniques, providing both instance-level insights and an overall understanding of the model's operation.

Within the framework depicted in Algorithm 6, the input data $\mathcal{D}$ is strictly a single instance, denoting a specific user-item interaction. This individual-oriented approach begins with the application of SHAP to identify important features, followed by validation of these features through counterfactual reasoning. In essence, these important features serve as informed launching points

[33]Within the CA-FATA framework, users are represented by vectors which quantify the importance of different contextual factors.

for the subsequent generation of counterfactual explanations. Consequently, SECE's methodology provides an in-depth investigation into individual instances. Unlike the global perspective offered by A^3R, SECE's focus remains strictly localized, honing in on specific instances rather than an aggregate view of the user population.

In sum, A^3R and CA-FATA can provide both local and global insights into the model while SECE focuses on the local level.

9.1.3 Model-specific vs Model-agnostic

It is important to highlight that Algorithm 5 is strictly applicable to RSs that utilize item features. For instance, in sequential recommendation scenarios (Wang et al., 2019c), both A^3R and CA-FATA fall short in applicability[34]. Consequently, Algorithm 5 is characterized as a model-specific explanation technique.

In contrast, the framework of Algorithm 6 does not impose any assumptions on the recommender model—$\mathcal{M}$ can represent any model. This versatility stems from two fundamental aspects: (1) the initial step of applying SHAP is to identify important features. As a model-agnostic method, SHAP can accommodate any model type; (2) the application of counterfactual reasoning to validate these critical features, again not mandating any model-specific assumptions. One significant advantage of being model-agnostic is the broad applicability—SECE can integrate with any model type, thereby offering enhanced flexibility (Molnar, 2020).

In summary, while A^3R and CA-FATA are limited to models that incorporate item features, SECE is characterized by its model-agnostic property, offering a more flexible approach to explanation generation in diverse model contexts.

Algorithm 5 and Algorithm 6, while sharing a common goal of attributing feature importance, represent different paradigms of explanation generation. Each has its strengths and unique capabilities, catering to different scenarios and recommendation models. Algorithm 5 can provide model-intrinsic explanations while Algorithm 6 provides post-hoc explanations. Another advantage of Algorithm 5 over Algorithm 6 is that Algorithm 5 can handle single instances or multiple instances, allowing it to offer both localized, instance-based explanations, and a global understanding of the model's behavior (Zhong and Negre, 2022a). In contrast, Algorithm 6 deals strictly with individual instances, maintaining a localized focus that delves into specific case studies without offering a holistic view. Despite this, Algorithm 6 has its own advantage of being model-agnostic. It does not impose any constraints on the model type, thanks to the flexibility of SHAP and counterfactual reasoning. This broad applicability allows for greater flexibility, accommodating diverse model contexts (Molnar, 2020). Therefore, the choice between these two algorithms ultimately depends on the specific context and requirements of the recommendation model in question.

9.2 Evaluation of explanations

We recognize that given the inherent human aspect involved in explanations, human involvement is paramount when comparing such explanations. As summarized in Section 3.7, explanations within the realm of RSs aim to enhance various aspects including transparency, effectiveness, efficiency,

[34]For example, we have not expanded the concepts of A^3R and CA-FATA to accommodate sequential recommendations.

persuasiveness, trust, satisfaction, and scrutability. These goals often serve as the metric for assessing the quality of explanations in RSs (Tintarev and Masthoff, 2015, Gedikli et al., 2014). In the case of A^3R, recommendations are explained through argumentation, while CA-FATA enhances this approach by considering user contexts. In the case study of SECE, we demonstrate its capability to generate context-aware explanations.

To facilitate an empirical comparison of explanations with and without contextual information, we developed a web-based questionnaire (Zhong and Negre, 2022b). This user study aims to evaluate the impact of these explanations on transparency, effectiveness, efficiency, persuasiveness, trust, and satisfaction. Each of these metrics is detailed in Section 3.7. The motivation for comparing context-aware explanations and context-free explanations lies in two primary considerations: (1) argumentative explanations have been empirically shown to enhance users' trust and acceptance (Naveed et al., 2018); (2) the inclusion of additional explanation types necessitates a larger participant pool and more time commitment from participants to complete the questionnaire.

The design and execution of the user study, along with the experimental setup, are presented in detail in Section 9.2.1. The collected data and the corresponding analysis are reported in Section 9.2.2.

9.2.1 Steps and experiment setup

We adopt the widely used contextual movie recommendation dataset CoMoDa (Košir et al., 2011), where there are 121 users, 1197 movies, 2296 ratings and 12 contextual factors. In the CoMoDa dataset, the average number of ratings of each user is about 12. In order to be consistent with the original dataset, participants are required to give their ratings for 12 movies under different contextual situations. In the original dataset, there are 12 contextual factors, some contextual factors could have a more important impact than others (Ferdousi, 2020). We wish to select the most impacting contextual factors to minimize the efforts of participants to specify his/her contextual situation. According to Ferdousi (2020), in the CoMoDa dataset, the 6 most impacting contextual factors are *(1) whether users have watched a movie before, (2) users are given a decision or decide themselves, (3) users' physical wellness, (4) users' moods, (5) users' location and (6) weather*. Note that in step 2 of our protocol, we recommend movies that participants have not interacted with in step 1 of our protocol, we do not consider the contextual factor *Interaction*. Since all the participants are given a movie, we do not consider the contextual factor *Decision* either. Therefore, in our case, we only consider the following 4 contextual factors: *Physical wellness*, *Mood*, *Location* and *Weather*.

9.2.1.1 Major steps

There are mainly four steps in the user study[35]: (1) participants are asked to rate 12 randomly selected movies given several contextual situations by applying the data set (Košir et al., 2011). This step is to learn users' preferences under different contextual situations; (2) the system generates recommendations for the participants under a randomly generated contextual situation, for more details we refer to Appendix A; (3) participants are presented with explanations but without detailed information of movies and they are asked to rate the movie, noted as r, the time consumed for giving the rating is noted as t, users are then presented with detailed information of the same

[35]Here is a demo that shows the procedure of user study: https://drive.google.com/drive/folders/1cYiJ8gLPYlMZu4k41gIBqnE2HytbK3KL?usp=sharing

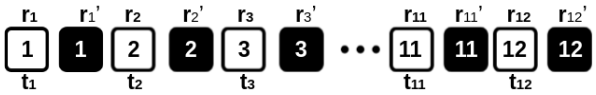


Figure 9.1: The number indicates the movie recommended. White indicates that only an explanation is displayed, black indicates that detailed information about the movie is displayed, r and r' are the ratings given by participants, and t is the time cost by participants to give a rating.

movie and rate them again, noted as $r^{'}$, Figure 9.1 illustrates this process; (4) participants are asked to evaluate the explanations in terms of transparency, effectiveness, efficiency, persuasiveness, trust, and satisfaction by a 1 (lowest) to 5 (highest) rating scale. In all, this user study follows a within-subject design (Keren, 2014) to get more data from participants. Therefore, the six metrics can be quantified as below: (1) efficiency: the time used by participants for giving a rating when only presented with explanations; (2) effectiveness: the difference between r and $r^{'}$, if the difference is near 0, it indicates good effectiveness since explanation has helped participants perfectly estimate the rating of recommendation; (3) persuasiveness: $r-r^{'} > 0$ means positive persuasiveness; $r-r^{'} < 0$ means negative persuasiveness; (4) transparency, satisfaction, and trust: they are reflected by users' responses in step 4; (5) in terms of efficiency, effectiveness, and persuasiveness, we will compare the measured effects and human-perceived effects.

The adopted protocol for this user study is largely derived from Bilgic and Mooney (2005). This protocol was also used by Gedikli et al. (2014) to study the influence of different explanation styles in terms of efficiency, effectiveness, persuasiveness, transparency, and satisfaction. Our protocol varies from Gedikli et al. (2014) in two significant ways:

- We assess the effects of explanations in terms of trust, which was not considered by Gedikli et al. (2014).
- For efficiency, effectiveness, and persuasiveness, we ask users to evaluate different styles of explanations concerning these three metrics. This is to identify any discrepancies between the human-perceived experiences and actual effects. In contrast, Gedikli et al. (2014) solely utilized objective methods to measure these three metrics.

9.2.1.2 Baselines

Concretely, we wish to compare the effects of explanations with and without contextual information. Table 9.2 presents the explanation types we plan to show to participants in the experiment. The *Avg* style explanation shows users the average rating of the recommended item; *Per* style explanation presents the distribution of ratings of the recommended item; *Simu* style explanation presents the average ratings of similar users. *Simu*, *Per*, and *Avg* style explanations are among the 21 explanation styles that Herlocker et al. (2000) compared in terms of persuasiveness. Since explanations that perform well in one goal may perform poorly in another (trade-off among goals) (Tintarev and Masthoff, 2015) and we investigate six different goals, we do not simply select the top-ranking explanation types proposed by Herlocker et al. (2000). We selected the top, middle, and end of the ranking list of the 21 explanation styles, namely *Simu*, *Per*, and *Avg* respectively. *Simi* style explanation simply says the recommended item is similar to the items a participant has interacted with before, which is popular in websites such as Amazon (Sarwar et al., 2001); *Content* style

explanation presents the characteristics of recommended items that may be preferred by users, it is widely adopted for the movie, book recommendation (Vig et al., 2009)[36]. *Context-aware* explanation leverages users' contextual information to explain recommendations (Sato et al., 2018). In order that these explanation styles can be compared fairly, they are all in textual forms.

Table 9.2: Explanations to be presented.

Name	Example
Avg	The average rating of this movie is 3.8
Per	80 percent of users rate this movie more than 4
Simu	The average rating of users whose preferences are similar to yours is 4.1
Simi	This movie is similar to movies you watched before
Content	This is a movie directed (acted) by
Context-aware	**The system suppose that you would like to watch this movie when it shines, healthy you are at home and in good moods**

9.2.2 Results of user study

Given the web-based nature of our questionnaire, we were able to distribute it to researchers both within our laboratory and externally, as well as to students at our university via mailing lists. The questionnaire was available online for a week, allowing participants to respond anonymously. In total, we received 147 responses.

With respect to demographic breakdown, the genders were evenly distributed with almost half of the respondents identifying as male and the other half as female. The majority of participants fell within the age range of 18 to 55 years; over 66% of them had achieved an education level of Bac+5 (equivalent to a master's degree level) or higher. Close to half of the participants were students (ranging from undergraduate to doctoral candidates), while the remainder were either employees of companies or involved in higher education. This distribution is largely due to the fact that our mailing lists primarily consisted of researchers and students. Over half of the participants reported watching movies regularly, indicating a high level of interest in this medium.

In order to analyze the observed differences between the different explanation types, we first applied analysis of variance (Tabachnick and Fidell, 2007). Upon identifying significant differences, we proceeded to employ the Tukey HSD test (Abdi and Williams, 2010) to pinpoint the specific discrepancies. All tests were conducted at a 95% confidence level.

9.2.2.1 Objective evaluation

In this section, we will present the results of objective evaluation: efficiency, effectiveness, and persuasiveness. Table 9.3 shows the results of these three metrics.

Observations indicate that participants presented with Context-aware explanations tend to take more time to make a decision, with decision-making in our context meaning the assignment of a score (See row Context-aware and column Efficiency (seconds)). While there were no significant differences among the explanation types in terms of $r - r'$, both Content style and Context-aware

[36] In fact, the content-based explanation also adheres to the argumentative spirit, closely resembling the toy template in Table 6.2

Table 9.3: Efficiency, effectiveness, and persuasiveness of different types of explanations (the average values). Efficiency is reflected by the time consumed by participants to give a rating (in seconds); effectiveness is reflected by the difference between r and $r^{'}$ (as presented in Figure 1), the closer it is to 0 the more effective an explanation is; persuasiveness is also measured by the difference between r and $r^{'}$, $r - r^{'} > 0$ means positive persuasiveness; $r - r^{'} < 0$ means negative persuasiveness. Note that $r \in [1, 5]$, $r' \in [1, 5]$, therefore, $r - r^{'} \in [-4, 4]$.

	Efficiency (seconds)	Effectiveness and persuasiveness
Avg	4.78	0.30
Content	5.37	**-0.02**
Context-aware	**8.25**	**-0.01**
Per	5.17	0.11
Simi	5.99	-0.18
Simu	6.89	0.10

Table 9.4: Subjective evaluation of efficiency, effectiveness, persuasiveness, satisfaction, trust, and transparency. Note that these are the results of the Likert-scale questionnaire, 1-5; the values in the table are the average of the participants.

	Efficiency	Effectiveness	Persuasiveness	Satisfaction	Trust	Transparency
Avg	3.16	2.98	3.01	3.00	2.85	3.26
Content	3.25	3.12	3.20	3.19	3.28	**3.46**
Context-aware	2.96	3.07	3.24	3.21	3.14	3.26
Per	3.32	3.26	3.18	3.11	3.04	3.12
Simi	**3.72**	**3.47**	**3.52**	**3.50**	**3.53**	3.42
Simu	3.49	3.33	3.30	3.36	3.30	3.14

explanations slightly outperformed others in helping participants estimate the quality of recommendations, as the difference between r and $r^{'}$ was closest to zero for these styles. Conversely, Avg and Simi explanation types led users to underestimate the quality of recommendations. The Per and Simu styles were found to be marginally more persuasive than the other styles, suggesting that the presentation of ratings from similar users can enhance persuasiveness.

9.2.2.2 Subjective evaluation

We now turn to the results of the subjective evaluation, as assessed via a Likert-scale questionnaire. It is important to note that the factors of efficiency, effectiveness, and persuasiveness were also evaluated by directly soliciting feedback from participants. Table 9.4 displays the results for participant-reported efficiency, effectiveness, persuasiveness, satisfaction, trust, and transparency. In the subjective evaluation, Simi style explanations were perceived as the most efficient, effective, and persuasive. However, this was not mirrored in the objective evaluation (see Table 9.3). In the objective assessment, Avg style explanations were found to be more efficient and persuasive, while Context-aware and Content style explanations proved to be more effective. These results highlight a discrepancy between the actual effects of the explanations and the participants' perceptions. The reasons behind this divergence and strategies to minimize it warrant further investigation.

Satisfaction, trust, and transparency are evaluated by directly asking participants. The last

Table 9.5: Spearman Rank Order Correlation of metrics in *Context-aware* explanations. The correlation is between −1 and +1, large positive values indicate that two metrics are positively correlated, and vice versa.

	Efficiency	Effectiveness	Persuasiveness	Satisfaction	Trust	Transparency
Efficiency	1.0***	0.45***	0.05	-0.17	-0.15	0.01
Effectiveness	0.45***	1.0***	0.01	-0.08	-0.0	-0.1
Persuasiveness	0.05	0.01	1.0***	0.04	-0.03	-0.17
Satisfaction	-0.17	-0.08	0.04	1.0***	0.82***	-0.04
Trust	-0.15	-0.0	-0.03	0.82***	1.0***	-0.11
Transparency	0.01	-0.1	-0.17	-0.04	-0.11	1.0***

three columns in Table 9.4 contain the results of participants-reported satisfaction, trust, and transparency. *Simi* style explanations achieve the highest level of satisfaction and trust, which shows that participants prefer explanations that compare the present recommendation and the items that they consumed before. *Content* style explanations provide information about recommendations and help participants understand how the system works. As a result, *Content* style explanations are perceived to be transparent.

Contrary to our hypothesis, context-aware explanations were not perceived as more satisfying or transparent. A potential reason for this could be related to the nature of our user study. Since the contextual situations were randomly generated, they may not have accurately reflected the real situations of the participants when they were responding to the questionnaire. We asked participants to envision themselves in these artificially-created scenarios, which is a non-trivial task. This could also account for why participants took longer to issue a rating when presented with context-aware explanations, as seen in Table 9.3.

9.2.2.3 Correlation between different metrics

Table 9.5 presents the *Spearman Rank Order Correlation* of metrics for evaluating *Context-aware* explanations, it can be observed that these metrics are not independent. For example, in the *Context-aware* explanation, trust and satisfaction, efficiency and effectiveness are positively correlated. However, transparency and persuasiveness, trust and efficiency are negatively correlated, this leads to the conclusion that it is hard to design explanations that can perform well in all aspects. In order to comprehensively evaluate the qualities of explanations, more metrics should be considered.

9.2.2.4 Aggregating metrics

The values in Table 9.4 are human perceived, human perception is always fuzzy and the evaluator's opinion by nature comes in linguistic form (Zadeh, 2013), which is the reason why we have specified the meaning of each score when participants respond to the Likert-scale questionnaire, see Figure 9.2. Therefore, the score in our questionnaire can be linked to categorical appraisal grades: "Very poor", "Poor", "Medium", "Good" and "Very good". On the other hand, the metrics are somehow correlated as indicated in Table 9.5, aggregating these metrics by applying the average operation is not realistic. To get a more general evaluation of explanations, we are inspired by the fuzzy synthetic evaluation (Lan* et al., 2005) that provides a fuzzy mapping between each of

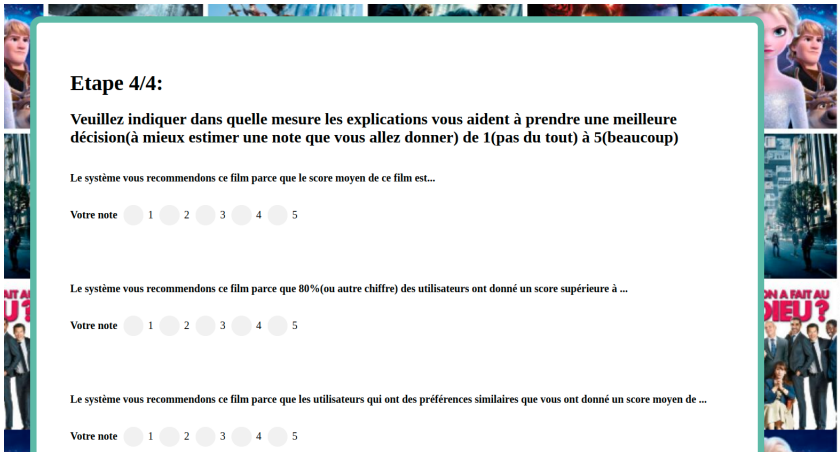


Figure 9.2: A screenshot of the Likert-scale questionnaire when evaluating the effectiveness of explanations. Participants are indicated that 1 means they think the explanation does not help them make good decision at all while 5 indicates that the explanation help them make good decisions.

the evaluation factors (e.g. efficiency, effectiveness, persuasiveness, trust, transparency, and satisfaction in our case) to a set of categorical appraisal grades ("Very poor", "Poor", "Medium", "Good", "Very good" in our case.). For instance, a score such as 3 on a 5-point scale could be classified under both "Medium", "Good" grades. The degree to which the score aligns with each grade can vary, embodying different membership degrees to each grade. This variation depends on the weights attributed to each evaluation factor and the average score given by participants. Taking an example, one might discern that a score of 3.2 for efficiency could correspond to the fuzzy sets "Medium", "Good" and "Very Good", with respective membership degrees of 0.7, 0.2, and 0.1. Assigning a degree of membership to multiple "fuzzy grades" allows for capturing and preserving more of the inherent uncertainties involved when responding to the questionnaire. We follow the steps given by Zhou and Chan (2017)[37]:

- **Step 1, determining the set of evaluation factors:** evaluation factors can be determined based on the goals of the product evaluation process: $\mathcal{F} = \{f_1, f_2, \ldots, f_s\}$ indicates a set of s factors. In our case, $\mathcal{F} = \{efficiency, effectiveness, persuasiveness, trust, transparency, satisfaction\}$.

- **Step 2, determining the set of appraisal grades:** this set defines the levels of appraisal grades, e.g. $\mathcal{V} = \{v_1, v_2, \ldots, v_p\}$ defines p levels of appraisal grades. In accordance with the linguistic description in the questionnaire, the appraisal grades set is defined as $\mathcal{V} = \{Verypoor, Poor, Medium, Good, Verygood\}$.

- **Step 3, setting the fuzzy mapping matrix:** the aim of the evaluation process is to establish a map from set $\mathcal{F}$ to set $\mathcal{V}$. For a specific factor f_i the fuzzy mapping to the appraisal vector $\mathcal{V}$ can be represented by the vector $r_i = (r_{i1}, r_{i2}, \ldots, r_{ik}, \ldots, r_{ip})$ where p denotes the number of levels of the appraisal grades in **Step 2**, r_{ik} denotes the fuzzy membership degree of evaluation factor i to grade k. Using the example of efficiency above, $r_1 = (0, 0, 0.7, 0.2, 0.1)$,

[37] In the paper, the authors propose a fuzzy comprehensive evaluation method to determine product usability.

then the measurement on the evaluation factor "efficiency" has a fuzzy membership of 0.7 in the grade "Medium", a fuzzy membership of 0.2 in the grade "Good" and a fuzzy membership of 0.1 in the grade "Verygood", respectively. Therefore, the size of the fuzzy mapping matrix $\mathcal{R}$ is $s \times p$, in our case, the size becomes 6×5. Intuitively, we set the r_{ik} as the proportion of participants that have given the score corresponding to the appraisal grade level k in terms of evaluation factor i. Still in the example of efficiency, if 7 out of 10 participants have given a score of 3 then $r_{13} = 0.7$.

- **Step 4, determining the weight of each evaluation factor:** for a thorough assessment, it's imperative to quantify the relative weight of each evaluation factor on the overall quality. This is particularly relevant as different stakeholders place varying levels of importance on these factors. For instance, service providers, such as online sellers, may place a higher emphasis on the persuasiveness of explanations, whereas users might prioritize the effectiveness of explanations. To this end, the weights can be represented by a vector: $\mathcal{W} = (w_1, w_2, \ldots, w_s)$, where $\sum_{i=1}^{i=s} w_i = 1$. To the best of our knowledge, this is the first work that applies fuzzy mapping matrix to aggregate different metrics for evaluating the qualities of explanations. We empirically set the weight of the six metrics the same: $\mathcal{W} = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$[38].

- **Step 5, getting the overall appraisal result:** the final assessment outcome can be computed by considering the relative weights of each evaluation factor. As a result, a singular vector corresponding to the overall evaluation can be expressed as $\mathcal{E} = (e_1, e_2, \ldots, e_j, \ldots, e_p) = \mathcal{W} \circ \mathcal{R}$, where $\circ$ is the composition operator. There are typically five types of composition operators (Zimmermann, 2011, Lan* et al., 2005), we have chosen the sum operator: $e_j = \sum_{i=1}^{i=p} w_i * r_{ij}$. We leave the comparison of the composition operations for future work.

Following the outlined steps, the comprehensive evaluation of the six types of explanations is summarized in Table 9.6. Among the context-free types, the *Simi*, *Content*, *Per*, and *Simu* classifications fall within the "Good" level, with the *Simi* type having the highest fuzzy membership. The *Avg* type is situated within the "Medium" category. These findings align with Table 9.4, where the *Simi* type scores higher than the other types, barring transparency. Interestingly, despite its fuzzy membership standing at just 0.2857, the *Context-aware* type falls within the "Very Good" level. We infer that participant evaluations of the *Context-aware* type could be polarized, given the considerable fuzzy membership for "Very Poor". This discrepancy could arise if the randomly generated contextual situation aligns closely with the participant's real-life situation, leading them to rate this explanation type more favorably.

9.2.3 Discussions

This primary study presents several limits: first, our context-aware explanations were rather simplistic, merely presenting contextual information. They could, in fact, be combined with item features, as we demonstrate in Table 7.2. Second, the contextual situations were randomly generated, therefore, the contextual information might not always align with the participants' real-life

[38] Note that the weight could be set differently according to the main goals of explanations. For example, explanations designed to persuade users to buy or to consume, the weight of persuasiveness can be set higher.

Table 9.6: Overall evaluation of explanation types. Note that, for each type, the largest fuzzy membership is in bold.

	Overall evaluation
Avg	(0.1054,0.1530,**0.3979**,0.2176,0.1258)
Content	(0.0544,0.1632,0.3197,**0.3367**,0.1258)
Context-aware	(0.1938,0.1667,0.1530,0.2006,**0.2857**)
Per	(0.0748,0.1598.0.3299,**0.3197**,0.1156)
Simi	(0.0442,0.1292,0.2142,**0.4217**,0.1904)
Simu	(0.0714,0.1292,0.3027,**0.3537**,0.1428)

situations. Lastly, in relation to argumentative explanations, they were only delivered by showcasing item features, comparable to the toy template in Table 6.2. We did not evaluate interactive explanations (as discussed in Section 6.4.2.2), which permit users to inform RSs when recommendations are inappropriate, nor did we evaluate the contrastive explanations covered in Section 6.4.2.3.

Despite these limitations, our user study allows us to derive some valuable insights. First, although context-aware explanations can assist users in better estimating the quality of recommendations, they require more time for users to make a decision. Second, using similar users' ratings to explain recommendations can be persuasive. Third, there may be a discrepancy between the actual effects of explanations and how they are perceived by participants. Lastly, there is an interrelation among the goals of explanations, indicating that it might not be feasible to design explanations that optimize all metrics simultaneously.

9.3 Conclusions

In this chapter, we begin by comparing three instantiations of GFGE, as depicted in Algorithm 2. We consider them from three perspectives: Model-intrinsic vs. Post-hoc; Local vs. Global; Model-specific vs. Model-agnostic. Algorithms 5 (A^3R in Chapter 6 and CA-FATA in Chapter 7) and 6 (SECE in Chapter 8) are both designed to attribute feature importance during explanation generation, but they employ different paradigms with unique strengths to accommodate various scenarios and recommendation models. Specifically, Algorithm 5 provides model-intrinsic explanations and can handle single or multiple instances, thereby offering localized insights on a case-by-case basis and a more global understanding of the model's behavior. In contrast, Algorithm 6 strictly handles individual instances, offering localized insights into particular case studies without a comprehensive perspective. Despite this, its strength lies in its model-agnostic nature, which does not impose constraints on the model type, thus displaying its broad applicability and flexibility. The choice between these two algorithms ultimately depends on the specific context and requirements of the recommendation model under consideration. Following this, we present the results of a preliminary user study where we compare different explanation types in RSs. Despite some limitations, the user study offers several key insights that can provide guidance for conducting future user studies.

Part IV

Concluding remarks

Chapter 10

Conclusions and perspectives

Contents

10.1 Conclusions **163**
10.2 Perspectives for future work **166**
10.2.1 Improvements of our work 166
10.2.2 Relations to other research topics in RSs 167
10.3 List of publications **168**

In this concluding chapter, we aim to recall and summarize the principal research contributions presented in the preceding chapters of this thesis and also consider potential avenues for further exploration and investigation in this field.

10.1 Conclusions

Artificial Intelligence (AI) has seamlessly integrated into daily life, with *Recommender Systems* (RSs), a subset of AI, serving as pivotal decision-support tools in this digital age. Their influence spans across diverse domains such as medicine and finance, instigating an escalating demand to comprehend the rationale underlying recommendations. The emphasis on explainable RSs, offering justified recommendations, has captivated both industrial and academic spheres aiming to enhance efficiency, transparency, and user satisfaction. Despite the advent of numerous explanation methodologies, a limited scope of research has been devoted to the generality of explainable RSs. This thesis, therefore, navigates through the dimensions of generality and explainability in RSs to foster advancements in this realm.

Upon reviewing existing works on explanations in AI, we have identified several frequently used, yet ill-defined notions, especially *interpretability* and *explainability* in the domain of AI (Miller, 2019, Arrieta et al., 2020). The same issue exists in the domain of RSs, which is an important branch of AI. The ill definitions of these notions have made comparing and generalizing existing explanation methods challenging. In this regard, the first research question we aim to address is: **RQ1, "What are the differences between 'interpretability' and 'explainability' in the context of AI, and more specifically in the context of RSs?"** We then clarify **RQ2, "Why 'explainability' and 'interpretability' are remarkably demanded in RSs?"** In sequence, we arrive at the core research question of this thesis: **RQ3, "How to explain recommendations?"** Various

explanation methods have been proposed in the domain of RSs, however, it remains unclear what the best way is to provide explanations under a given context for a given task (Zhou et al., 2021). This leads us to our final research question: **RQ4, "How to evaluate explanations in RSs?**"

Overall, this thesis focuses on the topic of developing a ***General Framework for Generating Explanations*** **(GFGE)**. Our initial step involves **addressing the ambiguity** surrounding commonly used terminologies within the field of AI, notably the terms *interpretability* and *explainability*, which pose significant challenges to the generality of current explanation generation methods in AI. To overcome this, we employ concepts from knowledge management and propose the ***Interpret/Explain Schema*** **(IES)** in the context of AI, which also applies in the domain of RSs. IES allows distinguishing the two frequently used, yet ill-defined terms. With this, we propose GFGE in AI. This framework seamlessly integrates existing methods for generating explanations. The framework is instantiated under two settings: the **model-intrinsic explanation setting** and the **post-hoc explanation setting**. Specifically, under the model-intrinsic explanation setting, we examine two cases: context-free and context-aware. Finally, a primary **user study** was conducted to explore the influence of contextual information on the impact of explanations within RSs.

IES and GFGE: this contribution addresses **RQ1, "What are the differences between 'interpretability' and 'explainability' in the context of AI, and more specifically in the context of RSs?"** To resolve the ambiguity of these poorly defined concepts, we expanded our review beyond AI-related literature, as discussions and research on explanations have a rich history predating the rise of AI. By examining works from philosophy, social science, and cognitive science (Dennett, 1989, Lewis, 1986, Lipton, 1990), we noted that the process of explanation typically encompasses a knowledge transfer. Consequently, we incorporated concepts from knowledge management (Despres and Chauvel, 1999, Arduin et al., 2015): sense-giving and sense-reading (Polanyi, 1967). With this, we have proposed the IES in Section 4.3, Chapter 4, which helps to define and disambiguate "interpretability" and "explainability". Under IES, developers interpret information from data, model and outputs of the model (sense-reading), then model developers endow a sense to this information (sense-giving) and convey it to users. Users receive this information, interpret it and get a meaning, thus understanding the output of the model. System developers, given their role in designing the system, are typically interested in every component of an AI system. They are expected to understand the function and significance of each part. Conversely, plain users are predominantly concerned with understanding why certain results are produced. The *interpretability* of data, which focuses on unraveling the information embedded in the data, caters mainly to developers. Similarly, the *interpretability* of models, addressing the internal workings of models, is primarily targeted at developers. When the internal mechanics of models are unclear, post-hoc *interpretability* becomes crucial, also aimed at developers. The *interpretability* of explanations is mainly directed at users, focusing on the quality of explanations. Therefore, the user's ability to interpret an explanation could reflect the quality of the explanation itself. *Explainability* of model results, chiefly targeting users, addresses the rationale behind specific results. Building on IES, we proposed GFGE (outlined in Algorithm 2, Section 4.3.4) for generating explanations in AI. This framework modularizes the process of generating explanations in AI, which may assist researchers in conceptualizing their explanatory approaches more effectively. The framework successfully integrates existing works related to explanations in AI, including those that develop interpretable models with clear internal mechanisms, as well as those that generate post-hoc explanations without clarifying models' internal mechanisms. Notably, the framework outlined in Algorithm 2 is generic, placing no specific constraints on data preprocessing methods, the results to be explained,

or the explanation method. Rather, these are considered parameters of Algorithm 2, which can be instantiated in various ways to modify how the model's results are explained.

Drives for conducting research related to explanations in AI: in Section 2.4, we categorize these desiderata into three driving forces: commercial drive, regulatory drive, and technical drive (Zhong and Negre, 2021). In Section 3.7, we have further summarized the goals of explanations in RSs: promoting transparency, effectiveness, efficiency, persuasiveness, trust, satisfaction, and scrutability (Tintarev and Masthoff, 2022). Ultimately, these considerations also address the second research question: **RQ2 "Why 'explainability' and 'interpretability' are remarkably demanded in RSs?"**

Model-intrinsic instantiations: this contribution addresses **RQ3.1, "How to build transparent models that can explain recommendations while ensuring the accuracy of recommendations?"** To illustrate the applicability of GFGE, we first instantiate it within a model-intrinsic explanation setting. Consequently, we have proposed **A**ttribute-**A**ware **A**rgumentative **R**ecommender (A^3R) in Chapter 6. Under A^3R, the rating prediction is mapped into Tripolar Argumentation Frameworks (TAFs) tailored to each user-item interaction: items' features are regarded as arguments that may support, attack or neutralize the recommendations of items; users' ratings towards determine the polarity and strengths of arguments. Notably, each step of A^3R has explicit semantics, signifying the decomposable nature of A^3R (Lipton, 2018), or what can be otherwise described as intelligible (Lou et al., 2012). Our experiments on four datasets show that A^3R achieves competitive performances compared with Factorization Machines (Rendle, 2010), and largely improves the prediction accuracy compared with the existing argumentation-based model (Rago et al., 2018, 2021). Recognizing the variability of users' preferences across different contexts, we developed **C**ontext-**A**ware **F**eature **A**ttribution **T**hrough **A**rgumentation (CA-FATA) in Chapter 7, which extends A^3R by leveraging users' contexts. It generates unique user representations under target contextual situations, implying that user ratings towards features are also context-aware. Accordingly, the TAF adapts to users' contexts. Our experiments using two real-world datasets demonstrate that CA-FATA outperforms several strong context-aware baselines that employ neural networks. Similar to A^3R, CA-FATA shows considerable improvement over the existing argumentation-based model. Furthermore, theoretical analyses indicate that the strength function for argumentation under A^3R and CA-FATA satisfies two desirable properties under AFs: *weak balance* (Rago et al., 2018) and *weak monotonicity* (Rago et al., 2021). These properties enable us to identify at least three applications of A^3R and CA-FATA: (1) simplified explanation templates for recommending and not recommending items; (2) interactive explanations allowing RSs to adapt recommendations based on users' real-time feedback; and (3) contrastive explanations that compare recommended and non-recommended items in terms of item features.

Post-hoc explanation instantiation: this contribution addresses **RQ3.2, "How to explain existing complex RSs using post-hoc explanation techniques while mitigating the limitations of such techniques?"** In the realm of post-hoc explanation setting, we instantiate GFGE through the introduction of the **S**hap-**E**nhanced **C**ounterfactual **E**xplanation (SECE) in Chapter 8. SECE validates the important features returned by SHAP via counterfactual reasoning, thus generating counterfactual explanations. SECE can be applied to explain any recommender model since we do not make any assumption on the internal mechanism of recommender models.

A primary user study: recognizing that explanations are embedded within human interactions, as discussed in Section 4.2, we conducted a user study to address "**RQ4, How to evaluate explanations in RSs?**". The study employed metrics identified in Section 3.1.2: promoting transparency, effectiveness, efficiency, persuasiveness, trust and satisfaction (Tintarev and Masthoff, 2022). Despite some limitations, the user study allows us to derive some valuable insights, which helps to conceptualize future user studies.

10.2 Perspectives for future work

The contributions of this thesis also open promising avenues for future works. We will first introduce several possible direct extensions to our work, then we discuss relations with other research topics in the domain of RSs.

10.2.1 Improvements of our work

Some direct improvements or extensions of our work can be envisaged:

On GFGE: in this thesis, our focus is on explanations designed for end-users. Therefore, under the IES depicted in Figure 4.1, model developers interpret information from data, model, and outputs to generate explanations for the users. However, it is worth considering cases where the target audience is not limited to end-users. For example, other stakeholders such as auditors or service providers could replace users, which would still be aligned with the principles of sense-giving and sense-reading (Polanyi, 1967).

On A^3R and CA-FATA: under A^3R and CA-FATA, we have applied the inner product to compute the importance of feature types (Equation 6.1), of contextual factors (Equation 7.1); the inner product is also utilized for computing users' ratings towards features (Equations 6.3 and 7.7). This means that the inner product operation is endowed with explicit semantics: each step is mapped into Argumentation Frameworks (AFs). More complex functions such as Multi-Layer Perceptron (MLP) could be applied, expecting more accurate rating prediction.

Both A^3R and CA-FATA follow the spirit of argumentation: features are regarded as arguments and users' ratings towards these arguments determine the polarity and the strength of them. In fully connected layers, each neuron performs a linear transformation on the input vector by utilizing a weights matrix. Regarding neurons as arguments may help to understand how neural networks achieve a prediction: the operations are mapped into AFs. There have been some primary works that integrate the spirit of argumentation into neural networks (Proietti and Toni, 2023, Albini et al., 2020). It would be interesting to explore how such integration benefits RSs.

Both A^3R and CA-FATA focus on rating prediction, it would also be interesting to extend A^3R and CA-FATA to ranking prediction problems.

On SECE: SECE only deals with RSs that utilize categorical features, one direct improvement would be to extend SECE to deal with numeric features. In Algorithm 4, after investigating the feature considered to be the most important according to SHAP, if the recommendation remains unchanged, we directly explore the first two most important features. It is possible that modifying only the second (or the third) most important leads to a counterfactual explanation. Therefore,

an improvement could be setting a threshold: investigating up to a certain number of features separately, if no counterfactual explanation is found, then starting to modify more than one feature as we do in Algorithm 4.

On evaluating explanations in RSs: Chapter 9 presents the results of a primary user study that compares context-aware and context-free explanations in RSs, which offer some insights for conducting future user studies. In this user study, we have adopted a within-subject design where participants are required to evaluate several types of explanations, this requires a longer time for participants to answer all the questions and may reduce participants' willingness to respond. A between-subject design can be envisaged to reduce the time needed, which will also require a larger pool of participants. Furthermore, it would be interesting to compare more types of explanations such as contrastive explanations and interactive explanations we detailed in Chapter 6.

The aforementioned perspectives represent a short-term view. In the following section, we discuss how our work relates to other contemporary research topics in RSs from a long-term perspective.

10.2.2 Relations to other research topics in RSs

In this thesis, one of our focuses is on the explainability in RSs. In fact, our work also relates to other hot research topics in RSs, including fairness, robustness, and audibility (accountability) in RSs. In the following, we discuss how our work relates to these topics.

Fairness through explainability: fairness in RSs, an emerging field of research, aims to ensure that the system's recommendations do not inadvertently favor specific groups of users, items, or providers due to inherent biases in the data, model, or algorithm (Wang et al., 2023). Explanations can provide users with a clearer understanding of the underlying basis of recommendations, potentially exposing any inherent system biases. Such transparency can bolster user trust and afford users the opportunity to refine their interactions with the system to better suit their preferences. In Section 6.4.2.2, we have shown that A^3R (also CA-FATA) allows users to tell the system when the recommendations are not appropriate: users may indicate that they actually do not like a feature or simply do not care about a feature type (this can be viewed as a potential bias), RS can make corresponding adjustments. This is just an empirical example, we plan to further quantify the fairness offered by A^3R and CA-FATA. On the other hand, SECE does not make any assumptions about the models, however, it helps to identify the important features that determine a recommendation, which may include biased features.

Robustness through explainability: robustness in RSs pertains to the system's resilience in maintaining its performance, reliability, and integrity in the face of adversarial or malicious activities such as data poisoning or false ratings (Mobasher et al., 2007). In other words, a robust RS can generate reliable recommendations despite the presence of noise, anomalies, or intentional manipulation in the data. Therefore, the robustness of an RS is an intrinsic characteristic to ensure its normal operation, understanding the reasons behind a recommendation may help to debug a model when attacked, especially when the system does not behave as expected (Doshi-Velez and Kim, 2017). There have been some works that show empirically that robustness is positively correlated to explainability in AI systems (Noack et al., 2021, Etmann et al., 2019). Therefore, it is worth investigating whether A^3R and CA-FATA can still provide accurate recommendations and

explain such recommendations when faced with malicious attacks. In SECE, we apply SHAP for selecting important features, these features are further validated through counterfactual reasoning. We plan to study whether SECE may help to detect malicious attacks, e.g. a sudden change of important features, etc.

Auditability (Accountability) through explainability: accountability typically encompasses three dimensions: responsibility, answerability, and sanctionability (Loi and Spielkamp, 2021). Auditability aims to scrutinize and understand harmful issues in AI systems (Bandy, 2021). With rising interest in these topics due to regulations like the General Data Protection Regulation (GDPR) (Voigt and Von dem Bussche, 2017), understanding the reasoning process behind recommendations becomes critical. It can help determine compliance with regulations and attribute responsibility when harm results from RSs (e.g., in medical applications). However, existing works on auditability or accountability rarely make this connection with explainability, suggesting a promising future research direction.

Additional Promising Perspectives: the above discussions focus on explainable recommendations for individual users. However, during the course of my Ph.D., I also applied the concept of explainable recommendations to public scenarios, specifically in crisis management. In our work (Le Ngoc et al., 2023a, **?**), we explored a constraint-based RS to distribute civilian vehicles to rescue points during floods[39]. In another work (Le Ngoc et al., 2023b), we explore how collaborative and social computing technologies help to contextualize crises. Under such scenarios, high-stakes decisions necessitate clarity for decision-makers about how recommendations (i.e., the distribution plan of vehicles) are made. Furthermore, fairness (how to equitably distribute vehicles to different danger zones?), robustness (how to create a resilient RS capable of handling various emergencies?), and auditability (how to define the responsibility of stakeholders?) also merit discussion. Thus, the concept of explainability in RSs has broader applicability, including in the realm of public recommendations.

10.3 List of publications

The publications that are mentioned in this thesis are the following:

- Zhong, J. and Negre, E. (2021). Ai: To interpret or to explain? In Congrès Inforsid(INFormatique des ORganisations et Systèmes d'Information et de Décision). This paper mainly contributes to Chapter 4, where we try to disambiguate the "interpretability" and "explainability", propose IES and GFGE in AI.
- Zhong, J. and Negre, E. (2022e). Towards improving user-recommender systems interactions. In 2022 IEEE/SICE International Symposium on System Integration (SII), pages 816–820. IEEE. This paper contributes to Section 3.7 in Chapter 3 and Section 9.2 in Chapter 9, where we discuss methods for evaluating explanations in RSs and how to generate context-aware explanations.
- Zhong, J. and Negre, E. (2022a). A^3R: Argumentative explanations for recommendations. In 2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA),

[39] In flood scenarios, people may be trapped in buildings and require evacuation to shelter centers.

pages 1–9. IEEE. This paper mainly contributes to Chapter 6 where we present A^3R, an instantiation of Algorithm 2 under the model-intrinsic explanation setting.

- Zhong, J. and Negre, E. (2023). Context-aware feature attribution through argumentation. Accepted by the CARS Workshop at Recsys 2023. This paper contributes to Chapter 7, where we present CA-FATA, an extension of A^3R.

- Zhong, J. and Negre, E. (2022c). Shap-enhanced counterfactual explanations for recommendations. In Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing, pages 1365–1372. This paper contributes to Chapter 8, where we present SECE, an instantiation of Algorithm 2 under the post-hoc explanation setting.

- Zhong, J. and Negre, E. (2022b). Context-aware explanations in recommender systems. In International Conference on Deep Learning, Artificial Intelligence and Robotics, pages 76–85. Springer. This paper contributes Section 9.2 in Chapter 9, where we present the results of the primary user study.

- Zhong, J. and Negre, E. (2022d). Towards better representation of context into recommender systems. International Journal of Knowledge-Based Organizations (IJKBO), 12(2):1–12. This paper contributes to Appendix A, where we present the detail of a black-box recommender model used in Chapter 8 and 9.

- Le Ngoc, L., Zhong, J., Negre, E., and Abel, M.-H. Constraint-based recommender system for crisis management simulations. In The 56th Hawaii International Conference on System Sciences, pages 1778–1789. In this paper, we develop a constraint-based recommender system that can be used for recommending civil vehicles to impacted areas during crisis management, which is mentioned in Chapter 10 where we discuss promising avenues for future work.

- Le Ngoc, L., Zhong, J., Negre, E., and Abel, M.-H. (2023b). Corec-cri: How collaborative and social technologies can help to contextualize crises? To appear in SMC 2023. In this paper, we investigate how collaborative and social computing technologies help to contextualize crises, including identifying impacted areas and real-time needs, which are mentioned in future work.

- J. Zhong, L. L. Ngoc, E. Negre, and M.-H. Abel, "Ontology-based crisis simulation system for population sheltering management," SIMULATION, p. 00 375 497 231 214 563, 2023. This paper is an extended version of Le Ngoc et al. (2023a), we further explore how ontology-based modeling benefits crisis management and how the evacuation process can be simulated, which are mentioned in future work.

Appendices

Appendix A

The black-box model we have developed

A.1 More detail on the black-box recommender in Section 8.4.1

In this section, we will provide more details on the black-box recommender model in the case study presented in Section 8.4, Chapter 8 and Section 9.2 in Chapter 9.

We will first introduce the CBPF method proposed by Ferdousi et al. (2018) and then present our modification, the symbols used are presented in Table A.1. To be more concrete, in a CARS, suppose there are m users; n items; q contextual factors (e.g. time, companion, location, etc); suppose we have $|c_k|$ (e.g. possible condition for companion is family, friend, colleague, etc.) possible conditions for contextual factor c_k, then $\sum_{k=1}^{k=q} |c_k| = p$, where p is the total number of contextual conditions and q is the total number of contextual factors; X_k is the kth observation in the dataset, r_k is the rating given by user u_k to item i_k, $c_{tk} = 1$ means contextual condition c_{tk} appears in k^{th} observation and $c_{tk} = 0$ means absent. For example, in a movie recommender system where ratings range from 1 to 5, $X_k = \{Mike, HarryPotter, 4, DaytypeWeekday = 1, DaytypeWeekend = 0, CompanionFamily = 0, CompanionFriends = 1, CompanionAlone = 0, MoodHappy = 1, MoodDisappointed = 0, MoodAngry = 0\}$ means that Mike watched the movie Harry Potter with his friends when he was happy in a weekday and he gave a score 4 out of 5 for this movie. A contextual condition c_j can be represented by $w_{c_j} = (w_{u_{cluster1},c_j}, w_{u_{cluster2},c_j}, \dots)$, the values are user-based (or item-based) PCC between contextual condition c_j and ratings calculated by Equation A.1; a target contextual situation $CS = (c_1, \dots c_q)$ is composed of q contextual conditions, which means that for each of the q contextual factors, only one contextual condition occurs in a target contextual situation. Considering a contextual factor "time", it can not be "morning" and "afternoon" at the same time.

The main steps of CBPF include:

- **Step 1**: calculate the user-based (or item-based) PCC between contextual condition c_j and ratings for each user (item) using Equation A.1. Note that for user-based PCC, the contextual conditions and ratings are selected according to users; for item-based PCC, the contextual conditions and ratings are selected according to items.

$$w_{u,c_j} = PCC_u(r, c_j) = \frac{\sum_{k \in K}(r_k - \bar{r_u})(c_{jk} - \bar{c_u})}{\sqrt{\sum_{k \in K}(r_k - \bar{r_u})^2}\sqrt{\sum_{k \in K}(c_{jk} - \bar{c_u})^2}} \tag{A.1}$$

Table A.1: List of key symbols

Symbol	Meaning
$U = \{u_1, u_2 \dots u_m\}$	Set of users
$I = \{i_1, i_2 \dots i_n\}$	Set of items
$C = \{c_1, c_2 \dots c_p\}$	Set of contextual conditions
$CF = \{cf_1, cf_2 \dots cf_q\}$	Set of contextual factors
$X_k = \{u_k, i_k, r_k, c_{1k}, c_{2k} \dots c_{pk}\}$	The kth observation in dataset
$w_{c_j} = (w_{u_{cluster1},c_j}, w_{u_{cluster2},c_j}, \dots)$	Representation of contextual condition c_j
$CS = (c_1, \dots c_q)$	A target contextual situation

where K is the set of observations $X_k = \{u, i_k, r_k, c_{1k}, c_{2k} \dots c_{pk}\}$ with user u, $\bar{c_u}$ is the mean value of the context condition c_j over observations for user u, $\bar{r_u}$ is the mean of the ratings given by the user u. Based on the calculated w_{u,c_j}, each contextual condition c_j can be represented by a vector whose size is the total number of users (or items) and whose values are the corresponding w_{u,c_j}, all between -1 and 1. Considering the large number of users and items, Ferdousi et al. (2018) proposed to first cluster the users and items into limited groups to reduce computation costs. For example, the characteristics such as age and/or sex, can be used to cluster users. Table A.2 is an example of this representation method. To be more clear, the values correspond to the PCC between the contextual conditions on the left and the ratings of a user cluster. For example, the first row in Table A.2 is the contextual condition "DaypeWeekday", its PCC with ratings for user cluster1, user cluster2 and user cluster3 is 0.18, -0.13 and -0.54 respectively.

- **Step 2:** represent contextual target situation based on its composing context conditions. In this step, Ferdousi et al. (2018) used 2 methods:
  - Aggregation: The mean of the corresponding representation vector of contextual conditions which compose a target contextual situation:

$$w_{CS} = \frac{\sum_{k=1}^{k=q} w_{c_k}}{q} \tag{A.2}$$

  - Concatenation: The concatenation of the composing contextual conditions:

$$w_{CS} = concat(w_{c_k}) \tag{A.3}$$

- **Step 3:** calculate the similarity between the target contextual situation CS^* and the contextual situations CS existing in the original dataset, the similarity here is the cosine similarity between the vectors that represent them.

$$sim(CS^*, CS) = cosine(\vec{w_{CS^*}}, \vec{w_{CS}}) \tag{A.4}$$

- **Step 4:** select a local dataset whose contextual situation is similar to the target contextual situation according to a similarity threshold. (Set as 0.5)
- **Step 5** A traditional 2D recommender technique[36] is applied in the local dataset to obtain recommendations.

[36] We follow Ferdousi et al. (2018) and use Vanilla Matrix Factorization

Table A.2: Examples of representation of contextual conditions (User-based and user clustered) (Step 1)

	Cluster1	Cluster2	Cluster3
DaytypeWeekday	0.18	-0.13	-0.54
DaytypeWeekend	0.45	0.56	0.62
CompanionFamily	0.67	0.12	-0.34
CompanionFriends	0.51	0.45	0.12
CompanionAlone	-0.18	-0.34	0.33
MoodHappy	0.45	0.67	0.65
MoodDisappointed	-0.45	-0.32	-0.31
MoodAngry	-0.24	-0.34	-0.12

The modification we made is mainly on **Step 3**, we believe that the representation of a target contextual situation should be personalized to select a more relevant local dataset and generate more accurate recommendations. Therefore, instead of aggregating or concatenating the representation of contextual conditions to represent a target contextual situation, we represent a target contextual situation by directly using the contextual conditions that occur in this contextual situation, in the following of this paper we call this method Relevance. For example, the target contextual situation for a user Mike is $CS = (DaytypeWeekday = 1, CompanionFriends = 1, MoodHappy = 1)$ and we suppose that according to his characteristic information (e.g. sex, age) he is in cluster2. By referring to Table A.2, contextual condition "DaytypeWeekday" is represented by a vector $w_{DaytypeWeekday} = (0.18, -0.13, -0.54)$, contextual condition "CompanionFriends" is represented by a vector $w_{CompanionFriends} = (0.51, 0.45, 0.12)$, contextual condition "MoodHappy" is represented by a vector $w_{MoodHappy} = (0.45, 0.67, 0.65)$.

- If the Aggregation method is adopted, according to Equation A.2, then the representation of this target contextual situation will be $w_{CS-Agg} = (\frac{0.18+0.51+0.45}{3}, \frac{-0.13+0.45+0.67}{3}, \frac{-0.54+0.12+0.65}{3})$ $= (0.38, 0.33, 0.08)$
- If the Concatenation method is adopted then $w_{DaytypeWeekday}$, $w_{CompanionFriends}$ and $w_{MoodHappy}$ are simply concatenated to represent this target contextual situation, the result will be $w_{CS-Con} = (0.18, -0.13, -0.54, 0.51, 0.45, 0.12, 0.45, 0.67, 0.65)$
- If the Relevance method is adopted, the PCC between "DaytypeWeekday" and ratings for cluster2 is -0.13; the PCC between "CompanionFriends" and ratings for cluster2 is 0.45; the PCC between "MoodHappy" and ratings for cluster2 is 0.67, then the representation of this target contextual situation will be $w_{CS-Rel} = (-0.13, 0.45, 0.67)$.

To summarize, the black-box model we propose here follows the pre-filtering principle.

Bibliography

Abdi, H. and Williams, L. J. (2010). Tukey's honestly significant difference (hsd) test. *Encyclopedia of research design*, 3(1):1–5.

Abdollahi, B. and Nasraoui, O. (2017). Using explainability for constrained matrix factorization. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 79–83.

Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., and Steggles, P. (1999). Towards a better understanding of context and context-awareness. In *International symposium on handheld and ubiquitous computing*, pages 304–307. Springer.

Adomavicius, G., Bauman, K., Tuzhilin, A., and Unger, M. (2021). Context-aware recommender systems: From foundations to recent developments context-aware recommender systems. In *Recommender Systems Handbook*, pages 211–250. Springer.

Adomavicius, G., Sankaranarayanan, R., Sen, S., and Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information systems (TOIS)*, 23(1):103–145.

Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749.

Adomavicius, G. and Tuzhilin, A. (2011). Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer.

Afchar, D. and Hennequin, R. (2020). Making neural networks interpretable with attribution: application to implicit signals prediction. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 220–229.

Al Jawarneh, I. M., Bellavista, P., Corradi, A., Foschini, L., Montanari, R., Berrocal, J., and Murillo, J. M. (2020). A pre-filtering approach for incorporating contextual information into deep learning based recommender systems. *IEEE Access*, 8:40485–40498.

Albini, E., Lertvittayakumjorn, P., Rago, A., and Toni, F. (2020). Deep argumentative explanations. *arXiv preprint arXiv:2012.05766*.

Amatriain, X. (2013). Mining large streams of user data for personalized recommendations. *ACM SIGKDD Explorations Newsletter*, 14(2):37–48.

Ancona, M., Ceolini, E., Öztireli, C., and Gross, M. (2019). Gradient-based attribution methods. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 169–191.

Arduin, P.-E., Grundstein, M., and Rosenthal-Sabroux, C. (2015). *Information and knowledge system*. John Wiley & Sons.

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115.

Baeza-Yates, R., Ribeiro-Neto, B., et al. (1999). *Modern information retrieval*, volume 463. ACM press New York.

Balog, K. and Radlinski, F. (2020). Measuring recommendation explanation quality: The conflicting goals of explanations. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 329–338.

Balog, K., Radlinski, F., and Arakelyan, S. (2019). Transparent, scrutable and explainable user models for personalized recommendation. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*, pages 265–274.

Baltrunas, L. and Amatriain, X. (2009). Towards time-dependant recommendation based on implicit feedback. In *Workshop on context-aware recommender systems (CARS'09)*, pages 25–30.

Baltrunas, L., Church, K., Karatzoglou, A., and Oliver, N. (2015). Frappe: Understanding the usage and perception of mobile app recommendations in-the-wild. *preprint arXiv:1505.03014*.

Baltrunas, L., Ludwig, B., Peer, S., and Ricci, F. (2011a). Context-aware places of interest recommendations and explanations. In *First International Workshop on Decision Making and Recommendation Acceptance Issues in Recommender Systems (DEMRA 2011)*, page 19.

Baltrunas, L., Ludwig, B., and Ricci, F. (2011b). Matrix factorization techniques for context aware recommendation. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 301–304.

Baltrunas, L. and Ricci, F. (2009). Context-dependent items generation in collaborative filtering. In *Proceedings of the 2009 Workshop on Context-Aware Recommender Systems*, pages 22–25.

Baltrunas, L. and Ricci, F. (2014). Experimental evaluation of context-dependent collaborative filtering using item splitting. *User Modeling and User-Adapted Interaction*, 24:7–34.

Bandy, J. (2021). Problematic machine behavior: A systematic literature review of algorithm audits. *Proceedings of the acm on human-computer interaction*, 5(CSCW1):1–34.

Baroni, P., Rago, A., and Toni, F. (2018). How many properties do we need for gradual argumentation? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Baroni, P., Rago, A., and Toni, F. (2019). From fine-grained properties to broad principles for gradual argumentation: A principled spectrum. *IJAR*, 105:252–286.

Barrett, C. and Tinelli, C. (2018). Satisfiability modulo theories. In *Handbook of model checking*, pages 305–343. Springer.

Bazire, M. and Brézillon, P. (2005). Understanding context before using it. In *Modeling and Using Context: 5thInternational and Interdisciplinary Conference CONTEXT 2005, Paris, France, July 5-8, 2005. Proceedings 5*, pages 29–40. Springer.

Bench-Capon, T. J. (2020). Before and after dung: Argumentation in ai and law. *Argument & Computation*, 11(1-2):221–238.

Bennett, J., Lanning, S., et al. (2007). The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York.

Bilgic, M. and Mooney, R. J. (2005). Explaining recommendations: Satisfaction vs. promotion. In *Beyond Personalization Workshop, IUI*, volume 5, page 153.

Bilodeau, B., Jaques, N., Koh, P. W., and Kim, B. (2022). Impossibility theorems for feature attribution. *arXiv preprint arXiv:2212.11870*.

Bistarelli, S. and Taticchi, C. (2022). A labelling semantics and strong admissibility for weighted argumentation frameworks. *Journal of Logic and Computation*, 32(2):281–306.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Bradley, A. P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159.

Breiman, L. (2001). Random forests. *Machine learning*, 45:5–32.

Briguez, C. E., Budán, M. C., Deagustini, C. A., Maguitman, A. G., Capobianco, M., and Simari, G. R. (2012). Towards an argument-based music recommender system. In *Computational Models of Argument*, pages 83–90. IOS Press.

Briguez, C. E., Budan, M. C., Deagustini, C. A., Maguitman, A. G., Capobianco, M., and Simari, G. R. (2014). Argument-based mixed recommenders and their application to movie suggestion. *Expert Systems with Applications*, 41(14):6467–6482.

Budán, M. C., Cobo, M. L., Martinez, D. C., and Simari, G. R. (2020a). Proximity semantics for topic-based abstract argumentation. *Information Sciences*, 508:135–153.

Budán, P. D., Escañuela Gonzalez, M. G., Budán, M. C. D., Martinez, M. V., and Simari, G. R. (2020b). Similarity notions in bipolar abstract argumentation. *Argument & Computation*, 11(1-2):103–149.

Burke, R. (2000). Knowledge-based recommender systems. *Encyclopedia of library and information systems*, 69(Supplement 32):175–186.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12:331–370.

Burke, R. (2007). Hybrid web recommender systems. *The adaptive web: methods and strategies of web personalization*, pages 377–408.

Carvalho, D. V., Pereira, E. M., and Cardoso, J. S. (2019). Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832.

Cayrol, C. and Lagasquie-Schiex, M.-C. (2005). On the acceptability of arguments in bipolar argumentation frameworks. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 378–389. Springer.

Celma, Ò. and Herrera, P. (2008). A new approach to evaluating novel recommendations. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 179–186.

Chapelle, O., Metlzer, D., Zhang, Y., and Grinspan, P. (2009). Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 621–630.

Chen, L. and Pu, P. (2005). Trust building in recommender agents. In *Proceedings of the Workshop on Web Personalization, Recommender Systems and Intelligent User Interfaces at the 2nd International Conference on E-Business and Telecommunication Networks*, pages 135–145. Citeseer.

Chen, X., Zhang, Y., and Wen, J.-R. (2022). Measuring" why" in recommender systems: a comprehensive survey on the evaluation of explainable recommendation. *arXiv preprint arXiv:2202.06466*.

Chin-Parker, S. and Bradner, A. (2010). Background shifts affect explanatory style: How a pragmatic theory of explanation accounts for background effects in the generation of explanations. *Cognitive processing*, 11:227–249.

Chouldechova, A. (2017). Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163.

Cosley, D., Lam, S. K., Albert, I., Konstan, J. A., and Riedl, J. (2003). Is seeing believing? how recommender system interfaces affect users' opinions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 585–592.

Costa, F., Ouyang, S., Dolog, P., and Lawlor, A. (2018). Automatic generation of natural language explanations. In *Proceedings of the 23rd international conference on intelligent user interfaces companion*, pages 1–2.

Cramer, H., Evers, V., Ramlal, S., Van Someren, M., Rutledge, L., Stash, N., Aroyo, L., and Wielinga, B. (2008). The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-adapted interaction*, 18(5):455.

Čyras, K., Rago, A., Albini, E., Baroni, P., and Toni, F. (2021). Argumentative xai: a survey. *arXiv preprint arXiv:2105.11266*.

Daher, J. B., Brun, A., and Boyer, A. (2017). A review on explanations in recommender systems.

Datta, A., Sen, S., and Zick, Y. (2016). Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE symposium on security and privacy (SP)*, pages 598–617. IEEE.

Dennett, D. C. (1989). *The intentional stance*. MIT press.

Deshpande, M. and Karypis, G. (2004). Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177.

Despres, C. and Chauvel, D. (1999). Knowledge management (s). *Journal of knowledge Management*, 3(2):110–123.

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Dourish, P. (2004). What we talk about when we talk about context. *Personal and ubiquitous computing*, 8:19–30.

Du, M., Liu, N., and Hu, X. (2019). Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77.

Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2).

Edwards, L. and Veale, M. (2017). Slave to the algorithm? why a'right to an explanation'is probably not the remedy you are looking for. *Duke L. & Tech. Rev.*, 16:18.

Etmann, C., Lunz, S., Maass, P., and Schönlieb, C.-B. (2019). On the connection between adversarial robustness and saliency map interpretability. *arXiv preprint arXiv:1905.04172*.

Ferdousi, Z. V. (2020). *From Traditional to Context-Aware Recommendations by Correlation-Based Context Model*. PhD thesis, Paris-Dauphine University, PSL Research University.

Ferdousi, Z. V., Colazzo, D., and Negre, E. (2018). Cbpf: leveraging context and content information for better recommendations. In *International Conference on Advanced Data Mining and Applications*, pages 381–391. Springer.

Ferwerda, B., Swelsen, K., and Yang, E. (2018). Explaining content-based recommendations. *New York*, pages 1–24.

Friedrich, G. and Zanker, M. (2011). A taxonomy for generating explanations in recommender systems. *AI Magazine*, 32(3):90–98.

Gabbay, D. M. (2016). Logical foundations for bipolar and tripolar argumentation networks: preliminary results. *Journal of Logic and Computation*, 26(1):247–292.

García, A. J. and Simari, G. R. (2004). Defeasible logic programming: An argumentative approach. *Theory and practice of logic programming*, 4(1-2):95–138.

Ge, M., Delgado-Battenfeld, C., and Jannach, D. (2010). Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 257–260.

Gedikli, F., Jannach, D., and Ge, M. (2014). How should i explain? a comparison of different explanation types for recommender systems. *International Journal of Human-Computer Studies*, 72(4):367–382.

Geng, S., Liu, S., Fu, Z., Ge, Y., and Zhang, Y. (2022). Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 299–315.

Ghazimatin, A., Balalau, O., Saha Roy, R., and Weikum, G. (2020). Prince: Provider-side interpretability with counterfactual explanations in recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 196–204.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., and Kagal, L. (2018). Explaining explanations: An approach to evaluating interpretability of machine learning. *arXiv preprint arXiv:1806.00069*.

Goodman, B. and Flaxman, S. (2017). European union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, 38(3):50–57.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42.

Gunawardana, A., Shani, G., and Yogev, S. (2021). Evaluating recommender systems. In *Recommender systems handbook*, pages 547–601. Springer.

Guo, M., Yan, N., Cui, X., Hughes, S., and Al Jadda, K. (2021). Online product feature recommendations with interpretable machine learning. *arXiv preprint arXiv:2105.00867*.

Hall, P. and Gill, N. (2019). *An introduction to machine learning interpretability*. O'Reilly Media, Incorporated.

Harper, F. M. and Konstan, J. A. (2015). The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19.

Hart, S. (1989). Shapley value. In *Game Theory*, pages 210–216. Springer.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Hastie, T. J. (2017). Generalized additive models. In *Statistical models in S*, pages 249–307. Routledge.

He, C., Parra, D., and Verbert, K. (2016). Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. *Expert Systems with Applications*, 56:9–27.

He, X. and Chua, T.-S. (2017). Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 355–364.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. (2017). Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182.

Heckel, R., Vlachos, M., Parnell, T., and Dünner, C. (2017). Scalable and interpretable product recommendations via overlapping co-clustering. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 1033–1044. IEEE.

Herlocker, J. L. (2000). *Understanding and improving automated collaborative filtering systems*. University of Minnesota.

Herlocker, J. L., Konstan, J. A., and Riedl, J. (2000). Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250.

Hilton, D. (2017). Social attribution and explanation.

Hoffman, R. R., Mueller, S. T., Klein, G., and Litman, J. (2018). Metrics for explainable ai: Challenges and prospects. *arXiv preprint arXiv:1812.04608*.

Hofmann, T. (2004). Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 22(1):89–115.

Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G. d., Gutierrez, C., Kirrane, S., Gayo, J. E. L., Navigli, R., Neumaier, S., et al. (2021). Knowledge graphs. *Synthesis Lectures on Data, Semantics, and Knowledge*, 12(2):1–257.

Hu, B., Shi, C., Zhao, W. X., and Yu, P. S. (2018). Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1531–1540.

Hu, Y., Koren, Y., and Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE international conference on data mining*, pages 263–272. Ieee.

Jain, S. and Wallace, B. C. (2019). Attention is not explanation. *arXiv preprint arXiv:1902.10186*.

Jannach, D., Zanker, M., Felfernig, A., and Friedrich, G. (2010). *Recommender systems: an introduction*. Cambridge University Press.

Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

Jones, N. A., Ross, H., Lynam, T., Perez, P., and Leitch, A. (2011). Mental models: an interdisciplinary synthesis of theory and methods. *Ecology and Society*, 16(1).

Josephson, J. R. and Josephson, S. G. (1996). *Abductive inference: Computation, philosophy, technology*. Cambridge University Press.

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

Karafili, E., Wang, L., and Lupu, E. C. (2020). An argumentation-based reasoner to assist digital investigation and attribution of cyber-attacks. *Forensic Science International: Digital Investigation*, 32:300925.

Keren, G. (2014). Between-or within-subjects design: A methodological dilemma. *A handbook for data analysis in the behaviorial sciences*, 1:257–272.

Kim, B., Glassman, E., Johnson, B., and Shah, J. (2015). ibcm: Interactive bayesian case model empowering humans via intuitive interaction.

Kim, D. and Suh, B. (2019). Enhancing vaes for collaborative filtering: flexible priors & gating mechanisms. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 403–407.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Konstan, J. A., McNee, S. M., Ziegler, C.-N., Torres, R., Kapoor, N., and Riedl, J. (2006). Lessons on applying automated recommender systems to information-seeking tasks. In *AAAI*, volume 6, pages 1630–1633.

Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Koren, Y., Rendle, S., and Bell, R. (2021). Advances in collaborative filtering. *Recommender systems handbook*, pages 91–142.

Košir, A., Odic, A., Kunaver, M., Tkalcic, M., and Tasic, J. F. (2011). Database for contextual personalization. *Elektrotehniški vestnik*, 78(5):270–274.

Kotkov, D., Wang, S., and Veijalainen, J. (2016). A survey of serendipity in recommender systems. *Knowledge-Based Systems*, 111:180–192.

Kotsiantis, S. and Kanellopoulos, D. (2006). Association rules mining: A recent overview. *GESTS International Transactions on Computer Science and Engineering*, 32(1):71–82.

Kouki, P., Schaffer, J., Pujara, J., O'Donovan, J., and Getoor, L. (2019). Personalized explanations for hybrid recommender systems. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 379–390.

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

Lan*, H., Ding, Y., and Hong, J. (2005). Decision support system for rapid prototyping process selection through integration of fuzzy synthetic evaluation and an expert system. *International Journal of Production Research*, 43(1):169–194.

Le Ngoc, L., Zhong, J., Negre, E., and Abel, M.-H. (2023a). Constraint-based recommender system for crisis management simulations. In *The 56th Hawaii International Conference on System Sciences*, pages 1778–1789.

Le Ngoc, L., Zhong, J., Negre, E., and Abel, M.-H. (2023b). Corec-cri: How collaborative and social technologies can help to contextualize crises? In *To appear in SMC 2023*.

Lewis, D. (1986). Causal explanation.

Li, J., Zhang, C., Zhou, J. T., Fu, H., Xia, S., and Hu, Q. (2021). Deep-lift: deep label-specific feature learning for image annotation. *IEEE Transactions on Cybernetics*, 52(8):7732–7741.

Li, L., Chen, L., and Dong, R. (2020a). Caesar: context-aware explanation based on supervised attention for service recommendations. *Journal of Intelligent Information Systems*, pages 1–24.

Li, L., Chen, L., and Zhang, Y. (2020b). Towards controllable explanation generation for recommender systems via neural template. In *Companion proceedings of the web conference 2020*, pages 198–202.

Lim, B. Y. and Dey, A. K. (2009). Assessing demand for intelligibility in context-aware applications. In *Proceedings of the 11th international conference on Ubiquitous computing*, pages 195–204.

Linden, G., Smith, B., and York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80.

Lipton, P. (1990). Contrastive explanation. *Royal Institute of Philosophy Supplements*, 27:247–266.

Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57.

Liu, J. and Wu, C. (2017). Deep learning based recommendation: A survey. In *Information Science and Applications 2017: ICISA 2017 8*, pages 451–458. Springer.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., and Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26.

Loi, M. and Spielkamp, M. (2021). Towards accountability in the use of artificial intelligence for public administrations. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 757–766.

Lombrozo, T. (2006). The structure and function of explanations. *Trends in cognitive sciences*, 10(10):464–470.

Lou, Y., Caruana, R., and Gehrke, J. (2012). Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–158.

Louizos, C., Shalit, U., Mooij, J. M., Sontag, D., Zemel, R., and Welling, M. (2017). Causal effect inference with deep latent-variable models. *Advances in neural information processing systems*, 30.

Lu, J., Wu, D., Mao, M., Wang, W., and Zhang, G. (2015). Recommender system application developments: a survey. *Decision Support Systems*, 74:12–32.

Luis-Argentina, S. (2008). Decision rules and arguments in defeasible decision making. *Computational Models of Argument: Proceedings of COMMA 2008*, 172:171.

Lundberg, S. M., Erion, G. G., and Lee, S.-I. (2018). Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*.

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

Luo, X., Zhou, M., Xia, Y., and Zhu, Q. (2014). An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Transactions on Industrial Informatics*, 10(2):1273–1284.

Mackie, J. L. (1980). *The cement of the universe: A study of causation*. Clarendon Press.

Malle, B. F. (2006). *How the mind explains behavior: Folk explanations, meaning, and social interaction*. MIT press.

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

Mattson, P., Cheng, C., Diamos, G., Coleman, C., Micikevicius, P., Patterson, D., Tang, H., Wei, G.-Y., Bailis, P., Bittorf, V., et al. (2020). Mlperf training benchmark. *Proceedings of Machine Learning and Systems*, 2:336–349.

McCarthy, K., Reilly, J., McGinty, L., and Smyth, B. (2005). Experiments in dynamic critiquing. In *Proceedings of the 10th international conference on Intelligent user interfaces*, pages 175–182.

McCrae, R. R. (1987). Creativity, divergent thinking, and openness to experience. *Journal of personality and social psychology*, 52(6):1258.

McInnes, L., Healy, J., and Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

McNee, S. M., Lam, S. K., Konstan, J. A., and Riedl, J. (2003). Interfaces for eliciting new user preferences in recommender systems. In *International Conference on User Modeling*, pages 178–187. Springer.

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38.

Misztal, J. and Indurkhya, B. (2015). Explaining contextual recommendations: Interaction design study and prototype implementation. In *IntRS@ RecSys*, pages 13–20.

Mittelstadt, B., Russell, C., and Wachter, S. (2019). Explaining explanations in ai. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 279–288.

Mobasher, B., Burke, R., Bhaumik, R., and Williams, C. (2007). Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. *ACM Transactions on Internet Technology (TOIT)*, 7(4):23–es.

Mohseni, S., Zarei, N., and Ragan, E. D. (2021). A multidisciplinary survey and framework for design and evaluation of explainable ai systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 11(3-4):1–45.

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

Molnar, C., Casalicchio, G., and Bischl, B. (2020). Interpretable machine learning–a brief history, state-of-the-art and challenges. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 417–431. Springer.

Montavon, G., Samek, W., and Müller, K.-R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15.

Musto, C., Gemmis, M. d., Lops, P., Narducci, F., and Semeraro, G. (2012). Semantics and content-based recommendations. In *Recommender systems handbook*, pages 251–298. Springer.

Musto, C., Rossiello, G., de Gemmis, M., Lops, P., and Semeraro, G. (2019). Combining text summarization and aspect-based sentiment analysis of users' reviews to justify recommendations. In *Proceedings of the 13th ACM conference on recommender systems*, pages 383–387.

Musto, C., Spillo, G., de Gemmis, M., Lops, P., and Semeraro, G. (2020). Exploiting distributional semantics models for natural language context-aware justifications for recommender systems. In *IntRS@ RecSys*, pages 65–71.

Naveed, S., Donkers, T., and Ziegler, J. (2018). Argumentation-based explanations in recommender systems: Conceptual framework and empirical results. In *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization*, pages 293–298.

Negre, E. (2017). Recommender system: Generality, evaluation and improvements. Habilitation à Diriger des Recherches, Université Toulouse 1 - Capitole.

Nikolakopoulos, A. N., Ning, X., Desrosiers, C., and Karypis, G. (2021). Trust your neighbors: a comprehensive survey of neighborhood-based methods for recommender systems. *Recommender Systems Handbook*, pages 39–89.

Noack, A., Ahern, I., Dou, D., and Li, B. (2021). An empirical study on the relation between network interpretability and adversarial robustness. *SN Computer Science*, 2:1–13.

Nóbrega, C. and Marinho, L. (2019). Towards explaining recommendations through local surrogate models. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 1671–1678.

Nozick, R. (1983). *Philosophical explanations*. Harvard University Press.

Nunes, I. and Jannach, D. (2017). A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction*, 27(3):393–444.

O'Connor, E., McCormack, T., and Feeney, A. (2014). Do children who experience regret make better decisions? a developmental study of the behavioral consequences of regret. *Child Development*, 85(5):1995–2010.

Panniello, U., Tuzhilin, A., and Gorgoglione, M. (2014). Comparing context-aware recommender systems in terms of accuracy and diversity. *User Modeling and User-Adapted Interaction*, 24:35–65.

Panniello, U., Tuzhilin, A., Gorgoglione, M., Palmisano, C., and Pedone, A. (2009). Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender systems. In *Proceedings of the third ACM conference on Recommender systems*, pages 265–268.

Papadimitriou, A., Symeonidis, P., and Manolopoulos, Y. (2012). A generalized taxonomy of explanations styles for traditional and social recommender systems. *Data Mining and Knowledge Discovery*, 24:555–583.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Park, H., Jeon, H., Kim, J., Ahn, B., and Kang, U. (2017). Uniwalk: Explainable and accurate recommendation for rating and network data. *arXiv preprint arXiv:1710.07134*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037.

Peake, G. and Wang, J. (2018). Explanation mining: Post hoc interpretability of latent factor models for recommendation systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2060–2069.

Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia*, 4(2):1883.

Polanyi, M. (1967). Sense-giving and sense-reading. *Philosophy*, 42(162):301–325.

Proietti, M. and Toni, F. (2023). A roadmap for neuro-argumentative learning.

Pu, P. and Chen, L. (2007). Trust-inspiring explanation interfaces for recommender systems. *Knowledge-Based Systems*, 20(6):542–556.

Pu, P., Chen, L., and Hu, R. (2012). Evaluating recommender systems from the user's perspective: survey of the state of the art. *User Modeling and User-Adapted Interaction*, 22:317–355.

Qin, J., Ren, K., Fang, Y., Zhang, W., and Yu, Y. (2020). Sequential recommendation with dual side neighbor-based collaborative relation modeling. In *Proceedings of the 13th international conference on web search and data mining*, pages 465–473.

Quijano-Sanchez, L., Sauer, C., Recio-Garcia, J. A., and Diaz-Agudo, B. (2017). Make it personal: a social explanation system applied to group recommendations. *Expert Systems with Applications*, 76:36–48.

Rago, A., Cocarascu, O., Bechlivanidis, C., Lagnado, D., and Toni, F. (2021). Argumentative explanations for interactive recommendations. *Artificial Intelligence*, 296:103506.

Rago, A., Cocarascu, O., Bechlivanidis, C., and Toni, F. (2020). Argumentation as a framework for interactive explanations for recommendations. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, volume 17, pages 805–815.

Rago, A., Cocarascu, O., and Toni, F. (2018). Argumentation-based recommendations: Fantastic explanations and how to find them. In *Proceedings of IJCAI*, pages 1949–1955.

Ramon, Y., Martens, D., Provost, F., and Evgeniou, T. (2020). A comparison of instance-level counterfactual explanation algorithms for behavioral and textual data: Sedc, lime-c and shap-c. *Advances in Data Analysis and Classification*, 14(4):801–819.

Rendle, S. (2010). Factorization machines. In *2010 IEEE International conference on data mining*, pages 995–1000. IEEE.

Rendle, S. (2012). Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):1–22.

Rendle, S., Krichene, W., Zhang, L., and Anderson, J. (2020). Neural collaborative filtering vs. matrix factorization revisited. In *Fourteenth ACM conference on recommender systems*, pages 240–248.

Resnick, P. and Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, 40(3):56–58.

Ribeiro, M. T., Singh, S., and Guestrin, C. (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*, pages 1135–1144.

Ricci, F., Rokach, L., and Shapira, B. (2015). Recommender systems: introduction and challenges. *Recommender systems handbook*, pages 1–34.

Ricci, F., Rokach, L., and Shapira, B. (2021). Recommender systems: Techniques, applications, and challenges. *Recommender Systems Handbook*, pages 1–35.

ROUGE, L. C. (2004). A package for automatic evaluation of summaries. In *Proceedings of Workshop on Text Summarization of ACL, Spain*.

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.

Russell, S. J. and Norvig, P. (2016). Artificial intelligence: a modern approach. malaysia.

Sagi, O. and Rokach, L. (2018). Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1249.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295.

Sassoon, I., Kökciyan, N., Sklar, E., and Parsons, S. (2019). Explainable argumentation for wellness consultation. In *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, pages 186–202. Springer.

Sato, M., Ahsan, B., Nagatani, K., Sonoda, T., Zhang, Q., and Ohkuma, T. (2018). Explaining recommendations using contexts. In *23rd international conference on intelligent user interfaces*, pages 659–664.

Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626.

Sharma, A. and Cosley, D. (2013). Do social explanations work? studying and modeling the effects of social explanations in recommender systems. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1133–1144.

Shrikumar, A., Greenside, P., and Kundaje, A. (2017). Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR.

Silveira, T., Zhang, M., Lin, X., Liu, Y., and Ma, S. (2019). How good your recommender system is? a survey on evaluations in recommendation. *International Journal of Machine Learning and Cybernetics*, 10(5):813–831.

Singh, J. and Anand, A. (2019). Exs: Explainable search using local model agnostic interpretability. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 770–773.

Slack, D., Hilgard, S., Jia, E., Singh, S., and Lakkaraju, H. (2020). Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. (2017). Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Symeonidis, P., Nanopoulos, A., and Manolopoulos, Y. (2009). Moviexplain: a recommender system with explanations. In *Proceedings of the third ACM conference on Recommender systems*, pages 317–320.

Tabachnick, B. G. and Fidell, L. S. (2007). *Experimental designs using ANOVA*. Thomson/Brooks/Cole Belmont, CA.

Tan, J., Xu, S., Ge, Y., Li, Y., Chen, X., and Zhang, Y. (2021). Counterfactual explainable recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1784–1793.

Teze, J. C., Gottifredi, S., García, A. J., and Simari, G. R. (2015). Improving argumentation-based recommender systems through context-adaptable selection criteria. *Expert Systems with Applications*, 42(21):8243–8258.

Teze, J. C. L., Godo, L., and Simari, G. R. (2018). An argumentative recommendation approach based on contextual aspects. In *SUM*, pages 405–412. Springer.

Tintarev, N. and Masthoff, J. (2007). A survey of explanations in recommender systems. In *2007 IEEE 23rd international conference on data engineering workshop*, pages 801–810. IEEE.

Tintarev, N. and Masthoff, J. (2015). Explaining recommendations: Design and evaluation. In *Recommender systems handbook*, pages 353–382. Springer.

Tintarev, N. and Masthoff, J. (2022). Beyond explaining single item recommendations. In *Recommender Systems Handbook*, pages 711–756. Springer.

Toniolo, A., Bowles, J. K., et al. (2020). Dialogue games for explaining medication choices. In *International Joint Conference on Rules and Reasoning*, pages 97–111. Springer.

Tsuchiya, S. (1993). Improving knowledge creation ability through organizational learning. In *ISMICK'93 Proceedings, International Symposium on the Management of Industrial and Corporate Knowledge*, pages 87–95.

Unger, M., Bar, A., Shapira, B., and Rokach, L. (2016). Towards latent context-aware recommendation systems. *Knowledge-Based Systems*, 104:165–178.

Unger, M., Tuzhilin, A., and Livne, A. (2020). Context-aware recommendations based on deep learning frameworks. *ACM Transactions on Management Information Systems*, 11(2):1–15.

Van Bouwel, J. and Weber, E. (2002). Remote causes, bad explanations? *Journal for the Theory of Social Behaviour*, 32(4):437–449.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Vassiliades, A., Bassiliades, N., and Patkos, T. (2021). Argumentation and explainable artificial intelligence: a survey. *The Knowledge Engineering Review*, 36.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., et al. (2017). Graph attention networks. *stat*, 1050(20):10–48550.

Velmurugan, T. and Santhanam, T. (2010). Computational complexity between k-means and k-medoids clustering algorithms for normal and uniform distributions of data points. *Journal of computer science*, 6(3):363.

Vig, J., Sen, S., and Riedl, J. (2009). Tagsplanations: explaining recommendations using tags. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 47–56.

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

Villegas, N. M., Sánchez, C., Díaz-Cely, J., and Tamura, G. (2018). Characterizing context-aware recommender systems: A systematic literature review. *Knowledge-Based Systems*, 140:173–200.

Vilone, G. and Longo, L. (2021). Notions of explainability and evaluation approaches for explainable artificial intelligence. *Information Fusion*, 76:89–106.

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

Visani, G., Bagli, E., Chesani, F., Poluzzi, A., and Capuzzo, D. (2022). Statistical stability indices for lime: Obtaining reliable explanations for machine learning models. *Journal of the Operational Research Society*, 73(1):91–101.

Voigt, P. and Von dem Bussche, A. (2017). The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555.

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

Wang, H., Zhang, F., Zhang, M., Leskovec, J., Zhao, M., Li, W., and Wang, Z. (2019a). Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 968–977.

Wang, H., Zhao, M., Xie, X., Li, W., and Guo, M. (2019b). Knowledge graph convolutional networks for recommender systems. In *The world wide web conference*, pages 3307–3313.

Wang, N., Wang, H., Jia, Y., and Yin, Y. (2018). Explainable recommendation via multi-task learning in opinionated text data. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 165–174.

Wang, S., Hu, L., Wang, Y., Cao, L., Sheng, Q. Z., and Orgun, M. (2019c). Sequential recommender systems: challenges, progress and prospects. *arXiv preprint arXiv:2001.04830*.

Wang, W. and Benbasat, I. (2007). Recommendation agents for electronic commerce: Effects of explanation facilities on trusting beliefs. *Journal of Management Information Systems*, 23(4):217–246.

Wang, X., He, X., Wang, M., Feng, F., and Chua, T.-S. (2019d). Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR*, pages 165–174.

Wang, Y., Ma, W., Zhang, M., Liu, Y., and Ma, S. (2023). A survey on the fairness of recommender systems. *ACM Transactions on Information Systems*, 41(3):1–43.

Wasinger, R., Wallbank, J., Pizzato, L., Kay, J., Kummerfeld, B., Böhmer, M., and Krüger, A. (2013). Scrutable user models and personalised item recommendation in mobile lifestyle applications. In *User Modeling, Adaptation, and Personalization: 21th International Conference, UMAP 2013, Rome, Italy, June 10-14, 2013 Proceedings 21*, pages 77–88. Springer.

Watson, A. and Shipman, S. (2008). Using learner generated examples to introduce new concepts. *Educational Studies in Mathematics*, 69:97–109.

Wu, J., He, X., Wang, X., Wang, Q., Chen, W., Lian, J., and Xie, X. (2022). Graph convolution machine for context-aware recommender system. *Frontiers of Computer Science*, 16(6):166614.

Xian, Y., Zhao, T., Li, J., Chan, J., Kan, A., Ma, J., Dong, X. L., Faloutsos, C., Karypis, G., Muthukrishnan, S., et al. (2021). Ex3: Explainable attribute-aware item-set recommendations. In *Proceedings of the 15th ACM Conference on Recommender Systems*, pages 484–494.

Xiao, J., Ye, H., He, X., Zhang, H., Wu, F., and Chua, T.-S. (2017). Attentional factorization machines: Learning the weight of feature interactions via attention networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 3119–3125. AAAI Press.

Zadeh, L. A. (2013). *Computing with words in Information/Intelligent systems 1: Foundations*, volume 33. Physica.

Zafar, M. R. and Khan, N. (2021). Deterministic local interpretable model-agnostic explanations for stable explainability. *Machine Learning and Knowledge Extraction*, 3(3):525–541.

Zamani, H. and Croft, W. B. (2020). Learning a joint search and recommendation model from user-item interactions. In *Proceedings of the 13th international conference on web search and data mining*, pages 717–725.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer.

Zeng, Z., Fan, X., Miao, C., Leung, C., Jih, C. J., and Soon, O. Y. (2018). Context-based and explainable decision making with argumentation. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1114–1122.

Zhang, M. and Hurley, N. (2008). Avoiding monotony: improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 123–130.

Zhang, Y., Chen, X., et al. (2020). Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 14(1):1–101.

Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., and Ma, S. (2014). Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 83–92.

Zhao, X., Zhu, Z., Zhang, Y., and Caverlee, J. (2020). Improving the estimation of tail ratings in recommender system with multi-latent representations. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 762–770.

Zhong, J. and Negre, E. (2021). Ai: To interpret or to explain? In *Congrès Inforsid (INFormatique des ORganisations et Systèmes d'Information et de Décision)*.

Zhong, J. and Negre, E. (2022a). A 3 r: Argumentative explanations for recommendations. In *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–9. IEEE.

Zhong, J. and Negre, E. (2022b). Context-aware explanations in recommender systems. In *International Conference on Deep Learning, Artificial Intelligence and Robotics*, pages 76–85. Springer.

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

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

Zhong, J. and Negre, E. (2022e). Towards improving user-recommender systems interactions. In *2022 IEEE/SICE International Symposium on System Integration (SII)*, pages 816–820. IEEE.

Zhong, J. and Negre, E. (2023). Context-aware feature attribution through argumentation. Accepted in CARS Workshop colocated with Recsys 2023.

Zhou, J., Gandomi, A. H., Chen, F., and Holzinger, A. (2021). Evaluating the quality of machine learning explanations: A survey on methods and metrics. *Electronics*, 10(5):593.

Zhou, K., Wang, H., Zhao, W. X., Zhu, Y., Wang, S., Zhang, F., Wang, Z., and Wen, J.-R. (2020). S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1893–1902.

Zhou, R. and Chan, A. H. (2017). Using a fuzzy comprehensive evaluation method to determine product usability: A proposed theoretical framework. *Work*, 56(1):9–19.

Zhu, J., Liapis, A., Risi, S., Bidarra, R., and Youngblood, G. M. (2018). Explainable ai for designers: A human-centered perspective on mixed-initiative co-creation. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8. IEEE.

Zimmermann, H.-J. (2011). *Fuzzy set theory—and its applications*. Springer Science & Business Media.

RÉSUMÉ

Cette thèse se concentre sur le domaine des recommandations explicables, souvent mentionnées dans le contexte de l'intelligence artificielle explicable (XAI). Ce sont des recommandations faites par un système qui fournit également des informations pour justifier pourquoi ces recommandations ont été générées. Plus spécifiquement, le problème des recommandations explicables implique de générer des recommandations précises et de créer des explications pour justifier ces recommandations.

Nous commençons par aborder l'ambiguïté répandue dans les termes couramment utilisés en Intelligence Artificielle (IA), tels que "interprétabilité" et "explicabilité". Pour lever cette ambiguïté sur des termes souvent utilisés mais mal définis, nous introduisons le Schéma Interpréter/Expliquer (IES). L'IES aide à clarifier ces termes et constitue la base d'un cadre général et innovant pour générer des explications en IA. Ce cadre intègre de manière transparente les approches existantes pour générer des explications: les explications intrinsèques au modèle et les explications post-hoc. Dans le cadre des explications intrinsèques au modèle, nous introduisons deux modèles qui instancient l'IES: *Attribute-Aware Argumentative Recommender* (A^3R) et *Context-Aware Feature Attribution Through Argumentation* (CA-FATA). Les deux conduisent l'attribution de caractéristiques par argumentation. A^3R considère les caractéristiques des éléments recommandables comme des arguments qui déterminent les recommandations, tandis que CA-FATA étend cette approche pour prendre en compte les contextes des utilisateurs. Les deux modèles montrent une performance compétitive face aux modèles existants, offrant des explications transparentes et adaptables. Dans le cadre des explications post-hoc, *Shap-Enhanced Counterfactual Explanation* (SECE) est introduite, validant l'attribution de caractéristiques par raisonnement contrefactuel. Cette approche peut être universellement appliquée à n'importe quel modèle de recommandation. La thèse comprend également une étude utilisateur primaire sous la forme d'un questionnaire explorant l'influence des informations contextuelles sur les explications au sein des systèmes de recommandation (RSs). Ce questionnaire centré sur l'utilisateur aide à souligner les objectifs de promotion de la transparence, de l'efficacité, de la persuasion et de la confiance dans les RSs.

Dans l'ensemble, cette recherche non seulement lève l'ambiguïté sur les termes clés du domaine, mais propose également des méthodologies innovantes pour les recommandations explicables. Les instanciations de l'IES démontrent son applicabilité et contribuent à une compréhension essentielle de la génération d'explication dans les RSs et donc en IA, offrant de nouvelles voies pour des RSs transparents et dignes de confiance.

MOTS CLÉS

Recommandations explicables, Apprentissage automatique, Système d'information

ABSTRACT

This thesis focuses on explainable recommendations, often referred to within the context of eXplainable Artificial Intelligence (XAI), which are recommendations made by a system that also provides information to justify why such recommendations have been generated. More specifically, the problem of explainable recommendations involves generating accurate recommendations and generating explanations to justify such recommendations.

We start by addressing the prevalent ambiguity in terms commonly used in AI, such as "interpretability" and "explainability". To disambiguate the frequently used yet ill-defined terms, we introduce Interpret/Explain Schema (IES). The IES helps clarify these terms and forms the foundation of a novel, general framework for generating explanations in AI. This framework seamlessly incorporates existing approaches for generating explanations: model-intrinsic and post-hoc explanations. Within the model-intrinsic explanation setting, we introduce two models that instantiate IES: Attribute-Aware Argumentative Recommender (A^3R) and Context-Aware Feature Attribution Through Argumentation (CA-FATA). Both of them conduct feature attribution through argumentation. A^3R views item features as arguments that determine recommendations, while CA-FATA extends this approach to consider user contexts. Both models demonstrate competitive performance against existing models, offering transparent and adaptable explanations. In the post-hoc explanation setting, the Shap-Enhanced Counterfactual Explanation (SECE) is introduced, validating feature attribution via counterfactual reasoning. This approach can be applied universally to any recommender model. The thesis also encompasses a primary user study exploring the influence of contextual information on explanations within recommender systems (RSs). This user-centric questionnaire helps to emphasize the goals of promoting transparency, effectiveness, persuasiveness, and trust in RSs.

Overall, this research not only disambiguates key terms in the field but also offers innovative methodologies for explainable recommendations. The instantiations of IES demonstrate its applicability and contribute a vital understanding of explanation generation in RSs and thus in AI, offering new pathways for transparent and trustworthy RSs.

KEYWORDS

Explainable recommendations, Machine learning, Information system