Sorbonne université

École doctorale Informatique, Télécommunications et Électronique (Paris)

Equipe LFI, LIP6

Génération d'explications post-hoc personnalisées

Adulam Jeyasothy

Thèse de doctorat d'Informatique

Présentée et soutenue publiquement le 20 février 2024

Devant un jury composé de :

Wassila Ouerdane	MICS, Univ. Paris-Saclay, Centrale Supélec	Rapportrice
Benjamin Quost	Heudiasyc, Univ. de Technologie de Compiègne	Rapporteur
Salem Benferhat	CRIL, Université d'Artois	Examinateur
Grégory Bourguin	LISIC, Univ. Littoral Côte d'Opale	Examinateur
Marc Plantevit	LRE, EPITA	Président du jury
Marie-Jeanne Lesot	LIP6, Sorbonne Université	Directrice de thèse
Christophe Marsala	LIP6, Sorbonne Université	Directeur de thèse
Thibault Laugel	AXA, Paris	Encadrant de thèse

Table des matières

Remerciements

Je remercie tout d'abord mes trois encadrants de thèse : Marie-Jeanne Lesot, Christophe Marsala et Thibault Laugel. Ils m'ont accompagnée tout au long de ces trois années et m'ont aidée à évoluer dans le domaine de la recherche.

Je remercie ensuite Agnès Rico pour sa collaboration, elle m'a aidée à étudier un nouvel axe au sein de ma thèse.

Je remercie également l'équipe LFI, et en particulier les doctorants pour leur bonne humeur et leur soutien.

Enfin, je remercie mes parents, mes amis et en particulier ma meilleure amie Ilham de m'avoir soutenue et de m'avoir encouragée à faire cette thèse.

Chapitre 1

Introduction

1.1 Contexte

L'intelligence artificielle est présente dans une majeure partie de notre vie aussi bien dans notre quotidien, lorsque l'on choisit un film ou que l'on souhaite trouver un itinéraire pour voyager, que pour des tâches plus spécifiques comme l'analyse d'images radiologiques pour le domaine médical ou la détection d'anomalies en cybersécurité. Les tâches résolues par l'intelligence artificielle peuvent être écrites comme différents problèmes formels : dans le cas de l'apprentissage automatique supervisé, ou *supervised machine learning*, il s'agit par exemple de construire un modèle de prédiction qui associe une valeur à une donnée en fonction de caractéristiques descriptives de celles-ci. Nous nous plaçons particulièrement dans le cadre de la classification où on considère un modèle de prédiction qui associe une étiquette, appelée classe, à une donnée. Des exemples de modèles d'apprentissage pour la classification incluent les arbres de décision, les machines à vecteurs de support ou encore les réseaux de neurones. A titre d'illustration, on peut considérer un cas d'estimation de bien immobilier : un modèle prédit par exemple si une maison, décrite par des caractéristiques comme sa localisation, son nombre de pièces ou son ancienneté, appartient aux classes cher, abordable ou bon marché. Le modèle a pour but de faciliter la tâche de l'agent immobilier et d'accélérer le processus d'estimation.

Les méthodes d'intelligence artificielle sont traditionnellement évaluées selon leur capacité à résoudre la tâche fixée. Ainsi, dans le cas de classification, la performance des modèles est définie par leur capacité à prédire correctement les classes à partir des caractéristiques d'une donnée, par exemple par des mesures comme la précision ou le rappel. Les modèles utilisés actuellement sont de plus en plus performants selon ces critères, souvent au prix d'une complexité accrue. Par exemple, les modèles d'arbres de décision sont souvent remplacés par des forêts aléatoires pour augmenter la performance. De même, les réseaux de neurones profonds sont plus performants mais aussi beaucoup plus complexes que les réseaux de neurones classiques.

Étant donné que les modèles récents sont complexes, il est difficile de savoir ce qui permet d'obtenir leurs résultats. Ils peuvent être vus comme des boîtes noires où seules

les informations d'entrée et de sortie sont connues sans qu'aucune indication ne soit accessible sur les raisons de la valeur de sortie. Or, ne pas connaître les raisons de la prédiction peut être problématique car cela peut diminuer la confiance de l'utilisateur, ou augmenter la curiosité de l'utilisateur qui souhaite acquérir des connaissances à partir du modèle. Premièrement, la perte de confiance peut amener l'utilisateur à ne pas utiliser les modèles bien qu'ils soient performants, en particulier dans le cas où la prédiction du modèle présente un enjeu important. Par exemple, considérons, dans un cadre médical, un médecin qui utilise un modèle pour prédire si un patient a une maladie. Il ne peut pas prendre le risque d'obtenir une prédiction fausse, donc il est nécessaire d'accompagner la prédiction d'une explication pour que celle-ci soit acceptable par le médecin. Deuxièmement, la curiosité de l'utilisateur a pour but de comprendre la procédure mise en œuvre par le modèle pour l'adopter dans certains cas de figure, notamment dans le cas où la prédiction concerne des tâches simples du quotidien. Par exemple, lors de l'estimation d'une maison, un utilisateur peut souhaiter connaître les raisons qui ont conduit aux prix de différentes maisons pour estimer son propre bien.

Ces questions ne se posent pas uniquement dans le cadre d'applications d'outils d'intelligence artificielle, elles correspondent également à des contraintes légales par la mise en place de loi aussi bien au niveau national qu'au niveau européen. En 2016, le parlement européen a mis en place le Règlement Général sur la Protection des Données (RGPD) qui a pour but de protéger les données des utilisateurs, en apportant une transparence sur la manière dont ces données sont utilisées. L'article 39.I.5 donne le droit à toute personne de demander des informations sur la procédure mise en œuvre lors d'un traitement automatisé et de la contester. En 2017 [1], ce droit a été revendiqué par un ancien étudiant dont la demande d'admission à une université sur l'application APB a été refusée. L'application n'ayant pas pu fournir les raisons du refus pour l'affectation, le tribunal administratif de Bordeaux a donné raison à l'étudiant. Ce jugement a notamment entraîné la mise en demeure de APB par la CNIL. Le RGPD est général et concerne tous les types d'outils informatiques, la commission européenne a instauré en 2021 une nouvelle loi centrée sur l'intelligence artificielle nommée AI Act. Cette loi impose que les systèmes d'IA fournissent des informations sur leurs limites, leurs applications et l'algorithme proposé, notamment pour éviter l'utilisation de caractéristiques personnelles. Toutefois, ces lois sont assez vagues, les notions comme transparence ou compréhension ne sont pas définies, ce qui rend difficile leur mise en application.

Dans le domaine scientifique, ces questions ont donné lieu à un nouveau domaine qui est l'intelligence artificielle explicable, ou *eXplainable Artificial Intelligence* (XAI) dont l'objectif général est d'enrichir les prédictions d'un modèle automatique par des informations pouvant constituer une explication (Verma et al., 2020; Molnar, 2022). Or, cette notion est vague et les besoins des utilisateurs peuvent être variés : l'utilisateur peut souhaiter une explication pour différentes raisons. Généralement, la question qu'il se pose est : "Pourquoi ?", qui elle-même peut être déclinée de différentes manières selon

1. https://www.liberation.fr/france/2016/06/23/apb-un-etudiant-recale-de-la-fac-par-tirage-au-sort-gagne-en-justice_1461536/

le but de l'utilisateur. Dans un cas, un utilisateur peut demander pourquoi, dans son cas précis, le modèle lui donne la prédiction obtenue, alors que, dans un autre cas, il peut s'interroger à une échelle plus large et demander globalement pourquoi le modèle donne ses résultats. Très souvent, cette question du "pourquoi" se transforme en "comment", l'utilisateur se demande comment il doit modifier ses caractéristiques pour avoir la prédiction souhaitée ?. Dans ce cas, l'explication a un but complètement différent, elle ne sert pas à comprendre le modèle mais elle a pour but d'aider l'utilisateur à obtenir ce qu'il souhaite. Diverses formes d'explications peuvent alors être considérées, selon le type d'explications attendu. Par exemple, les vecteurs d'importance constituent une forme d'explication qui répond à la question : "Pourquoi obtient-on cette prédiction ?", ils mettent en valeur les caractéristiques principales considérées par le modèle. Pour l'exemple des appartements, ce type d'explication peut être demandé dans le cas où un vendeur souhaite comprendre l'estimation d'un appartement, une explication pourrait être : "le quartier où se situe l'appartement a une grande importance sur la prédiction". Dans le cas où l'utilisateur veut savoir comment il peut obtenir une autre prédiction, les explications contre-factuelles fournissent les modifications qu'il doit effectuer pour avoir ce qu'il souhaite. Pour l'exemple de la vente des appartements, une explication contre-factuelle peut être attendue par un acheteur qui souhaite une maison qui a l'ensemble des caractéristiques souhaitées mais que le modèle estime chère, il cherche alors à savoir quelles concessions doivent être effectuées pour avoir une maison moins chère. Une explication peut par exemple être : "il faut changer de quartier pour trouver une maison moins chère".

Les méthodes de l'IA explicable diffèrent d'abord selon leur articulation avec les méthodes d'apprentissage : les approches dites *post hoc* considèrent un modèle déjà entraîné et l'étudient dans une étape ultérieure distincte de celle d'entraînement. Les approches dites *ad-hoc* quant à elles construisent l'explication au fur et à mesure que le modèle est appris, pour construire un système qui génère à la fois la prédiction et l'explication. Les méthodes d'explications diffèrent également selon les hypothèses qu'elles font sur les informations considérées comme disponibles : les approches *agnostiques* vis-à-vis du modèle et des données considèrent qu'aucune connaissance n'est disponible, ni sur le modèle d'apprentissage (par exemple sur la forme de la frontière de décision), ni sur les données (par exemple sur leur distribution). Ces approches ont la particularité d'être applicables à tout type de modèles et tout type d'applications comme par exemple le domaine médical, le domaine bancaire ou encore le domaine de l'immobilier.

1.2 Problématique

De nombreuses méthodes de génération d'explications se placent dans un cas agnostique selon le modèle et selon les données. Cela leur donne une grande généralité, elles sont applicables à tout domaine et tout modèle. Dans de nombreux cas d'applications réalistes où l'utilisateur qui s'interroge est en effet en bout de chaîne et n'a pas accès aux détails techniques du développement de la méthode de prédiction, cette généralisation

est un grand avantage. Cependant, il a été montré que cette absence de connaissances entraîne des problèmes, tels que le risque de générer des explications non réalistes (Laugel et al., 2019) ou la modification des attributs selon des actions non réalisables (Barocas et al., 2020). Pour l'exemple des appartements, une explication problématique peut être : "La maison doit être à Paris et le code postal associé doit être 94000". Cette explication n'est pas réaliste car une maison à Paris est nécessairement associée à un code postal commençant par 75, les maisons qui ne respectent pas la contrainte ci-dessus n'existent pas. Dans le cas où une personne souhaite augmenter la valeur de sa maison, une autre explication problématique serait : "la maison doit être dans une autre ville". Cette explication n'est pas utile, car cette action n'est pas faisable par le vendeur. En conséquence, les explications peuvent ne pas être toujours comprises ou utilisées par l'utilisateur (Rudin, 2019).

Pour résoudre ces problèmes, certaines méthodes proposent d'enrichir les informations considérées en entrée et d'intégrer des connaissances en plus de la prédiction du modèle (Mahajan et al., 2019; Frye et al., 2020; Ustun et al., 2019). Ces connaissances peuvent être associées au domaine étudié, constituer une vérité générale ou être associées à l'utilisateur. Dans l'exemple précédent, associer les codes postaux commençant par 75 à Paris est une connaissance générale, l'intégration de cette information évite de proposer une explication qui considère indépendamment ces deux caractéristiques. Dans d'autres cas de figure, la connaissance est propre à l'utilisateur, les méthodes intègrent alors l'utilisateur dans la boucle. En reprenant, l'exemple des appartements, considérons un vendeur qui connaît le nom des villes mais pas leur coordonnées géographiques. Pour une maison qui est chère, il va prévilégier l'explication : "la maison est chère car elle est à Paris" plutôt que : "la maison est chère car elle se situe à une latitude de 49° et une longitude de 2°". L'explication choisie est en accord avec sa connaissance. Dans ce dernier cas, on parle de personnalisation de l'explication pour qu'elle soit adaptée à l'utilisateur.

Une bonne explication est définie selon plusieurs critères, dans l'idéal l'explication proposée doit être "bonne" selon tous les critères considérés. Définir une bonne valeur des critères est aussi difficile que définir une bonne explication. Pour cela, une solution est d'intégrer des besoins utilisateur sur les critères étudiés. Dans ce cas, on considère un second niveau de personnalisation qui ne se limite pas aux connaissances utilisateur mais qui est lié aux besoins utilisateur.

Ainsi, pour proposer une explication personnalisée deux solutions sont possibles, par intégration de connaissances et besoins utilisateur. Cependant, ces informations ne sont pas toujours disponibles, il est tout de même nécessaire que l'explication soit adaptée à l'utilisateur. Une solution à ce problème est de proposer non pas une unique explication mais plusieurs explications à l'utilisateur. On lui laisse alors le choix de sélectionner l'explication qui lui convient le mieux. Par exemple, dans le cas d'un cours, deux cas de figure sont possibles. On considère d'abord le cas où un enseignant effectue le cours individuellement avec chaque élève, il a alors des informations sur le type

d'élève, son niveau, ses connaissances, etc. A chaque cours, l'enseignant adapte ses explications pour qu'elles soient personnalisées à l'élève. Dans le cas plus courant où le professeur enseigne à plusieurs élèves en même temps, il va alors proposer plusieurs explications pour que chaque élève choisisse celle qu'il comprend. La personnalisation ici est plus implicite, mais elle permet de proposer une explication adaptée à l'utilisateur.

1.3 Contributions

Dans cette thèse, nous cherchons à expliquer la prédiction d'un modèle de classification pour une instance particulière, nous étudions ainsi les méthodes locales post-hoc. Nous ne considérons aucune connaissance sur les modèles ou les données, nous nous plaçons dans le cadre particulier d'agnosticité vis-à-vis du modèle et des données. Particulièrement, notre étude se concentre sur la génération d'explications personnalisées.

Notre première contribution s'intéresse à l'intégration de connaissances dans les méthodes d'explications post-hoc dans le but de proposer une explication personnalisée adaptée à l'utilisateur. Nous proposons un formalisme général qui intègre des connaissances utilisateur pour générer une explication : il mesure d'une part la qualité d'une explication par rapport au modèle, et d'autre part sa compatibilité aux connaissances utilisateur. Le formalisme général proposé a pour but d'intégrer la connaissance utilisateur pour tout type de connaissances et tout type d'explication.

Notre seconde contribution instancie ce formalisme général dans le cadre d'explications sous forme d'exemples contre-factuels et de connaissances sous forme d'ensembles d'attributs. Pour résoudre ce nouveau problème d'optimisation, nous proposons une méthode nommée *Knowledge Integration in Counterfactual Explanations* (KICE). Cette méthode génère une explication qui réalise un compromis entre sa proximité à l'instance étudiée et sa compatibilité aux connaissances utilisateur . Nous évaluons cette méthode à travers des expérimentations.

Notre troisième contribution se concentre sur d'autres instanciations du formalisme général, elles considèrent des variantes de la seconde contribution. La première considère un autre type d'explication, les vecteurs d'importance d'attributs, en gardant la même connaissance utilisateur sous forme d'un ensemble d'attributs. La seconde instanciation considère quant à elle le même type de connaissances, c'est-à-dire des explications contre-factuelles, mais intègre un autre type de connaissances sous forme de règles expertes. Nous proposons deux méthodes pour résoudre ces problèmes d'optimisation. La première est appelée *Knowledge Integration in Surrogate Models*, KISM, et la seconde *Rule Knowledge Integration in Counterfactual Explanations*, rKICE. Ces méthodes sont évaluées en effectuant différentes expérimentations.

Les méthodes présentées dans la première partie de la thèse proposent d'effectuer un compromis entre la qualité de l'explication selon le modèle et la compatibilité de celle-ci selon les connaissances. Cependant, définir un bon compromis est difficile. Notre quatrième contribution se concentre sur une nouvelle agrégation des critères qui intègre des besoins sur chacun des critères. Pour cela, nous présentons les propriétés que

l'agrégation doit vérifier dans le but de proposer l'explication la plus adaptée. Parmi les opérateurs existants, nous étudions l'intégrale de Gödel qui satisfait ces propriétés et discutons de son utilisation dans le cas de l'IA explicable. Nous proposons une nouvelle méthode appelée *Gödel Integrals for Counterfactual Explanations*, GICE qui résout ce nouveau problème d'optimisation et comme précédemment nous effectuons une étude expérimentale sur des données de référence pour évaluer les explications obtenues.

Les travaux présentés ci-dessus se concentrent sur la génération d'une unique explication adaptée à l'utilisateur. La dernière contribution s'intéresse aux risques de se restreindre à une seule explication si aucune information sur l'utilisateur n'est connue et discute de la génération de plusieurs explications contre-factuelles, en particulier des exemples divers. Cette notion de diversité est définie de nombreuses manières, nous présentons une typologie des types de diversité et les méthodes de l'état de l'art utilisant cette notion.

1.4 Structure

Le chapitre 2 présente l'état de l'art associé à cette thèse, particulièrement les explications post-hoc et les connaissances utilisateur . Le chapitre 3 présente le formalisme général que nous proposons pour l'intégration de connaissances pour générer des explications post-hoc. Les chapitres 4 et 5 étudient la notion d'incompatibilité définie dans le chapitre 3. Le chapitre 4 se concentre sur les explications contre-factuelles avec intégration de connaissances sous forme d'ensemble d'attributs. Le chapitre 5 étudie l'incompatibilité pour deux types d'explications différentes et différents types de connaissances. Le chapitre 6 se concentre sur l'agrégation de la pénalité et de l'incompatibilité. Enfin, le chapitre 7 discute de la question de la génération d'explications contre-factuelles diverses. Nous terminons cette thèse en résumant nos travaux et en présentant nos perspectives pour des travaux futurs.

1.5 Publications

Les travaux menés dans le cadre de la thèse ont donné lieu aux publications suivantes :

Adulam Jeyasothy, Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. Integrating prior knowledge in post-hoc explanations. *Proceedings of Information Processing and Management of Uncertainty in Knowledge-Based Systems*, IPMU, 2022

Adulam Jeyasothy, Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. Intégration de connaissances dans les méthodes d'explications post-hoc. *Rencontres francophones sur la logique floue et ses applications*, LFA, 2022

Adulam Jeyasothy, Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. A general framework for personalising post hoc explanations through user knowledge integration. *International Journal of Approximate Reasoning*, IJAR, 160 : 108944, 2023

Adulam Jeyasothy, Agnès Rico, Marie-Jeanne Lesot, Christophe Marsala, and Thibault Laugel. Knowledge Integration in XAI with Gödel Integrals. In *IEEE International Conference on Fuzzy Systems*, Fuzz-IEEE, 2023

Adulam Jeyasothy, Agnès Rico, Marie-Jeanne Lesot, Christophe Marsala, and Thibault Laugel. Intégration de connaissances en XAI avec les intégrales de Gödel. *Rencontres francophones sur la logique floue et ses applications*, LFA, 2023

Thibault Laugel, Adulam Jeyasothy, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. Achieving diversity in counterfactual explanations : A review and discussion. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '23, page 1859–1869. Association for Computing Machinery, 2023

Chapitre 2

État de l'art

Dans un contexte où les méthodes d'intelligence artificielle sont de plus en plus utilisées comme outil pour aider divers utilisateurs à accomplir de nombreuses tâches variées, le domaine de l'IA explicable vient augmenter ces méthodes en enrichissant leurs résultats par une *explication*. Il existe de très nombreuses définitions de cette notion d'explication, qui peut prendre de nombreuses formes et être générées par une multitude de méthodes variées : un grand nombre d'articles, comme par exemple (Adadi and Berrada, 2018; Guidotti, 2022; Das and Rad, 2020; Bodria et al., 2021; Burkart and Huber, 2021), en proposent des catégorisations et structurations qui ne sont ni en accord les unes avec les autres sur les regroupements des méthodes existantes ni sur les axes permettant ces organisations, ce qui illustre combien le domaine est vaste.

Les méthodes existantes se distinguent notamment sur ce qu'elles souhaitent expliquer, les hypothèses, parfois implicites, faites sur l'utilisateur qui reçoit l'explication ou les raisons pour lesquelles l'explication est demandée. Outre les formes choisies pour exprimer les explications, elles se différencient également selon des caractéristiques techniques détaillées dans ce chapitre, comme les hypothèses d'agnosticité, la distinction post-hoc vs. ad-hoc ou encore locale vs. globale. Ainsi, certaines méthodes font l'hypothèse que des connaissances sur le modèle d'IA à expliquer sont disponibles, d'autres exploitent des connaissances sur les données alors que d'autres encore adoptent une posture agnostique. Plus récemment, de nouvelles méthodes intègrent également *des connaissances utilisateur* qui donnent des indications sur ce que l'utilisateur destinataire de l'explication connaît sur le domaine dans lequel l'explication est générée. Elles peuvent se présenter sous de nombreuses formes, comme des ensembles d'attributs, des liens entre les attributs ou encore des systèmes de règles. Ces méthodes ont pour but de proposer une explication enrichie, qui soit personnalisée pour chaque utilisateur. Nos travaux de thèse se placent dans ce cadre : ils visent à générer des explications personnalisées.

Après avoir brièvement discuté de la notion d'explicabilité, ce chapitre présente les motivations du domaine de l'IA explicable dans la section 2.2. Puis, la section 2.3 présente les caractéristiques distinguant les méthodes de l'état de l'art. Ensuite, la section 2.4 décrit différentes formes d'explications qui ont été proposées. Les sections 2.5 et 2.6 décrivent plus en détail les deux formes d'explication étudiées dans cette thèse : les

exemples contre-factuels et les vecteurs d'importance d'attributs. La section 2.7 considère la question de l'intégration de connaissances utilisateur : elle introduit les types de connaissances utilisateur existantes et étudie les méthodes d'IA explicable qui les intègrent.

2.1 Notions d'explications

La notion d'explication est très complexe, il existe une multitude de façons de la définir et les recherches sur cette notion ne se limitent pas au domaine de l'IA explicable. Dans cette section, tout d'abord, nous présentons quelques caractéristiques de la notion d'explications qui ont été proposées dans les sciences sociales. Puis, nous décrivons les différents termes utilisés dans l'IA explicable pour nommer la notion d'explicabilité. Enfin, nous présentons les trois questions primordiales exprimées par Barredo Arrieta et al., 2020 pour l'étude de toute recherche en explicabilité.

2.1.1 Les explications en sciences sociales

Le domaine de l'explicabilité n'est pas limité aux informaticiens, la notion d'explication touche de nombreux domaines. Elle soulève des questions sur les notions de compréhension ou de procédure d'explication, qui sont particulièrement étudiées dans les sciences cognitives (Srinivasan and Chander, 2020). En psychologie, les articles de Karsenty, 1996 et Simon, 1992 par exemple montrent à quel point définir la notion d'explication est complexe. Comme le souligne Karsenty, la notion d'explicabilité fait intervenir plusieurs processus cognitifs différents : la généralisation, la particularisation, le raisonnement contrastif et la justification. Ces processus font notamment référence aux différents termes pour désigner l'explicabilité, par exemple l'interprétabilité, la justification ou encore la transparence comme discuté dans la sous-section suivante. De plus, Karsenty défend qu'une bonne explication s'appuie sur le contexte dans lequel elle est fournie pour être compréhensible. Ceci est cohérent avec la problématique considérée qui consiste à intégrer une connaissance de l'utilisateur pour proposer une explication adaptée. Ainsi, les questions soulevées en psychologie sont très similaires à celles qui se posent en IA explicable, fournissant une riche base de références.

C'est pourquoi Miller, 2019 propose de faire des liens entre les notions d'explicabilité dans les sciences cognitives en établissant notamment quatre caractéristiques nécessaires : la contrastivité, la sélection, la causalité et l'intéraction. Une explication doit être contrastive, c'est-à-dire qu'elle doit expliquer pour quelle raison une action se produit plutôt qu'une autre. L'utilisateur doit sélectionner une explication parmi plusieurs possibilités, pour choisir celle qui lui convient le mieux. Une explication doit se baser sur la notion de causalité plutôt que de la notion de probabilité. Enfin, une explication consiste à transférer des connaissances, par un processus interactif entre celui qui fournit et celui qui reçoit l'explication.

2.1.2 Le terme d'explicabilité

De nombreux termes variés sont utilisés pour définir les objectifs de l'IA explicable comme l'interprétablité, la transparence, la justification, la correction ou encore l'explicabilité : Vilone and Longo, 2021 recensent près de 36 termes utilisés par les articles de l'état de l'art pour désigner cette notion dans différents cas de figure. Chacun de ces termes est associé à un but différent de l'explication. Par exemple, la correction fait référence à la capacité de l'explication à fournir des éléments permettant de corriger le modèle étudié. L'interprétabilité quant à elle désigne la capacité de l'explication à définir un concept abstrait. Bien que chaque terme soit associé à des concepts différents, Vilone and Longo, 2021 proposent de regrouper ces termes selon quatre catégories. La première s'intéresse à justifier les raisons pour lesquelles le modèle est utilisable. La seconde s'intéresse à contrôler le modèle en fournissant des éléments qui expliquent son fonctionnement. La troisième catégorie a un but pédagogique qui consiste à découvrir de nouvelles connaissances. Enfin, la dernière catégorie a pour but d'améliorer la performance du modèle considéré. Dans la suite de la thèse, nous choisissons le terme générique d'explicabilité pour désigner l'ensemble des facettes.

2.1.3 Les explications en intelligence artificielle

Il existe une multitude de manière de définir une bonne explication dans le domaine de l'IA explicable. Elles dépendent de trois questions préliminaires qu'un concepteur de méthode d'IA explicable doit se poser (Barredo Arrieta et al., 2020) : qui ?, quoi ? et pour quoi ?. La première demande *qui* souhaite une explication. La réponse à cette question permet de déterminer le type d'utilisateur considéré : un expert du domaine ou un non-expert qui n'a aucune connaissance technique. A titre illustratif, considérons le cas d'une méthode d'IA qui permet de prédire le prix d'un bien immobilier, les explications données au client qui achète une maison, au client qui vend une maison ou à l'agent immobilier ne sont pas les mêmes car leurs connaissances et leurs attentes sont différentes. Cette question est rarement étudiée par les méthodes de l'état d'art qui génèrent des explications, très souvent les méthodes font l'hypothèse implicite que tous les utilisateurs sont similaires. Dans les études d'applications concrètes, un type d'utilisateur avec un profil précis est souvent choisi, ce qui évite de différencier les utilisateurs.

Dans un second temps, il est important de répondre à la question *quoi*, c'est-à-dire ce qu'on souhaite expliquer. La génération d'explications peut concerner des tâches d'IA différentes, par exemple cela peut s'appliquer à la planification, l'allocation de ressources, la recommandation ou encore l'IA générative. Comme dit dans l'introduction, nous considérons dans cette thèse particulièrement le cas de l'apprentissage supervisé et plus précisément la classification. Dans ce cadre, la question de ce qu'on souhaite expliquer est à nouveau à considérer : certaines méthodes cherchent à expliquer le fonctionnement du modèle pour que l'utilisateur comprenne comment le modèle a pu obtenir le résultat qu'il fournit. D'autres méthodes comme Ribeiro et al., 2016 se concentrent sur la frontière de décision et, par exemple, expliquent des zones qui sont associées à

différentes classes. Enfin, d'autres méthodes comme Guidotti et al., 2019 s'intéressent à la classe prédite par un modèle déjà entraîné. Dans cette thèse nous nous concentrons particulièrement sur ce troisième cas, l'explication de prédictions.

Une dernière question concerne le but poursuivi par un utilisateur et peut être formulée comme : *pour quoi* l'utilisateur souhaite une explication ?. L'objectif est alors d'identifier précisément le besoin de l'utilisateur, pour lui fournir une explication adaptée. La réponse à cette question renseigne principalement sur la motivation de l'utilisation de l'IA explicable, ces motivations sont détaillées dans la section suivante.

2.2 Domaine d'IA explicable : motivations

Ces dernières années, le défi principal relevé par les méthodes d'apprentissage automatique a porté sur la création de modèles d'apprentissage performants, c'est-à-dire dans le cas de classification, qui obtiennent des précisions élevées. Le domaine de l'IA explicable est utilisé pour enrichir ces modèles en proposant, en complément, une explication : le but est de proposer à la fois un modèle d'IA qui automatise rapidement et précisément des tâches, ainsi qu'une explication qui accompagne celui qui utilise le modèle. Cette explication permet de ne pas laisser l'utilisateur face à une boîte noire qui représente le modèle, mais à l'aider à mieux comprendre son fonctionnement.

On peut identifier derrière ce besoin d'explication différentes motivations (Gerlings et al., 2021; Barredo Arrieta et al., 2020) qui répondent à plusieurs problématiques. Parmi elles, nous présentons dans cette section quatre problèmes rencontrés lors de l'utilisation des modèles d'IA qui motivent la génération d'une explication. Le premier problème est soulevé par la complexité du modèle, qui peut augmenter le manque de confiance de l'utilisateur. Le deuxième problème est associé au domaine légal qui impose un droit d'explicabilité par l'intermédiaire de nouvelles réglementations. La troisième problématique fait référence à d'éventuelles interprétations énoncées par l'utilisateur quand seul le résultat de la prédiction est fourni, qui peut conduire à des conceptions erronées. Enfin, le dernier problème est dû à des biais éventuels du modèle d'approche automatique.

2.2.1 Confiance de l'utilisateur dans le modèle

La confiance de l'utilisateur vis-à-vis du modèle est une des principales motivations pour proposer une explication. Un éventuel manque de confiance peut être dû à différentes raisons. Premièrement, les modèles performants sont de plus en plus complexes, leurs fonctionnements ne sont pas transparents. L'utilisateur ne peut alors pas s'assurer que la procédure mise en œuvre par le modèle est correcte, ce qui nuit à la confiance qu'il peut lui accorder. Cette notion de confiance (Chamola et al., 2023; Ferrario and Loi, 2022) varie selon le type d'utilisateur considéré : si celui-ci est le concepteur du modèle, il a une connaissance technique ce qui peut lui permettre de comprendre en partie le modèle alors qu'un utilisateur non expert peut ne rien comprendre au modèle. Dans ce

cas, on peut imaginer que l'utilisateur sans connaissances techniques ait une plus faible confiance vis-à-vis du modèle étant donné qu'il le comprend moins. On considère alors que proposer une explication peut permettre d'augmenter cette confiance.

Une deuxième raison à une faible confiance de l'utilisateur est le risque que le modèle se trompe. Un modèle de classification est rarement parfait. Même si son taux de bonne classification soit élevé, il reste des risques d'erreur. L'utilisateur peut alors se demander s'il fait partie des exceptions qui sont mal prédites. Pour certaines applications, en particulier celles qui ont un enjeu important, ce manque de confiance est problématique. Par exemple, dans le cadre médical, un médecin qui n'a pas confiance vis-à-vis du modèle ne va pas considérer sa prédiction dans son analyse. Étant donné que le résultat peut impacter la vie du patient, il ne peut prendre aucun risque.

Un troisième cas de figure se présente lorsque l'utilisateur est un expert de son domaine et qu'il est capable d'effectuer la prédiction en se basant sur ses propres connaissances. Si le modèle d'apprentissage donne une prédiction différente de celle qu'il s'attend pour une instance donnée, il se peut que ce dernier ne lui fasse pas confiance sur une autre instance. Ce manque de confiance vient du fait qu'un utilisateur ne comprend pas pour quelles raisons le modèle n'obtient pas le même résultat que lui.

Nous avons considéré ici les explications comme un outil possible pour agir sur la confiance, avec des risques éventuels. Il faut noter qu'il existe des méthodes dédiées au développement d'outils d'IA de confiance, sans nécessairement passer par des explications. Elles constituent le domaine de *Trustworthy Artificial Intelligence* (Kaur et al., 2022), qui vise à construire des modèles dignes de confiance, indépendamment de leur interprétabilité. Il est donc important de distinguer les notions d'explicabilité et de confiance, et de les évaluer indépendamment.

2.2.2 Interprétations du modèle

Lorsqu'un modèle d'apprentissage automatique est utilisé, l'utilisateur ne reçoit que le résultat final. Ainsi, il peut l'interpréter de différentes manières (Khosravi et al., 2022), et donc pas nécessairement de la manière dont le modèle a été conçu. Considérons un exemple d'enseignement de la vie quotidienne, on imagine un professeur qui donne des cours de mathématiques. Pour un exercice donné, s'il fournit uniquement les réponses à l'élève, il semble évident que les réponses ne sont pas suffisantes pour comprendre l'exercice. L'élève va alors créer sa propre procédure qui permet d'obtenir les mêmes résultats que le professeur, or cette procédure peut être erronée. L'élève interprète alors le résultat différemment de l'interprétation attendue par le professeur. L'explication permet alors de s'assurer qu'il y a un accord entre celui qui fournit le résultat et celui qui l'obtient. Cet exemple montre que le manque d'informations en plus du résultat peut amener l'utilisateur à imaginer une autre interprétation. Fournir une explication permet à l'utilisateur de mieux comprendre le résultat et donc de l'accompagner vers la compréhension attendue par celui qui fournit le résultat.

2.2.3 Réglementations : RGPD et AI Act

Introduit en 2016 au niveau des instances européennes par le parlement européen, le Réglement Général sur la Protection des Données, ou RGPD[1], constitue un ensemble d'obligations dont l'objectif est de renforcer la protection des données personnelles des utilisateurs. Dans son article 39.I.5, il mentionne un "droit à l'explication" : pour tout modèle d'IA utilisé, des explications doivent être fournies si l'utilisateur le demande (Goodman and Flaxman, 2017). Toutefois ces règles sont très vagues sur la définition d'une explication, ce qui ouvre la voie à diverses interprétations.

De plus, il a été souligné (Poullet, 2021) qu'une telle procédure ne suffit pas à protéger les droits du destinataire de l'explication : les explications peuvent être utilisées pour manipuler l'utilisateur, c'est-à-dire que le client peut être aiguillé vers une opinion différente qui n'est pas dans son avantage. Considérons un exemple d'assurance de voiture où un utilisateur demande une explication du prix prédit par un modèle d'IA. Considérons de plus que deux explications sont possibles : le prix est élevé à cause de l'âge ou le prix est élevé à cause de la couleur de la voiture. La première explication n'est pas utile à l'utilisateur car il ne peut pas modifier son âge, par contre la seconde peut l'aider à diminuer le prix de son assurance en achetant une voiture d'une couleur différente. L'assureur peut préférer fournir la première explication pour imposer le prix au client. Ici, l'assureur utilise la confiance de l'utilisateur évoquée dans la section 2.2.1 pour le manipuler comme il souhaite.

En réponse à ce problème, une nouvelle réglementation nommée AI Act[2] (Commission européenne, 2021) a été mise en place en 2021. Elle a pour but de veiller à ce que les algorithmes d'intelligence artificielle respectent les droits fondamentaux des citoyens. Panigutti et al., 2023 discutent le rôle de l'IA explicable depuis l'apparition de cette nouvelle loi. Ils défendent que le domaine de l'IA explicable soit encore plus utile car l'AI Act demande une plus grande transparence des modèles d'apprentissage et une intégration de l'humain dans la boucle. De plus, cette loi vise à éviter les risques de mauvaises utilisations des algorithmes d'IA, c'est pourquoi elle demande si besoin de fournir les applications pour lesquelles ces modèles sont considérés. Ainsi, cette nouvelle réglementation s'assure que la mise en place d'explication n'a pas pour but d'influencer l'utilisateur : les méthodes d'IA explicable doivent être utilisées pour aider l'utilisateur à avoir une meilleure compréhension du modèle étudié.

2.2.4 Biais

Les modèles d'apprentissage automatique sont utilisés dans de nombreux contextes où les résultats peuvent être biaisés selon des caractéristiques discriminatoires, comme la sélection lors d'entretien (Fulk et al., 2022), le prix d'assurances (Blier-Wong et al., 2021) ou encore l'acceptation d'un crédit immobilier (Sadok et al., 2022). On dit qu'il existe un biais lorsqu'une instance est favorisée par rapport à une autre en raison de

1. https://gdpr-info.eu/
2. https://eur-lex.europa.eu/legal-content/FR/TXT/?uri=CELEX%3A52021PC0206

caractéristiques sensibles comme le genre (Dastin, 2022) ou les origines de l'utilisateur (Vincent, 2018).

Un modèle d'apprentissage est entraîné sur un jeu de données. Celui-ci peut être biaisé par exemple en ayant des données non équilibrées : plus d'hommes que de femmes. Les modèles qui s'appuient sur ces données peuvent reproduire voire amplifier ces biais. L'étude de ces biais est au cœur d'un autre domaine appelé équité ou *fairness* (Mehrabi et al., 2021). Certaines méthodes d'IA explicable de l'état de l'art proposent de ne pas éliminer ces biais mais de les mettre en évidence (Virtanen, 2022). Par exemple, lors d'un entretien un homme peut être favorisé par rapport à une femme car le modèle se base sur des données d'entraînement majoritairement liées à des hommes. Dans ce cas, l'explication proposée peut détecter si la raison de la non-sélection d'une personne est liée à son genre.

2.3 Caractéristiques des méthodes d'IA explicable

Nous considérons ici les méthodes d'IA explicable appliquées à des modèles d'apprentissage supervisé : l'objectif considéré dans cette thèse est d'expliquer la prédiction d'un classifieur défini comme une fonction $f : \mathcal{X} \longrightarrow \mathcal{Y}$ où $\mathcal{X}$ désigne l'espace d'entrée, inclus dans $\mathbb{R}^d$ avec d le nombre d'attributs, et $\mathcal{Y}$ l'espace de sortie qui peut être un ensemble de classes, $\mathcal{Y} = \{0, 1\}$ pour une classification binaire ou un espace continu pour une tâche de régression. Nous notons de façon générique e^* l'explication générée.

Il existe une multitude d'approches d'IA explicable qui diffèrent selon de nombreuses caractéristiques, Guidotti, 2022 ou Verma et al., 2020 proposent des taxonomies pour les structurer. Dans cette section, nous nous focalisons particulièrement sur trois caractéristiques fréquemment utilisées qui représentent les distinctions les plus courantes : la première section distingue les approches qui expliquent la prédiction associée à une instance, dites locales, et celles qui explique l'ensemble du modèle pour n'importe quelle instance, dites globales. La seconde section différencie les approches expliquant la prédiction obtenue d'un modèle entraîné, dites post-hoc, et celles qui génèrent l'explication au fur et à mesure de la construction du modèle, dites ad-hoc. Enfin, la dernière section distingue les méthodes selon des connaissances sur les données et le modèle supposées disponibles pour générer l'explication.

2.3.1 Explications globales ou locales

On peut d'abord distinguer les méthodes d'interprétabilité selon qu'elles proposent des explications globales ou locales. Les méthodes globales expliquent le comportement global du classifieur, elles expliquent alors pour toute instance x la prédiction $f(x)$. L'explication globale e^* ne dépend pas d'une instance particulière mais uniquement du modèle étudié f.

Ainsi, les méthodes BETA (Lakkaraju et al., 2017) et TREPAN (Craven and Shavlik, 1995) répondent à cette tâche en définissant des règles de décision. Considérons par

exemple, dans le cas des appartements, un utilisateur qui souhaite comprendre la procédure adoptée par le modèle pour obtenir les prédictions. En utilisant une méthode comme TREPAN, une explication plausible est : "les appartements de plus de 40 m^2 sont chers et ceux de moins de 40 m^2 ne sont pas chers". Un autre exemple est la méthode Partial Dependence Plot (Friedman, 2001) qui présente l'impact des attributs sur la prédiction. Si on reprend l'exemple des appartements, la méthode peut indiquer que le lien entre la surface et le comportement du modèle est monotone : plus la surface de l'appartement augmente, plus la probabilité que l'appartement soit cher augmente. Ces deux explications sont valables pour toute instance x.

Les approches locales (Ribeiro et al., 2016; Apley and Zhu, 2016) quant à elles expliquent le comportement du classifieur pour une instance en particulier. Ainsi, pour une instance spécifique $x \in \mathcal{X}$, leur objectif est d'expliquer la prédiction associée $f(x)$. L'explication locale e^* dépend du modèle f à expliquer et de l'instance considérée x. Pour l'exemple des appartements, un vendeur souhaite comprendre pour quelles raisons son studio qui se situe dans le 11ème à Paris et qui fait 20 m^2 est considéré comme pas cher. Une explication plausible est que l'appartement se situe à Paris, mais qu'il est trop petit. Cette explication est locale donc elle n'est pas forcément valable pour un autre studio à Paris. Dans cette thèse, nous nous intéressons particulièrement aux approches locales.

Les approches globales et locales se différencient sur ce qu'elles expliquent, elles considèrent des *quoi* ? différentes. Pour les premières, l'explication concerne toutes les prédictions du modèle alors que pour les secondes elle concerne une prédiction pour une instance donnée.

2.3.2 Explications post-hoc ou ad-hoc

Une question importante lors de la génération d'une explication est de savoir à quel moment s'effectue la procédure de génération. Une distinction est faite entre les approches, dites *ad-hoc*, qui expliquent le modèle lors de sa construction et celles qui expliquent le résultat du modèle déjà entraîné, nommées *post-hoc*. Les méthodes ad-hoc (Čyras et al., 2021) construisent l'explication au fur et à mesure que le modèle est créé, les deux tâches s'effectuent alors simultanément. Une grande partie des méthodes ad-hoc considèrent des modèles d'apprentissage intrinsèquement interprétables ou possédant des possibilités d'interprétation intrinsèques tels que les arbres de décision ou les modèles linéaires. En effet, un arbre de décision de petite taille peut s'expliquer lui-même, par l'ensemble des règles qui le constituent et peut être vu comme lisible. Les méthodes post-hoc (Lundberg and Lee, 2017; Laugel et al., 2018a; Lash et al., 2017) visent à expliquer un classifieur déjà entraîné. Dans ce cas, le classifieur et la méthode d'interprétabilité sont distincts. Deux exemples, basés sur les exemples contre-factuels, et les vecteurs d'importances locales, sont décrits dans les sections 4 et 5.

Un avantage des approches ad-hoc est d'utiliser des connaissances sur le classifieur pour générer les explications ce qui permet par exemple de garantir un alignement entre

le principe de prédiction et l'explication fournie (Rudin, 2019). Par contre, un inconvénient des méthodes ad-hoc qui proposent des modèles interprétables est qu'elles ont une tâche supplémentaire qui est de proposer un modèle performant en plus de l'explication, il peut alors y avoir un compromis entre la notion de performance et d'explicabilité. Les méthodes post-hoc quant à elles peuvent être appliquées à tout modèle d'apprentissage, dans une étape ultérieure, ce qui les rend plus générales. Nous étudions particulièrement le cas des modèles post-hoc dans cette thèse.

2.3.3 Connaissances sur les données et le classifieur

Pour proposer une explication, il faut se demander quelles informations sont disponibles pour expliquer le modèle. Les méthodes se distinguent selon les paramètres pris en entrée pour générer des explications. En particulier, on les catégorise selon les hypothèses qu'elles font sur le modèle étudié ou les données d'entraînement. Une première information est le type de modèle considéré qui peut aider à avoir des informations sur la frontière de décision ou sur la procédure mise en œuvre par le modèle pour classifier. Par exemple, les frontières de décision des régressions linéaires ou les arbres de décision peuvent être caractérisées facilement. Les méthodes dites *model-agnostic* ne considèrent aucune connaissance sur le modèle. Au contraire, certaines méthodes (Guo et al., 2018; Artelt and Hammer, 2020) utilisent des caractéristiques propres au modèle dans la génération de l'explication ou pour prévoir la forme de la frontière de décision. Par exemple, TreeSHAP (Lundberg et al., 2018) propose uniquement des explications dans le cadre des modèles à base d'arbres de décision (ex : XGBoost, Random Forest, etc.), car l'utilisation des caractéristiques propres à ces modèles permet de générer une meilleure explication. De même, Artelt and Hammer, 2020 considèrent des types de classifieurs spécifiques, des arbres de décision ou des régressions linéaires, car il est plus simple de prévoir les frontières de décision associées à ces modèles, ce qui permet d'obtenir plus facilement l'explication.

La deuxième information porte sur les données traitées : certaines méthodes font l'hypothèse que des données d'apprentissage ou d'autres données qui suivent la même distribution sont disponibles. Les méthodes, dites *data-agnostic*, sont les approches ne considérant aucune connaissance sur les données. D'autres (Poyiadzi et al., 2020; Artelt and Hammer, 2020; Laugel et al., 2020) proposent d'intégrer des connaissances sur les données, notamment pour proposer une explication dite réaliste où le réalisme est défini par sa cohérence aux caractéristiques du domaine étudié. Une manière d'obtenir ces informations est de s'appuyer sur des données. Poyiadzi et al., 2020 et Artelt and Hammer, 2020 considèrent une fonction de densité apprise sur des données. Dans le premier cas, la densité est apprise sur toutes les données alors que la seconde méthode apprend la densité sur les données d'une classe en particulier. Ainsi la seconde approche considère une connaissance plus précise et fait l'hypothèse que des données étiquetées sont disponibles. La méthode de Laugel et al., 2020 quant à elle considère les données d'entraînement pour s'assurer que l'explication générée est proche d'une instance prédite de

la même classe. Il existe également de nombreuses approches post-hoc (Guidotti, 2022) qui reposent sur une étape d'échantillonnage d'instances dans l'espace des données lors de la génération d'explication : une information sur les données peut aider à savoir comment échantillonner l'espace. Il est possible de remplacer l'étape d'échantillonnage par l'utilisation directe des données disponibles.

Disposer d'informations sur le modèle ou sur les données peut aider à générer une meilleure explication plus rapidement et plus facilement. Cependant, ces informations peuvent être difficiles à obtenir. Pour le type de modèle, cela dépend de la personne qui demande l'explication. Par exemple, si l'explication est demandée par l'informaticien qui a conçu le modèle, il dispose de cette information. Par contre, si l'explication est demandée par un informaticien d'une autre entreprise, il n'aura pas forcément accès au détail du modèle et doit mettre en œuvre une méthode d'explication model agnostic, qui s'applique à tout type de classifieurs. Un autre exemple où il est risqué de proposer une méthode qui dépend du type de modèle considéré est lorsqu'un informaticien veut réentrainer son modèle ou qu'il veut le modifier, car les caractéristiques utilisées par la méthode d'explicabilité peuvent changer et donc la méthode peut ne plus fonctionner. Pour les connaissances sur les données, cela dépend des domaines choisis. Obtenir des informations sur les données peut demander un effort particulier. Par exemple, dans le cadre médical les données sont très sensibles donc elles sont rarement diffusées.

Les approches qui ne font aucune hypothèse ni sur le classifieur ni sur les données disponibles sont dites *model-agnostic* et *data-agnostic*. Ceci implique qu'il n'est pas possible d'utiliser les caractéristiques du classifieur ou des données pour construire l'explication. Ainsi, l'avantage de ces méthodes est qu'elles peuvent être utilisées dans tout type de domaines pour tout type de classifieur.

2.4 Formes d'explication

Dans la section précédente, nous avons présenté trois caractéristiques des méthodes d'IA explicable qui permettent de comparer les approches. Une autre différence vient de la forme que prend l'explication : elle peut se présenter sous des formes variées qui répondent à différents problèmes, dont par exemple Guidotti, 2022 propose une vue d'ensemble pour les méthodes post-hoc. Le choix du type d'explication dépend du contexte étudié, du type d'utilisateur considéré et surtout de ce que l'utilisateur veut expliquer. Nous proposons dans cette section de présenter quatre types d'explications, et nous abordons la question de leur présentation à l'utilisateur.

2.4.1 Fonction d'influence

Certaines méthodes proposent une explication définie comme une fonction d'influence qui associe à chaque valeur d'un attribut donné, la probabilité d'appartenir à

une certaine classe. Elle répond alors à la question suivante : "Quel impact a chaque attribut sur les prédictions du modèle ?". Souvent, cette forme d'explication est proposée par des méthodes qui expliquent globalement le modèle.

Les méthodes Partial Dependence Plot (Friedman, 2001) ou Accumulated Local Effects (Apley and Zhu, 2016) constituent des exemples de cette famille d'explications, avec une représentation graphique de la fonction d'influence. Ces approches montrent l'effet marginal des attributs sur le résultat prédit par un modèle d'apprentissage. La seconde est plus rapide que la première et permet de tenir compte de la corrélation entre les attributs. Ainsi, elle est vue comme une méthode plus performante que Partial Dependence Plot.

Les fonctions d'influence ont l'avantage d'être intuitives pour l'utilisateur : il est facile de comprendre l'impact d'un attribut sur la prédiction finale. Par contre, ce type d'explication est très sensible à la corrélation entre les attributs. Même si ALE considère les corrélations, interpréter ses résultats pour les attributs fortement corrélés est difficile. Dans ce cas de figure, il est difficile d'extraire indépendamment l'impact de chaque attribut.

2.4.2 Classifieur

Dans certains cas, l'explication proposée est un classifieur. On parle alors d'un classifieur interprétable (boîte blanche), c'est-à-dire un classifieur transparent dont les raisons de la prédiction sont compréhensibles. Ce type d'explication répond à la question suivante : "Quelle procédure est utilisée pour classifier les instances ?". Les modèles transparents (Molnar, 2022) sont souvent des modèles linéaires, des arbres de décision (Craven and Shavlik, 1995) ou encore des règles de décision (Guidotti et al., 2019). Ces types de modèle représentent des procédures de classification classiques qu'un utilisateur peut adopter. Il faut noter que la simplicité de ces modèles n'est pas suffisante, et que leurs paramètres sont également importants : les arbres de décision doivent être de faible profondeur et les règles de décision doivent avoir peu de conditions dans la prémisse de chaque règle pour être interprétables.

Dans un second temps, des explications peuvent être extraites de ces classifieurs. Par exemple, dans le cas des arbres de décision les caractéristiques impactant la prédiction sont facilement obtenues, elles peuvent être extraites à partir des nœuds de l'arbre. Un autre exemple est une des méthodes les plus connues *LORE* (Guidotti et al., 2019) : elle propose d'entraîner un arbre de décision et d'extraire la branche associée à l'instance étudiée, ainsi elle formule l'explication sous la forme d'une règle. Dans le cas des méthodes ad-hoc, l'explication est souvent globale, elle correspond au modèle lui-même. Cependant, dans le cas des méthodes post-hoc un nouveau classifieur plus interprétable est entraîné, celui-ci cherche à imiter au mieux le modèle à expliquer. On parle alors de modèle de substitution, ou *surrogate model*, qui constitue une approximation locale du classifieur à expliquer avec la question de la définition du voisinage (Laugel

et al., 2018b). Par exemple, une des méthodes les plus classiques est LIME (Ribeiro et al., 2016) qui propose des modèles linéaires.

2.4.3 Vecteurs d'importance des attributs

Un autre type d'explication présente l'importance des attributs dans la prédiction par le modèle en associant un poids à chaque attribut. Ces explications répondent à la question : "Quel impact a chaque attribut sur les prédictions ?". Les vecteurs d'importance des attributs répondent à la même question que les fonctions d'influence mais une grande différence entre les deux explications est que la première étudie l'impact d'un attribut pour n'importe quelle valeur alors que la seconde étudie l'impact de chaque valeur de l'attribut.

Différentes sémantiques sont utilisées pour calculer l'importance d'un attribut. Une première approche, appelée *Permutation Feature Importance* (Fisher et al., 2019) permute les valeurs d'un attribut et observe si la prédiction finale est modifiée. Cela permet de mesurer l'impact de chaque attribut sur la prédiction. Une seconde approche *LIME* (Ribeiro et al., 2016) utilise un modèle de substitution sous forme linéaire pour extraire les poids associés à chaque attribut. Ces poids forment un vecteur qui est l'explication finale. Enfin, *SHAP* (Lundberg and Lee, 2017) se base sur la théorie des jeux pour calculer le poids de chaque attribut, elle considère comme LIME un modèle linéaire mais utilise une fonction de coût différente. Dans le chapitre 5, nous considérons comme LIME une explication qui est un vecteur d'importance extrait des poids d'un modèle interprétable.

2.4.4 Instance

L'explication peut également être une instance dans l'espace des données. Le plus souvent dans un cadre d'explications locales, pour expliquer la prédiction d'une donnée particulière, trois formes différentes associées à des sémantiques qui répondent à des questions différentes peuvent être distinguées : les prototypes (Kaufmann and Rousseeuw, 1987), les critiques (Kim et al., 2016) ou les exemples contre-factuels (Wachter et al., 2018; Mazzine and Martens, 2021; Stepin et al., 2021). Les prototypes et les critiques répondent à la question suivante : "Pourquoi l'instance x est prédite $f(x)$?" en présentant des données similaires à l'instance étudiée.

Les prototypes sont choisis de manière à représenter la classe étudiée $f(x)$ et doivent être tels que le lien entre les caractéristiques et la prédiction soit évidente. De ce fait, l'utilisateur ne questionne pas la prédiction des prototypes.

Les critiques quant à elles représentent des exceptions appelées *outliers*, des instances qui ne sont pas similaires à la majorité des données. Elles sont souvent utilisées en complément des prototypes pour caractériser les cas rares différents des prototypes. Ainsi, les approches utilisant des exceptions ont pour but d'expliquer les prédictions du modèle qui peuvent surprendre, car leurs caractéristiques ne sont pas similaires aux données qui ont une prédiction attendue.

Enfin les explications contre-factuelles, que nous détaillons ci-dessous dans la section 2.5, représentent une instance qui est à la fois proche de x et prédite différemment de l'instance étudiée. Elles répondent à une autre question qui est : "Que dois-je modifier pour avoir une autre prédiction ?". Dans cette thèse, nous étudions particulièrement ce dernier type d'explication.

2.4.5 Visualisation : XUI

En plus du type d'explication généré, la manière donc celle-ci est présentée est une question très importante. De nombreuses méthodes défendent que proposer une visualisation de l'explication la rend plus compréhensible. Par exemple, les méthodes (Friedman, 2001; Apley and Zhu, 2016) générant des fonctions d'influence présentent l'explication finale sous forme de graphiques ou de diagrammes. D'autres méthodes comme SHAP (Lundberg and Lee, 2017) ou LIME (Ribeiro et al., 2016) ajoutent une présentation visuelle à leurs explications pour qu'elles soient plus compréhensibles.

En plus de ces méthodes d'explicabilité qui proposent des visualisations, des études en *eXplanation User Interfaces* (XUI) se concentrent sur comment présenter une explication à l'utilisateur à travers une interface (Chromik and Butz, 2021). Dans cette thèse, nous n'étudions pas la manière dont l'explication est présentée à l'utilisateur. Nous proposons uniquement pour les jeux de données en deux dimensions une visualisation des explications dans l'espace des données.

2.5 Exemples contre-factuels

Les exemples contre-factuels constituent une forme d'explication très répandue, dans le cadre de l'IA explicable : Verma et al., 2020 et Guidotti, 2022 en présentent plus de 50. C'est la forme d'explication que nous privilégions dans nos travaux, nous la présentons donc en détails.

Les exemples contre-factuels constituent une explication intuitive pour tout type d'utilisateurs. Les approches existantes s'appuient sur des critères de qualité variés pour définir une bonne explication. Dans cette section, nous présentons le principe de base où l'explication contre-factuelle est obtenue par une méthode post-hoc locale.

Tout d'abord, nous présentons le principe d'un raisonnement contre-factuel. Puis, nous formalisons les explications contre-factuelles. Ensuite, nous décrivons les différents critères considérés par les méthodes pour définir une bonne explication. Enfin, nous présentons la méthode Growing Spheres (GS) (Laugel et al., 2018a) sur laquelle nous nous basons pour proposer une nouvelle méthode dans le chapitre 4.

2.5.1 Principe du raisonnement contre-factuel

Le raisonnement contre-factuel consiste à imaginer ce qui peut se produire si la configuration initiale observée est modifiée. Ce principe très utile est mis en œuvre

dans de nombreux domaines comme les études des comportements des enfants (Nyhout and Ganea, 2019) ou la géographie historique (Day, 2010) pour le développement économique. A titre illustratif, dans un cadre historique, un raisonnement contre-factuel consiste à se demander ce qui se passerait si une réalité historique ne s'était pas produite : "Que se serait-il passé si la Première Guerre mondiale n'avait pas eu lieu ?". Ce raisonnement consiste à étudier les conséquences en modifiant la situation de départ.

Une explication contre-factuelle utilise un principe similaire. Toutefois, elle ne s'intéresse pas aux conséquences mais aux causes qui amènent à une situation différente à l'arrivée. Si l'on considère le cadre historique présenté précédemment, une explication contre-factuelle consiste à identifier les actions qui auraient pu être effectuées pour éviter une réalité historique : "Quelles actions auraient dû être menées pour que la Première Guerre mondiale n'ait pas lieu ?".

Nous étudions particulièrement les explications contre-factuelles dans le cas des modèles d'apprentissage automatique. Par exemple, considérons le classifieur f qui prédit si un appartement est cher ou non selon ses caractéristiques et x un appartement spécifique. On suppose que le classifieur f prédit que l'appartement x n'est pas cher. Le raisonnement contre-factuel étudie la modification de $f(x)$ si les caractéristiques de x changent : "L'appartement peut-il prendre de la valeur s'il est mieux isolé ?". L'explication contre-factuelle quant à elle étudie les modifications de x pour que la prédiction $f(x)$ change : "Quelles modifications doivent être effectuées sur l'appartement pour qu'il prenne de la valeur ?". Une explication contre-factuelle plausible est sous la forme : l'appartement doit avoir des fenêtres double vitrage et une cuisine américaine.

2.5.2 Motivations

Dans cette section, nous présentons trois principes propres aux explications contre-factuelles qui motivent le choix de ce type d'explications.

Principe intuitif Comme introduit dans la section précédente, le raisonnement contre-factuel n'est pas limité à l'IA explicable. Ce raisonnement est utilisé dès le plus jeune âge (Nyhout and Ganea, 2019), ce qui le rend très intuitif. Par exemple, dans l'éducation des enfants, le raisonnement contre-factuel est souvent utilisé pour leur apprendre les conséquences des leurs actions et particulièrement pour leur apprendre les tâches à effectuer pour atteindre un objectif. On considère un cas où un enfant n'a pas de cadeau et souhaite en avoir un, il se demande alors ce qu'il doit faire pour l'avoir, une information que lui donne ses parents : "si tu as une bonne note alors tu auras un cadeau". L'enfant comprend alors les actions à réaliser pour avoir ce qu'il souhaite. Des raisonnements comme ceux-là sont utilisés dans une grande partie de l'éducation en particulier pour distinguer des notions comme le bien et le mal chez un enfant.

Le raisonnement contre-factuel est également utilisé dans le cadre de l'enseignement. Considérons l'exemple fictif d'un enfant qui apprend à distinguer les légumes et confond une carotte avec un concombre. Une explication lui permettant de comprendre est : "c'est long comme un concombre mais le légume doit être vert pour que ce soit

un concombre" qui correspond à une formulation de raisonnement contre-factuel : "si le légume était vert, ce serait un concombre". L'enfant peut comprendre la caractéristique supplémentaire qui lui permettrait de dire que c'est un concombre. Ici, ce principe est utilisé pour apprendre de nouvelles connaissances. Étant donné que les raisonnements contre-factuels sont présents dès l'enfance, ce principe d'explication est très intuitif. Ainsi, il est compréhensible par tout le monde aussi bien par des experts du domaine que par des non-experts.

Recours Une des particularités des explications contre-factuelles est qu'elles n'ont pas uniquement un but informatif mais permettent de faire un recours : elles peuvent aider l'utilisateur à avoir une nouvelle prédiction en lui fournissant une liste d'actions à effectuer pour obtenir ce qu'il souhaite. Ainsi, elles mettent en place une interaction avec l'utilisateur, il est alors actif face à l'explication et non passif.

La notion de recours propre aux explications contre-factuelles est un grand avantage mais elle soulève aussi de nombreuses interrogations car il n'est pas facile de traduire la notion de recours en informatique. Nous discutons cette notion dans la section 2.7.4 sous le nom d'actionnabilité.

De plus, proposer une liste d'actions rend l'explication plus concrète que par exemple les vecteurs d'importance des attributs qui associent des poids à chaque attribut. En effet, la sémantique de ces poids n'est pas toujours compréhensible pour un utilisateur. Bien que l'explication finale soit obtenue par une méthode informatique, les actions proposées à l'utilisateur ne reflètent pas des caractéristiques techniques qui peuvent être associées au modèle. De ce fait, l'explication contre-factuelle peut être plus compréhensible pour un utilisateur qui n'a aucune connaissance technique.

Personnalisation Comme indiqué précédemment, une explication contre-factuelle répond principalement à la question : "Quelles modifications dois-je effectuer pour obtenir un autre résultat ?". Cette question peut être personnalisée selon les attentes ou les besoins utilisateur. Par exemple, dans le cas des appartements, considérons maintenant une classification en trois classes : pas cher, abordable et cher. Dans un premier cas, on considère un vendeur qui a un appartement classé abordable, il souhaite savoir : "Quelles modifications doivent être effectuées pour que l'appartement soit prédit comme cher ?". Dans un second cas, on considère un acheteur qui souhaite le même appartement classé abordable, il souhaite savoir : "Quelles modifications doivent être effectuées pour que l'appartement ne soit pas cher ?". Dans les deux cas de figure, l'appartement est classé abordable, par contre les attentes ne sont pas les mêmes de la part des deux utilisateurs, dans le premier cas, l'explication souligne les points forts de l'appartement ; dans le second cas au contraire elle détaille ses inconvénients. La personnalisation de ces questions ajoute une précision sur la classe souhaitée.

2.5.3 Formalisation du principe général

Définition Dans cette section, nous présentons la formalisation des explications contre-factuelles pour des tâches d'apprentissage supervisé, en reprenant les notations classiques, par exemple utilisées par Wachter et al., 2018, Guidotti et al., 2019 ou encore Laugel et al., 2018a. L'objectif est d'expliquer la prédiction d'un modèle d'apprentissage automatique entraîné $f : \mathcal{X} \longrightarrow \mathcal{Y}$ pour une instance donnée $x \in \mathcal{X}$. Cette instance x peut être écrite comme un vecteur de d valeurs : $(x_1, ..., x_d)$ où chaque x_j est la valeur associée à l'attribut a_j. Dans la suite et dans toute la thèse, on fait l'hypothèse que les attributs descriptifs de $\mathcal{X}$ sont des attributs numériques à valeurs réelles. Dans le cas des attributs catégoriels, les notions de distance qui interviennent dans la définition des explications contre-factuelles doivent être adaptées. Une explication contre-factuelle notée e répond à la question : "Quelles modifications dois-je effectuer pour obtenir une autre prédiction ?".

Formalisme Dans le cas de l'explication du modèle f pour l'instance x, cette question revient à demander : "Quelles valeurs d'attributs dois-je modifier pour avoir une autre prédiction ?". L'objectif est alors d'identifier les valeurs x_j à modifier et les nouvelles valeurs x'_j pour qu'en notant $e = (x'_1, ..., x'_d)$, on ait $f(e) \neq f(x)$. L'explication contre-factuelle est alors définie comme l'ensemble des modifications à effectuer pour changer de prédiction, c'est-à-dire que le vecteur $|e - x|$. L'exemple e lui-même est parfois considéré comme l'explication.

Fonction de coût On définit donc l'ensemble des exemples contre-factuels candidats comme l'ensemble des instances telles que le classifieur f considéré prédit une classe différente de celle de l'instance étudiée :

$$\mathcal{E}_{x,f} = \{e \in \mathcal{X} | f(e) \neq f(x)\} \tag{2.1}$$

L'explication contre-factuelle finale qu'on propose est une liste d'actions que l'utilisateur doit effectuer pour avoir une autre prédiction. Habituellement, on considère un contexte où l'on souhaite que ces actions soient aussi peu coûteuses que possible. On définit donc une fonction de coût dépendant de l'instance étudiée notée $cost_x(e)$ qui mesure le coût total de ces actions. L'exemple contre-factuel $e \in \mathcal{X}$ est défini par la résolution du problème d'optimisation suivant :

$$e^* = \underset{e \in \mathcal{X}}{\operatorname{argmin}}\, cost_x(e) \text{ avec } f(e) \neq f(x) \tag{2.2}$$

qui s'écrit également :

$$e^* = \underset{e \in \mathcal{E}_{x,f}}{\operatorname{argmin}}\, cost_x(e)$$

La fonction de coût à minimiser définit la qualité de l'explication contre-factuelle, différentes définitions de cette qualité sont discutées dans la section suivante.

2.5.4 Critères de qualité

Comme vu dans les sciences cognitives, il n'y a pas de consensus sur la définition d'une bonne explication. La formalisation de la qualité d'une explication n'est alors pas simple : il existe de nombreuses définitions (Guidotti, 2022). Celles-ci varient selon les critères pris en compte ainsi que leur agrégation. Nous présentons dans cette section quatre critères les plus courants : tout d'abord, une explication contre-factuelle doit être *valide*, c'est-à-dire qu'elle doit appartenir à la classe souhaitée. Ensuite, l'exemple contre-factuel doit être *proche* pour que les efforts effectués par l'utilisateur soient les plus faibles possibles. Un troisième critère présenté est la *parcimonie* dans le but de présenter une explication compréhensible. Enfin, les exemples contre-factuels doivent être dans une région dense pour assurer leur *actionnabilité*.

Validité Selon le formalisme adopté et présenté précédemment, une explication contre-factuelle est valide si elle est associée à une prédiction différente de l'instance étudiée. Au lieu de seulement considérer le résultat obtenu par le modèle f, certaines approches (Wachter et al., 2018) considèrent une fonction de probabilité qui mesure la probabilité que l'instance appartienne à une classe. Au lieu de considérer une prédiction différente, ces approches maximisent la probabilité que l'exemple contre-factuel soit dans la classe souhaitée et permettent ainsi d'imposer une contrainte de confiance de prédiction. Elles évitent alors le risque de proposer une explication qui a une probabilité faible car la prédiction associée peut être une erreur ou peut changer si le modèle est réentrainé.

Proximité La mesure de qualité la plus classique est la proximité de l'exemple contre-factuel à l'instance étudiée. Plus l'exemple contre-factuel est éloigné dans l'espace des données de l'instance étudiée, plus les efforts à effectuer sont élevés. Il est donc nécessaire d'avoir un exemple proche pour s'assurer que les efforts sont faibles : un bon exemple contre-factuel est une instance similaire à l'instance étudiée.

Cette proximité peut être mesurée par différentes distances, le plus souvent par les normes l_1 (Wachter et al., 2018) ou l_2 (Lash et al., 2017). Certaines approches comme celle d' Artelt and Hammer, 2020 utilisent une distance de Manhattan pondérée ou une combinaison des normes l_1 et l_2 comme Van Looveren and Klaise, 2021. Le critère de proximité se formalisme comme suit :

$$c_{prox} = \| x - e \|$$

avec $\| \cdot \|$ la norme choisie pour définir la proximité.

Parcimonie Une explication ne doit pas être complexe pour être compréhensible par l'utilisateur (Miller, 2019). Pour assurer la compréhension de l'explication, une solution courante est de modifier peu d'attributs, ce qui revient à proposer une liste contenant peu d'actions. Ainsi, il est plus simple pour l'utilisateur de concentrer ses efforts sur le même attribut que sur plusieurs attributs. Le critère de parcimonie (Guidotti et al., 2019) est considéré dans le but de proposer des explications qui modifient peu d'attributs. Il est souvent mesuré par la norme l_0 (Dandl et al., 2020; Laugel et al., 2019) afin de compter le nombre d'attributs modifiés pour obtenir l'explication candidate à partir de l'instance étudiée. Le critère de proximité se formalise alors comme suit :

$$c_{parcimonie} = \| x - e \|_0$$

Densité Une autre propriété des explications est qu'elles soient plausibles, c'est-à-dire que l'exemple contre-factuel ait des caractéristiques qui soient réalistes. Le réalisme d'une explication est défini par sa cohérence par rapport aux caractéristiques du domaine étudié, une explication réaliste n'est alors pas une exception. Par exemple, pour le cas des appartements une explication sous forme d'exemples contre-factuels peut demander d'avoir un appartement avec 6 pièces et une surface de 20 m^2 pour avoir la prédiction souhaitée. Ce cas de figure n'est pas réaliste dans la vie réelle donc c'est une explication qui n'est pas utile.

Pour éviter ce problème, certaines approches comme FACE (Poyiadzi et al., 2020) ou celle d' Artelt and Hammer, 2020 incluent comme critère de qualité la densité pour imposer que l'exemple contre-factuel soit situé dans une zone dense de l'espace des données. L'objectif est d'éviter que l'explication proposée soit une exception, pour cela les méthodes effectuent une estimation de la distribution des données. Ce critère est souvent mesuré par une fonction de densité apprise sur des données réelles. Dans l'approche FACE, Poyiadzi et al., 2020 souhaitent globalement que l'explication soit dans une région dense, la fonction de densité est apprise sur l'ensemble des données. Artelt and Hammer, 2020 considèrent une contrainte plus forte en étudiant la distribution des données appartenant à la classe souhaitée. L'explication finale doit non seulement ne pas être un cas rare mais elle doit également ressembler aux instances dans la classe souhaitée.

Il faut noter que le fait d'intégrer ces contraintes de densité rajoute des contraintes sur les informations considérées en entrée : ces méthodes ne sont pas complètement agnostiques aux données car elles nécessitent une information sur la distribution des données. La méthode FACE considère seulement des données alors que Artelt et Hammer considèrent en plus que les données sont étiquetées.

2.5.5 Génération : exemple de Growing Spheres (GS)

Il existe de nombreuses méthodes générant des explications contre-factuelles (Guidotti, 2022; Verma et al., 2020). Dans cette section, nous détaillons particulièrement l'algorithme Growing Spheres (Laugel et al., 2018a) sur lequel s'appuient les propositions

de la thèse.

L'approche Growing Spheres a pour but de générer une explication proche de l'instance étudiée et parcimonieuse, c'est-à-dire qu'elle instancie le problème (2.2) en intégrant les contraintes de proximité et de parcimonie. La proximité de l'explication est définie par une distance euclidienne entre l'instance étudiée et l'explication candidate. Étant donné un classifieur f et une instance x, Growing Spheres cherche alors à résoudre le problème suivant :

$$e^* = \underset{e \in \mathcal{X}}{\operatorname{argmin}} \parallel x - e \parallel_2 + \parallel x - e \parallel_0 \quad \text{avec} \quad f(x) \neq f(e)$$

Pour résoudre ce problème, Growing Spheres utilise une heuristique qui consiste à décomposer le problème en deux étapes. Tout d'abord, la méthode s'intéresse à la recherche d'une explication proche de l'instance étudiée. La méthode résout alors le problème d'optimisation suivant :

$$\tilde{e} = \underset{e \in \mathcal{X}}{\operatorname{argmin}} \parallel x - e \parallel_2 \quad \text{avec} \quad f(x) \neq f(e)$$

Ce problème d'optimisation est résolu en générant à chaque étape des instances dans des boules ouvertes de plus en plus grandes autour de l'instance étudiée, jusqu'à ce qu'une instance d'une autre classe soit trouvée. Le rayon de la première boule est définie par un paramètre ν_0, puis à chaque étape le rayon de la boule est augmenté d'une valeur ϵ. Dans le chapitre 4, nous reprenons ce principe pour la méthode que nous proposons, KICE.

La deuxième étape de l'algorithme Growing Spheres a pour but de rendre l'explication obtenue $\tilde{e}$ la plus parcimonieuse possible. Pour cela, elle propose de projeter $\tilde{e}$ sur un hyperplan où certaines dimensions sont fixées par les valeurs du vecteur x. Elle définit l'ensemble des hyperplans $\mathcal{H}_i : \mathcal{H}_i = \{z \in \mathcal{X} | z_i = x_i\}$, puis étudie l'ensemble des projections sur chacun des hyperplans. L'explication finale est la plus parcimonieuse, c'est-à-dire celle qui minimise la norme l_0. Elle résout alors lors de cette seconde étape un nouveau problème d'optimisation :

$$e^* = \underset{e \in \mathcal{P}_{\tilde{e}}}{\operatorname{argmin}} \parallel x - e \parallel_0$$

avec $\mathcal{P}_{\tilde{e}}$ l'ensemble des projections sur les hyperplans $\mathcal{H}_i$ de l'explication $\tilde{e}$.

2.6 Modèles de substitution

Nous nous intéressons dans cette section à une seconde forme d'explication courante, les modèles de substitution, ou *surrogate models*. Elles sont utilisées dans de nombreux domaines, par exemple pour expliquer la détection de structure moléculaire en chimie (Gandhi and White, 2022) ou encore la prédiction du comportement d'un agent

autonome en robotique (Gavriilidis et al., 2023). Contrairement aux explications contre-factuelles, où le résultat est un exemple, les approches par substitution expriment l'explication sous la forme d'un modèle de prédiction simple, qui constitue une approximation de f. Dans cette section, nous présentons le principe de base où le modèle de substitution est obtenu par une méthode post-hoc locale.

Tout d'abord, nous présentons le principe des modèles de substitution. Puis, nous décrivons la formalisation de cette approche. Enfin, nous présentons la méthode Local Importance Model-agnostic Explanation (LIME) sur laquelle nous nous basons pour proposer une nouvelle méthode dans le chapitre 5.

2.6.1 Principe

Les modèles de substitution ont pour but de faire une approximation du modèle à expliquer par un modèle considéré comme transparent et interprétable. Un modèle interprétable est souvent défini comme les arbres de décision, les règles de décision ou encore la régression linéaire car ceux-ci sont considérés simples à comprendre. Considérer un modèle sous l'une de ces formes n'est pas suffisant pour qu'il soit interprétable, il faut de plus que ses paramètres conduisent à un modèle de complexité faible. Dans le cadre des modèles linéaires, cette complexité est souvent définie par le nombre d'attributs ayant un coefficient différent de 0. Elle fait référence au critère de parcimonie considéré dans le cas des explications contre-factuelles, le modèle est plus parcimonieux lorsque peu d'attributs sont pris en compte. Pour les arbres de décision, la complexité peut être mesurée par la profondeur de l'arbre qui détermine le nombre d'attributs considéré dans chaque branche.

Le modèle de substitution est considéré comme une approximation du modèle étudié. L'explication générée est le modèle de substitution lui-même ou une extraction d'information à partir de ce modèle telle qu'un vecteur d'attributs importants. Les modèles de substitution sont utilisés aussi bien dans le cas d'explications globales comme TREPAN (Craven and Shavlik, 1995), celle de Baehrens et al., 2010 ou celle proposée par Hara and Hayashi, 2016 que pour les explications locales telles que LIME (Ribeiro et al., 2016), LORE (Guidotti et al., 2018) ou LEMNA (Guo et al., 2018).

2.6.2 Formalisation

La procédure de génération d'explication d'un modèle $f : \mathcal{X} \longrightarrow \mathcal{Y}$ est l'apprentissage d'un modèle de substitution $g : \mathcal{X} \longrightarrow \mathcal{Y}$. Cet apprentissage se déroule en trois étapes détaillées ci-dessous. Tout d'abord, la première étape, appelée échantillonnage, consiste à construire un ensemble d'apprentissage $X' = \{(z_i, f(z_i)) | i = 1, ..., n\}$ sur lequel le modèle g est entraîné où n désigne un paramètre à fixer pour fixer la taille de cet ensemble. Ensuite, l'entraînement de g s'effectue sur $\mathcal{X}'$ où la classe associée à chacune des instances est obtenue par le modèle f. Enfin, la dernière étape est l'extraction de l'explication par reformulation plus compréhensible, c'est-à-dire non technique du modèle de substitution g.

Échantillonnage La génération d'un ensemble d'instances $\mathcal{Z} = \{z_i\}_{i \leq n}$ est une étape cruciale. Dans le cas d'une méthode globale, la génération s'effectue dans tout l'ensemble de l'espace des données, les instances z_i couvrent l'intégralité de l'espace de données. Dans le cas d'une méthode locale, les instances z_i sont générées autour de l'instance étudiée x. Le choix de cette zone autour de l'instance étudiée a une influence importante. Une première question est de connaître la forme de la distribution de ces instances. Poyiadzi et al., 2021 visualisent différentes formes de distribution utilisées par les méthodes existantes et montrent les différents impacts qu'elles peuvent avoir. Par exemple, la distribution peut être uniforme comme le propose LEMON (Collaris et al., 2023) ou pas comme le propose LIME (Ribeiro et al., 2016).

Une seconde question est le choix des paramètres de la distribution, notamment la taille de la zone autour de l'instance étudiée. Si cette zone est trop petite, on risque de considérer trop peu d'informations sur le modèle mais si elle est trop élevée, on risque de considérer des informations trop éloignées de l'instance étudiée et donc non pertinentes. Si l'on considère que des informations supplémentaires sont disponibles sur les données, dans un cadre non agnostique, elles peuvent être exploitées pour la génération de l'ensemble $\mathcal{Z}$.

Entraînement du modèle La seconde étape considère le précédent échantillonnage $\mathcal{Z} = \{z_i\}_{i \leq n}$ qui étiquette les nouvelles données, en associant à chaque instance générée z_i, la classe prédite $f(z_i)$. L'objectif de la fonction de substitution g est d'imiter, en le simplifiant, le comportement de f, et non de traiter les vraies classes.

Ainsi, on construit un nouveau jeu de données $X' = \{(z_i, f(z_i)), z_i \in \mathcal{Z}\}$ qui contient l'échantillonnage effectué et les nouvelles classes sur lequel on entraîne le modèle de substitution $g : \mathcal{X} \longrightarrow \mathcal{Y}$ qui dépend du type d'explication souhaité.

Extraction de l'explication Enfin, la dernière étape consiste à extraire une explication à partir du modèle de substitution utilisé. Cette étape peut être vue comme une étape de reformulation d'une explication technique en une explication compréhensible pour l'utilisateur. Comme dit précédemment, l'extraction de l'explication dépend du modèle de substitution considéré. Certaines formes d'explications peuvent être plus facilement extraites du modèle de substitution que d'autres. Par exemple, si g est un modèle linéaire, les coefficients associés à chaque attribut peuvent être interprétés comme des scores d'importance.

Problème d'optimisation Comme on a dit précédemment, un modèle de substitution est un modèle simple qui cherche à faire une approximation du modèle étudié. La notion d'approximation est capturée par une mesure de *fidélité* de l'explication par rapport au modèle notée L. La notion de simplicité du modèle est capturée par Ω qui évalue la complexité d'un modèle, par exemple le nombre de variables prises en compte par le modèle de substitution ou la profondeur dans le cas d'un arbre de décision. Le problème

d'optimisation étudié est le suivant (Ribeiro et al., 2016; Jia et al., 2019) :

$$e^* = \underset{e \in \mathcal{E}}{\operatorname{argmin}}\, L(f, e, \pi_x) + \Omega(e) \tag{2.3}$$

avec $\mathcal{E}$ l'ensemble des modèles de substitution du type considéré et π_x le poids associé à chaque instance de l'ensemble $\mathcal{Z}$ par rapport à la donnée x à expliquer. La fonction de coût est la somme de la fidélité et la complexité, l'explication est donc un compromis entre ces notions. Par exemple, un arbre de décision de profondeur élevée peut être performant, c'est-à-dire fidèle au modèle f, mais complexe et donc peu interprétable. Dans le cas contraire, un arbre de décision de profondeur 1 est très compréhensible mais pas forcément fidèle. Le compromis permet la prise en compte de ces deux notions.

2.6.3 Local Interpretable Model-agnostic Explanations (LIME)

Nous détaillons dans cette section une des approches les plus connues générant des modèles de substitution locaux : LIME de Ribeiro et al., 2016 qui peut être appliqué quand le modèle à expliquer f fournit une fonction de probabilité p_f de chaque classe. Cette fonction de probabilité n'est pas une limitation importante car elle est facilement obtenue par les modèles de classification. LIME considère comme modèles de substitution candidats $e \in \mathcal{E}$ les fonctions linéaires, une explication sous forme de vecteur d'importance des attributs peut alors être obtenue à partir des poids du modèle. LIME est une méthode post-hoc qui génère une explication locale sans aucune connaissance sur le modèle et les données. Nous présentons dans cette section les trois étapes : échantillonnage, entraînement et extraction de l'explication.

Échantillonnage La méthode LIME propose d'utiliser un noyau exponentiel pour définir l'échantillonnage. Ainsi, elle génère des instances qui suivent une loi gaussienne autour de l'instance étudiée. Les instances générées se situent dans tout l'espace des données, mais la concentration est plus forte près de l'instance considérée.

Entraînement LIME considère comme modèle de substitution un modèle linéaire de la forme $e(z) = w_0 + \sum_{i=1}^{d} w_i.z_i$. La méthode choisie pour la fonction de perte qu'on cherche à minimiser est une erreur quadratique moyenne pondérée de l'explication par rapport au modèle à expliquer. LIME associe un poids $\pi_x(z)$ à chaque instance z selon sa proximité à l'instance étudiée selon un noyau gaussien. La mesure de fidélité pénalise moins l'écart de prédiction pour une instance éloignée de l'instance étudiée que pour une instance proche. La mesure de fidélité considérée est alors écrite comme suit :

$$L(f, g, \pi_x) = \sum_{z \in \mathcal{Z}} \pi_x(z)(p_f(z) - e(z))^2$$

La complexité $\Omega(e)$ quant à elle mesure le nombre d'attributs associés au modèle linéaire e, c'est-à-dire ceux qui ont un coefficient non nul. Ainsi, le problème d'optimisation considéré s'écrit comme un problème de régression lasso.

Extraction de l'explication L'explication proposée est un vecteur d'importance des attributs extrait du modèle de substitution utilisé. Ce vecteur représente les coefficients w_i associés à chaque attribut par le modèle linéaire. De plus, pour une meilleure compréhension de l'explication, la méthode propose également une visualisation. Sur cette visualisation, LIME classe tout d'abord les attributs dans l'ordre décroissant des poids $|w_i|$. Puis, elle différencie les attributs qui contribuent positivement, associés un poids positif, et négativement, associés à un poids négatif, à l'appartenance à la classe prédite par le modèle à expliquer $f(x)$.

2.7 Intégration de connaissances additionnelles

Jusqu'ici, les approches d'interprétabilité se placent dans un cadre *user-agnostic*, c'est-à-dire qu'elles n'exploitent aucune connaissance sur l'utilisateur. Le seul choix effectué par l'utilisateur est la forme d'explication qu'il souhaite avoir. Les approches récentes d'interprétabilité proposent de relâcher cette contrainte d'agnosticité en intégrant des connaissances antérieures, ou *prior knowledge*, pour proposer des explications plus adaptées : l'explication est alors réaliste, actionnable ou personnalisée. Cette prise en compte de connaissances a pour but d'intégrer l'humain dans la boucle, se plaçant dans le cadre appelé *human-in-the-loop*, qui vise à ne pas séparer la procédure de génération d'explication de l'utilisateur considéré.

Tout d'abord, cette section présente ce qui motive les approches à intégrer des connaissances additionnelles. Ensuite, nous discutons des diverses formes que peuvent prendre ces dernières. Enfin, nous discutons de la question de leur exploitation et leur intégration dans la génération d'explications pour améliorer la qualité et l'intelligibilité de ces dernières en présentant d'abord les méthodes proposant des explications contrefactuelles réalistes, puis actionnables.

2.7.1 Motivations

Dans la première partie de l'état de l'art, nous avons vu que le domaine de l'IA explicable contient de nombreuses méthodes d'interprétabilité qui enrichissent les modèles d'IA en proposant différentes formes d'explications. Toutefois, comme décrit dans la section 2.1.1, définir une bonne explication est une tâche difficile. Les critères les plus fréquemment utilisés pour définir la qualité incluent la proximité, la parcimonie ou encore la fidélité.

Réalisme Cependant, ces critères ne sont pas suffisants, il faut également considérer la notion de réalisme. Une explication réaliste signifie qu'elle est en accord avec les caractéristiques du domaine étudié. Comme évoqué dans la section 2.5.4, le réalisme peut être mesuré par le biais de la densité des données pour obtenir des explications qui ne sont pas des exceptions. Une autre façon de considérer le réalisme est à travers les liens de causalité. Par exemple, dans le cas de appartements, une explication non réaliste est :

"l'appartement doit contenir plus de chambres mais moins de pièces". Comme dit, dans la section 2.5.4, inclure cette notion nécessite d'intégrer des connaissances et donc de relâcher les hypothèses d'agnosticité.

Actionnabilité Un second critère qui peut être lié à la notion de réalisme est l'actionnabilité. Comme évoqué dans la section 2.5.2, ce critère associé au principe de recours est propre aux explications contre-factuelles. Une explication contre-factuelle non actionnable signifie que les modifications proposées à l'utilisateur ne peuvent pas être effectuées. En reprenant l'exemple des appartements, une explication non actionnable pour un client qui souhaite vendre est : "l'appartement doit être situé dans un autre quartier". Cette action est irréalisable car il est impossible de changer l'emplacement d'un appartement.

Une question commune à l'ensemble des connaissances considérées est de définir leur portée, c'est-à-dire de savoir par qui elles sont définies et à qui elles s'appliquent (Hind, 2019). En effet, elles peuvent être considérées comme des connaissances antérieures, normalement partagées entre plusieurs utilisateurs et qui doivent être prises en compte de manière générale. Les exemples fournis ci-dessus sur le nombre de chambres et de pièces ou sur l'emplacement d'un appartement entrent dans cette catégorie. Ces connaissances communes sont à distinguer des connaissances spécifiques, associées individuellement à une utilisation spécifique, on parle alors de connaissances utilisateur. Ainsi, pour un utilisateur donné, prêt à faire des travaux dans un appartement, une modification sur l'ajout du double vitrage peut être considéré comme actionnable, alors que ce ne sera pas le cas pour un autre utilisateur. L'intégration d'une telle connaissance spécifique permet alors de *personnaliser* le résultat de la génération d'explication, ce qui constitue un avantage certain sur la qualité de l'explication.

Personnalisation La personnalisation de l'explication peut être vue sous différents angles selon l'interprétation qui est faite des connaissances utilisateur : elle peut également être considérée comme un critère indépendant du réalisme et de l'actionnabilité. A titre d'exemple, les attributs indiqués par l'utilisateur peuvent être interprétés au-delà de l'actionnabilité, comme les attributs qui font sens pour lui et qu'il est en mesure de comprendre : il peut alors exprimer une préférence pour une explication qui n'exploite que ces attributs. Ainsi, pour l'exemple des appartements, les attributs *chambres* et *pieces* sont compréhensibles, une modification sur le nombre de chambres ou de pièces est souhaitée par l'utilisateur.

Cette section considère principalement l'intégration de connaissances pour les objectifs de réalisme et d'actionnabilité, la notion de personnalisation est étudiée tout au long de la thèse, dans les chapitres 3 à 6.

2.7.2 Connaissances additionnelles

Une question préliminaire, avant leur intégration dans les méthodes d'explication, concerne les formes que peuvent prendre les connaissances. Plusieurs existent, avec de nombreuses variantes dans chaque cas. On peut par exemple distinguer les connaissances exprimées par des instances, comme des prototypes (Van Looveren and Klaise, 2021), celles sur la distribution des données, comme les fonctions de densité (Poyiadzi et al., 2020) et celles associées aux attributs. Nous considérons ce dernier cas, dans lequel des informations sur les caractéristiques sont fournies : par exemple tous les attributs n'ont pas la même importance ou certains attributs sont liés à d'autres. Nous n'étudions pas la collecte des connaissances, nous considérons directement que les connaissances sont fournies sous une certaine forme. Dans cette section, les deux premières sous-sections présentent des formes de connaissances qui considèrent des informations sur les attributs indépendamment et les deux dernières sous-sections s'intéressent aux liens entre les attributs. Dans cette section, nous notons E les connaissances.

2.7.2.1 Ensembles d'attributs

Un premier type de connaissances fournit des informations sur les attributs individuels : la connaissance indique les attributs dits "actionnables", cela signifie ici ceux qui peuvent être modifiés par opposition à ceux qui doivent rester inchangés (Ustun et al., 2019).

Cette information peut être particulièrement cruciale dans le cas d'une explication exprimée sous forme d'exemples contre-factuels, car elle permet d'éviter de proposer des actions irréalisables. Lorsque cette connaissance est fournie par l'utilisateur, elle ne se restreint pas à l'actionnabilité, elle peut représenter également les attributs connus par l'utilisateur. Dans ce cas, l'explication proposée s'appuie sur des notions connues plutôt que des notions inconnues.

2.7.2.2 Intervalles associés aux attributs

Un second type d'information enrichit l'ensemble des attributs en associant à chacun un intervalle de valeurs (pour un attribut numérique), indiquant des plages de valeurs actionnables. Pour l'exemple des appartements, considérons une connaissance $E = \{pieces = [0,5]\}$, les appartements qui ne vérifient pas cette connaissance sont des cars rares. Ainsi, la prise en compte de cette connaissance permet d'éviter les appartements avec un nombre de pièces élevé. Considérons maintenant une connaissance fournie par l'utilisateur qui représente sa préférence : $E = \{surface = [40,60]\}$, seuls les appartements ayant une surface entre 40 et 60 seront alors proposés à l'utilisateur. Ainsi, les intervalles définissent une restriction de l'espace de données qui évite des exceptions ou d'être confronté à des cas non désirés.

Une forme de connaissances liée porte sur la monotonie des attributs, qui renseigne également sur les variations admissibles des valeurs d'attributs. Elle peut par exemple être exprimée comme "l'attribut ne peut qu'augmenter". Une telle connaissance peut

être formalisée par des intervalles (Mahajan et al., 2019; Lash et al., 2017) : elle peut s'écrire $[x, +\infty[$, c'est-à-dire prendre pour borne inférieure la valeur associée à l'attribut pour la donnée considérée et ne pas prendre de valeurs pour la borne supérieure. Contrairement au cas précédent, dans ce cas, l'intervalle est local, c'est-à-dire qu'il dépend de l'instance étudiée. Comme pour la connaissance précédente, celle-ci peut être commune ou propre à un utilisateur. Par exemple : "l'âge ne peut qu'augmenter" est une vérité générale valable pour tous les utilisateurs. Par contre, "la surface ne peut qu'augmenter" signifie que l'utilisateur souhaite une surface minimale qu'il ne souhaite pas modifier.

2.7.2.3 Interactions entre attributs

Les connaissances examinées ci-dessus portent sur les attributs de manière individuelle. D'autres connaissances fournissent des informations sur les liens entre eux, notamment leurs liens de causalité (Mahajan et al., 2019; Frye et al., 2020). Dans ce cas, la connaissance n'est pas fournie pas l'utilisateur, elle est commune à tous les utilisateurs. Les attributs sont divisés en deux sous-ensembles : les attributs endogènes et les attributs exogènes. La modification des attributs exogènes a alors un impact sur des attributs endogènes et la connaissance explicite ces effets des uns sur les autres.

Les liens entre les attributs peuvent exprimer des contraintes de co-variation. Dans ce cas, en plus d'associer une monotonie à un attribut comme dans la section précédente, la connaissance associe un lien de monotonie entre deux attributs. Par exemple, une connaissance est "augmenter le niveau d'étude implique d'augmenter l'âge".

La causalité n'est pas restreinte à la monotonie, elle peut donner des informations plus riches encore sur les valeurs prises par les attributs directement. Dans ce cas les liens de causalité peuvent être représentés par un graphe de causalité qui exprime les dépendances fonctionnelles entre les valeurs des attributs. Les relations causales distinguent des variables exogènes U et des variables endogènes V. Il existe alors un ensemble de fonctions qui définissent les liens entre eux en se basant sur des caractéristiques du domaine étudié. Une fonction f_v calcule les valeurs d'un attribut V à partir des valeurs des attributs $U = \{u_1, ..., u_k\}$. Une explication est dite non réaliste si la valeur d'un attribut V n'est pas celle attendue en appliquant la fonction de causalité f_v sur les attributs U. Par exemple, dans le cas des appartements il y a un lien de causalité entre les attributs exogènes : latitude et longitude, et l'attribut endogène : code postal.

2.7.2.4 Règles de décision

Masri et al., 2019 proposent un type de connaissances différent, qui prend la forme d'un classifieur et plus précisément d'un système de règles. Un tel système n'indique pas seulement des liens entre les attributs, mais également entre les attributs et les classes. Une règle de décision est composée de deux parties, la prémisse et la conséquence. La prémisse est composée de conditions sur les attributs et la conséquence est

associée à une classe. Ainsi, les instances vérifiant les conditions de la prémisse sont prédites dans la classe associée à la conséquence selon l'utilisateur. De telles informations peuvent être représentées sous une forme ontologique (Bourguin et al., 2021) qui donne également des informations sur les propriétés des classes qu'on peut interpréter comme des règles.

Ce type de connaissances peut être vu comme un classifieur de l'utilisateur (Suryanto and Compton, 2000) qui décrit la procédure qui celui-ci met en œuvre pour classer une instance. On peut dire que cette connaissance est une explication fournie par l'utilisateur. Cette explication n'explique pas la prédiction donnée par le classifieur mais la classe à laquelle l'utilisateur associe une instance. Dans le cas où l'utilisateur classe une instance dans la même classe que le modèle, une explication n'est pas demandée, cette connaissance est une réponse de l'utilisateur pour la question : "Pourquoi le modèle donne cette prédiction ?". L'utilisateur exprime les raisons, pour lesquelles, selon lui la maison est prédite comme chère, cela ne signifie pas que les raisons correspondent à celles du modèle.

2.7.3 Explications réalistes

Un premier type d'explications souhaité sont les exemples réalistes, c'est-à-dire ceux qui est en accord avec les caractéristiques du domaine étudié. Nous détaillons tout d'abord les motivations pour générer une explication réaliste. Puis nous présentons la méthode FACE de Poyiadzi et al., 2020 et celle de Artelt and Hammer, 2020 qui considèrent une fonction de densité et la méthode de Mahajan et al., 2019 qui considère un graphe de causalité pour proposer des explications réalistes.

Définition Un exemple contre-factuel est utile s'il est réaliste, c'est-à-dire que ses caractéristiques respectent les contraintes du domaine. Si celles-ci sont en désaccord avec le domaine, alors l'exemple contre-factuel est considéré comme une exception. On peut distinguer trois façons de mesurer cet accord.

Une première manière est de considérer une fonction de densité $\widehat{p}$ (Poyiadzi et al., 2020; Artelt and Hammer, 2020) entraînée sur un jeu de données qui suit une distribution de données réalistes. Cette fonction indique si une instance est dans une région où il y a de nombreuses données ou non, cela permet de déterminer si l'instance a des caractéristiques cohérentes ou non avec le contexte étudié. Une exception e est une instance dans une région où il y a peu d'instances, elle vérifie alors :

$$\widehat{p}(e) \text{ faible}$$

Une explication est dite réaliste si $\widehat{p}(e)$ est élevée. Selon les contextes étudiés, il n'est pas toujours simple d'avoir un tel jeu de données. Le cas le plus simple est lorsque le contexte considéré n'est pas agnostique aux données, ainsi la fonction de densité est directement entraînée sur le jeu de données. S'il n'y a aucune connaissance sur les données, alors une fonction de densité déjà entraînée est considérée.

Une seconde manière est de considérer un graphe de causalité (Mahajan et al., 2019; Frye et al., 2020) qui définit l'impact des valeurs de certains attributs sur d'autres. La création de ce graphe n'est pas simple, il existe une multitude d'études sur ce sujet (Zanga et al., 2022) qui s'intéressent à l'apprentissage des liens entre les variables. La plupart des méthodes dans le domaine de l'IA explicable considère que le graphe de causalité est connu, notamment les fonctions entre les variables. Une exception selon un graphe de causalité est une instance e qui vérifie :

$$\exists i \in V, e_i \neq f_v(e_{u_1}, ..., e_{u_k})$$

Une version plus expressive enrichit le graphe de causalité par des fonctions de probabilité : dans un réseau bayésien, la fonction qui lie une variable endogène aux variables exogènes donne la probabilité d'obtenir une certaine valeur pour un ensemble de valeurs des attributs exogènes. Une explication est dite non réaliste si la valeur d'un attribut V a une faible probabilité sachant les valeurs des attributs U. Une exception est une instance e qui vérifie :

$$\exists i \in V, f_v(e_i | e_{u_1}, ..., e_{u_k}) \text{ faible}$$

Enfin, des prototypes peuvent définir les caractéristiques du domaine (Van Looveren and Klaise, 2021). Chaque classe est associée à un prototype qui définit les caractéristiques qu'une instance doit vérifier pour appartenir à une classe donnée. Une explication réaliste est similaire au prototype. La cohérence de l'explication avec le domaine est mesurée par la similarité entre l'explication et le prototype.

Densité Les méthodes de Poyiadzi et al., 2020 et Artelt and Hammer, 2020 proposent d'intégrer une fonction de densité mais entraînée sur des ensembles différents car elles considèrent deux buts distincts. La première souhaite une explication qui ne soit pas une exception et donc dans la distribution des données. La seconde émet une contrainte plus forte pour obtenir une explication dans une région dense dans la classe souhaitée, ainsi éviter les explications mal prédites.

La méthode Feasible and Actionable Counterfactual Explanations (FACE) (Poyiadzi et al., 2020) définit un exemple contre-factuel comme une instance dans une région dense telle qu'il existe un chemin $C(x, e)$ entre cette instance et l'instance étudiée passant par des instances appartenant à des régions denses. Ainsi, cette méthode défend le principe que si les étapes de modifications pour atteindre l'explication sont faisables alors l'exemple contre-factuel est réalisable.

Cette méthode s'appuie sur un principe différent des méthodes classiques où l'exemple contre-factuel n'est pas étudié seul mais l'explication est définie par un ensemble d'instances. Les modifications qui permettent à l'utilisateur d'atteindre l'exemple contre-factuel final sont représentées par un chemin $C(x, e) = x_0, ..., x_p$ tel que $x_0 = x$ et $x_p = e$. La fonction de coût ne dépend donc pas seulement de l'exemple candidat mais également du chemin utilisé.

La connaissance est intégrée dans la fonction de coût à travers des contraintes sur les instances du jeu de données. Le problème d'optimisation considéré est alors le suivant (Poyiadzi et al., 2020) :

$$e^* = \underset{e \in \mathcal{E}_{x,f}}{\operatorname{argmin}} \parallel x - e \parallel_2$$

tel que $\widehat{p}(x') \geq \delta$ et $|x' - x''| \leq \epsilon$ pour tout x', x'' les points successifs du chemin $C(x, e)$.

où $\widehat{p}$ désigne la fonction de densité, δ et ϵ deux réels. L'inconvénient de cette méthode est que le choix des seuils a un impact important sur l'exemple contre-factuel obtenu. Lorsque les classes sont bien séparées, il est difficile de passer d'une classe à l'autre si la valeur de ϵ choisi est trop faible ; il est donc possible de ne trouver aucun exemple contre-factuel.

A la différence de la méthode FACE, la méthode d'Artelt and Hammer, 2020 considère une contrainte de densité relative à la classe souhaitée : elle ne considère pas la probabilité globale $\widehat{p}$, mais $\widehat{p}_c$ où $c \in \mathcal{Y} \backslash f(x)$. On note $\widehat{p}_{classe}$ la fonction de densité basée sur les données appartenant à la classe souhaitée. Ainsi, l'exemple contre-factuel obtenu est dans une région dense dans la classe souhaitée. Le problème d'optimisation étudié est le suivant :

$$e^* = \underset{e \in \mathcal{E}_{x,f}}{\operatorname{argmin}} \parallel x - e \parallel_2 \text{ avec } \widehat{p}_{classe}(e) \geq \delta$$

Un inconvénient de la méthode d'Artelt and Hammer, 2020 par rapport aux autre méthodes est qu'elle ne considère pas un contexte model-agnostic, la méthode est utilisée uniquement pour les modèles linéaires et les arbres de décision. Elle a ainsi des connaissances sur le type de modèle et des connaissances partielles sur les données à travers une fonction de densité.

Graphe de causalité En considérant également le cas des explications contre-factuelles, Mahajan et al., 2019 proposent une méthode permettant d'intégrer des relations causales entre les attributs : elle repose sur la définition d'une nouvelle distance entre une explication candidate et l'instance d'origine, celle-ci est élevée si les relations causales ne sont pas satisfaites.

La distance causale entre deux points x et e mesure la différence entre les variables endogènes obtenues à partir des variables exogènes de x et les variables endogènes de e. Cette mesure est une fonction de comparaison et non une distance car elle ne vérifie pas la propriété de symétrie. La fonction de coût étudiée est alors :

$$e^* = \underset{e \in \mathcal{E}_{x,f}}{\operatorname{argmin}} \sum_{u \in U} |x_u - e_u| + \sum_{v \in V} |f_v(x_{u_1}, ..., x_{u_k}) - e_v|$$

On retrouve d'une part la distance l_1 qui se concentre sur la proximité du contre-factuel et d'autre part la distance causale qui pénalise les instances ne vérifiant pas les liens causaux. Un inconvénient de cette méthode est qu'elle considère que toutes les modifications selon les attributs endogènes sont équivalentes. Par exemple, considérons

deux liens de causalité : $u_1 \longrightarrow v_1$ et $u_2 \longrightarrow v_2$. Le premier lien est associé à la fonction identité et le second est associé à la fonction au carré. Ainsi, pour une modification similaire sur les attributs u_1 et u_2, la modification sur l'attribut v_2 est plus élevée que sur l'attribut v_1. Cela est dû au fait que pour un attribut exogène v la distance entre x_v et e_v n'est pas considérée dans la fonction de coût.

2.7.4 Explications actionnables

Cette seconde section s'intéresse à l'actionnabilité des explications, c'est-à-dire que la liste des modifications proposées à l'utilisateur pour obtenir la classe souhaitée doit être réalisable. Tout d'abord, nous présentons une définition de l'actionnabilité. Puis, nous présentons deux méthodes qui ont pour but de proposer des explications actionnables, celle de Ustun et al., 2019 et celle de Mahajan et al., 2019.

Définition Comme dit dans la section 2.5.2, la notion d'actionnabilité est propre aux explications contre-factuelles, étant donné que celles-ci ne fournissent pas une simple information mais aussi des actions à effectuer. Pour que ces explications soient utiles, il faut que les modifications proposées soient réalisables par l'utilisateur ; on dit alors que l'explication contre-factuelle est actionnable. Une action selon un attribut i de l'instance x pour obtenir l'explication e est définie comme suit :

$$A(x, e, i) = e_i - x_i$$

Les actions possibles sont exprimées à trois niveaux d'informations : les attributs qui sont modifiables ou non, le sens de modification des attributs et les intervalles associés à chaque attribut. Premièrement, nous considérons la connaissance qui est un ensemble E contenant les attributs actionnables, c'est-à-dire ceux qui sont modifiables. Une explication e actionnable modifie uniquement les attributs actionnables, c'est-à-dire vérifie :

$$\forall i \notin E, A(x, e, i) = 0$$

Deuxièmement, seulement certaines actions peuvent être effectuées sur un attribut. La valeur de l'attribut peut seulement augmenter ou diminuer. La connaissance requise est la monotonie d'un attribut. La connaissance E comprend deux ensembles : E_+ qui contient les attributs qui peuvent augmenter et E_- qui contient les attributs qui peuvent diminuer. Une explication e actionnable augmente ou diminue les valeurs des attributs seulement si c'est possible, c'est-à-dire que la valeur associée à l'attribut i augmente uniquement si $i \in E_+$. Ainsi, une explication actionnable vérifie :

$$\forall i \in E_+, A(x, e, i) \geq 0 \quad \& \quad \forall i \in E_-, A(x, e, i) \leq 0$$

Enfin, un attribut ne peut pas prendre n'importe quelle valeur, il est associé à un intervalle. Dans ce cas la connaissance considérée associe à chaque attribut i un intervalle $[v^i_{inf}, v^i_{sup}]$. Une explication e actionnable est définie comme :

$$\forall i, e_i \in [v^i_{inf}, v^i_{sup}]$$

Pour proposer une explication actionnable, plusieurs types de connaissances peuvent être intégrés. Ustun et al., 2019 intègrent des actions exprimées par les trois types de connaissances précédents, alors que Mahajan et al., 2019 considèrent la monotonie des attributs pour modifier les attributs dans la direction souhaitée.

Ensemble d'attributs La méthode de Ustun et al., 2019 considère les trois connaissances décrites précédemment. Pour généraliser ces trois types de connaissances, cette méthode considère un ensemble d'actions $a_i = e_i - x_i$ définies comme des vecteurs.

L'intégration de cette connaissance s'effectue à travers la définition de l'espace de recherche. Ustun et al. proposent de restreindre l'ensemble de recherche pour interdire l'utilisation de certains attributs ou directions. En notant E l'ensemble des actions possibles, l'espace de recherche devient :

$$\mathcal{E}'_{x,f,E} = \{e \in \mathbb{R}^d | f(e) \neq f(x) \text{ et } |e - x| \in E\}$$

Cela permet d'éviter de proposer des explications qui demandent d'effectuer des modifications irréalisables.

De plus, Ustun et al. définissent une bonne explication comme une explication faisable c'est-à-dire une explication dans une région dense de la classe souhaitée et qui repose sur des modifications actionnables.

Ainsi, la méthode résout le problème d'optimisation suivant :

$$e^* = \underset{a \in A(x)}{\operatorname{argmin}} cost(a, x) \text{ avec } f(x + a) \neq f(x)$$

avec $A(x)$ l'ensemble des actions réalisables à partir de x. Pour cette fonction de coût, ils donnent quelques exemples notamment une distance normalisée :

$$cost(x + a, x) = max_{j \in J_A} |Q_j(x_j + a_j) - Q_j(x_j)|$$

où Q_j désigne la fonction de répartition de la variable j. Cette distance mesure le déplacement le plus important selon un attribut actionnable. Ustun et al. proposent donc une explication proche effectuant uniquement les actions réalisables.

Monotonie des attributs Une autre méthode qui propose des explications actionnables en considérant comme connaissances le sens de modifications des attributs est la méthode proposée par Mahajan et al., 2019. Contrairement à Ustun et al., 2019 qui proposent de modifier l'espace de recherche, l'intégration de la connaissance s'effectue

en modifiant la fonction de coût. Mahajan et al., 2019 favorisent les exemples contre-factuels qui modifient les attributs considérés dans la connaissance comme souhaité. Ainsi, la fonction de coût est définie pour tout attribut i de E qui ne peut qu'augmenter comme $-\min(0, e_i - x_i)$ et pour tout attribut i de E qui ne peut que diminuer comme $-\min(0, x_i - e_i)$. Si on considère le cas où un attribut i peut seulement augmenter, et que la valeur de l'attribut diminue alors la fonction de coût est négative, mais si l'attribut augmente le coût est nul. Ainsi, toutes les explications qui diminuent cet attribut sont considérées comme les pires cas et celles qui augmentent cet attribut sont équivalentes.

2.8 Bilan

Nous avons présenté dans cette section les caractéristiques du domaine de l'IA explicable en présentant différents types d'explication. Dans notre thèse nous nous plaçons dans un contexte agnostique au modèle et aux données, c'est-à-dire qu'aucune information sur le modèle ou les données ne sont connues. Nous étudions la génération d'explications locales post-hoc ; nous expliquons ainsi la prédiction $f(x)$ d'une instance donnée x. Nous nous intéressons particulièrement à deux types d'explications : les exemples contre-factuels et les vecteurs d'importance des attributs.

Nous avons présenté les différents types de connaissances utilisateur et leur intégration dans les méthodes existantes. Jusqu'ici, nous avons détaillé les méthodes qui proposent des explications réalistes et des explications actionnables. Dans notre thèse, nous nous intéressons particulièrement aux explications personnalisées.

Chapitre 3

Intégration de connaissances pour générer des explications post-hoc

Comme montré dans le chapitre précédent, il existe de nombreuses méthodes d'interprétabilité post-hoc, mais certaines de ces méthodes entraînent des explications non adaptées à l'utilisateur, par exemple non actionnables, irréalistes ou non compréhensibles (cf. section 2.7.2). Dans les trois cas, le risque peut être diagnostiqué comme résultant du manque d'informations considérées en entrée, notamment pour les méthodes qui se placent dans un cadre agnostique, où aucune information sur le modèle ou sur les données n'est connue. Dans ce chapitre, nous nous concentrons sur les explications non adaptées parce que non compréhensibles. Pour répondre à ce problème, nous proposons d'intégrer des connaissances additionnelles de l'utilisateur en considérant des explications propres à chaque utilisateur qui permettent de générer des explications personnalisées. Nous proposons une formalisation générale pour cette intégration de la connaissance utilisateur. Ce cadre générique s'applique à tout type d'explications et tout type de connaissances, il ne fait aucune hypothèse restrictive.

Ce chapitre discute tout d'abord les avantages et les inconvénients de la personnalisation d'une explication. Puis, nous présentons le formalisme général qui consiste à optimiser une fonction de coût, mais en intégrant des connaissances utilisateurs. Les trois sections suivantes décrivent chaque terme de la fonction de coût considérée dans ce formalisme : la fonction de pénalité (section 3.3), la fonction d'incompatibilité (section 3.4) et la fonction d'agrégation (section 3.5). La dernière section 3.6 montre comment l'approche proposée par Ustun et al., 2019 peut s'inscrire dans le formalisme général proposé, les chapitres 4 et 5 présentent d'autres instanciations.

3.1 Personnalisation de l'explication : avantages et inconvénients

Dans la section 2.7, nous avons motivé la prise en compte de connaissances liées au domaine applicatif dans lequel les explications sont générées pour proposer des explications réalistes et actionnables. Dans ce chapitre, nous nous concentrons particulièrement sur l'intégration de connaissances utilisateur. Cette connaissance peut ne pas être commune à un ensemble d'utilisateurs mais propre à chaque utilisateur. L'explication fournie est alors adaptée à l'utilisateur considéré, elle ne convient pas à tout type

d'utilisateurs : on parle d'explication personnalisée. Nous discutons les avantages et les inconvénients de la personnalisation d'une explication.

3.1.1 Avantages

La personnalisation d'une explication a pour but de donner une réponse adaptée à chaque utilisateur indépendamment (Becker, 2023), à plusieurs niveaux. Tout d'abord, elle peut s'adapter à différentes questions : un utilisateur peut vouloir comprendre les raisons d'une prédiction ou vouloir connaître les modifications à effectuer pour avoir ce qu'il souhaite. Le type d'explication choisi pour répondre à ces deux questions peut différer car l'explication peut fournir des informations différentes.

Ensuite, dans le cas des explications contre-factuelles la personnalisation peut se faire selon la classe souhaitée par l'utilisateur. Par exemple, un vendeur demande : "Qu'est-ce-que je dois modifier pour que mon bien prenne de la valeur ?", alors qu'un acheteur se demande : "Quelles attentes je dois modifier pour avoir un bien moins cher ?". Dans ces deux cas, on étudie la génération d'un exemple contre-factuel qui explique la classification des appartements, mais les attentes des deux utilisateurs sont opposées. L'explication fournie au vendeur va alors souligner les points forts du bien considéré alors que les points faibles sont mis en avant pour l'acheter. Dans cette thèse, nous ne nous intéressons pas à ce type de personnalisation et nous considérons que le type d'explications et la classe souhaitée sont fixés.

Un troisième niveau, que nous considérons dans nos travaux, envisage la personnalisation selon les connaissances propres à l'utilisateur, qui diffèrent d'une personne à l'autre. Cette personnalisation se concentre sur la notion de compréhension de l'explication par l'utilisateur. Les connaissances peuvent être exprimées de pleins de facettes différentes, que ça soit par le choix d'attributs ou par un processus de classification qui s'appuie sur des connaissances antérieures de l'utilisateur. Cette diversité de formes de connaissances induit des solutions d'intégration variées. Une autre façon de personnaliser est de répondre à la curiosité de l'utilisateur qui souhaite acquérir de nouvelles connaissances. Dans la section 3.4, nous présentons ces deux points de vue qui répondent à des attentes différentes de l'utilisateur. Dans un premier cas discuté dans la section 3.4.2, l'explication a pour but d'être plus compréhensible en étant dans le langage des connaissances. Dans un second cas discuté dans la section 3.4.3, l'explication a pour but d'apprendre de nouvelles notions à l'utilisateur en étant complémentaire aux connaissances.

3.1.2 Inconvénients

Un inconvénient principal à la personnalisation est que l'ajout de ce second objectif entraîne un risque que le premier objectif, défini par la qualité telle que discuté dans le chapitre 2 et particulièrement la section 2.5.4, ne soit plus atteint. En effet, ces deux objectifs peuvent être contradictoires, ainsi les optimiser simultanément est difficile. Il

est nécessaire de définir dans quelle mesure il est accepté de perdre en qualité de l'explication par rapport au modèle pour avoir une explication compatible avec les connaissances utilisateur.

Un second inconvénient est que la personnalisation peut être utilisée pour manipuler l'utilisateur. Comme l'évoquent Conati et al., 2021 dans le cas de l'enseignement, une des motivations de la personnalisation d'une explication est qu'elle permet de mettre en confiance l'utilisateur, car l'explication se base sur ses connaissances. Or, comme évoqué dans la section 2.2, la confiance de l'utilisateur peut être manipulée. Il est possible de proposer une explication qui s'appuie uniquement sur les connaissances utilisateurs. Par contre, cela ne signifie pas que l'explication représente le modèle. Ainsi, sous couvert de personnalisation, une explication peut être manipulée (Slack et al., 2021; Carli et al., 2022). Il est donc important de distinguer les notions d'explicabilité et de personnalisation, c'est-à-dire qu'il faut mesurer la qualité d'une explication selon ces deux notions indépendamment.

Ces deux inconvénients précédents montrent que la personnalisation d'une explication ne doit pas s'effectuer sans considérer l'accord avec le modèle. Or, la contrainte de fidélité avec le modèle implique qu'une explication totalement en accord avec les connaissances n'existe pas toujours. Il est alors impossible de proposer une explication qui représente le modèle et qui soit totalement en accord avec les connaissances. Une piste de solution qui constitue une perspective à plus long terme, non discutée dans cette thèse, consiste à ne pas se restreindre aux connaissances mais également à proposer à l'utilisateur d'apprendre de nouvelles connaissances indispensables pour la compréhension du modèle.

3.2 Formalisme général

Comme détaillé dans la section 2.3.2, la plupart des méthodes d'explication post-hoc minimisent une fonction de coût pour générer des explications. De la même manière, pour proposer une explication personnalisée, nous proposons de formuler la question de la génération d'une explication en intégrant une connaissance utilisateur, sous la forme d'un problème d'optimisation qui mesure la notion d'explication d'une part et la notion de personnalisation de l'autre. Ainsi, nous proposons un cadre générique qui s'exprime comme une fonction de coût enrichie : cette dernière ajoute au terme de pénalité classique qui évalue la qualité d'une explication candidate par rapport à l'instance x et au classifieur f considérés, un terme d'incompatibilité qui dépend de la connaissance de l'utilisateur E et qui mesure la qualité d'une explication candidate par rapport à la connaissance. Le problème d'optimisation est formellement défini comme suit :

$$e^* = \underset{e \in \mathcal{E}}{\operatorname{argmin}}\, agg(penalty_x(e, f), incompatibility_x(e, E)) \tag{3.1}$$

où $penalty_x$, $incompatibility_x$ et agg sont trois fonctions décrites et discutées successivement dans les trois sections suivantes. Il faut noter qu'elles dépendent du contexte étudié : les motivations de l'utilisateur, le type d'explication à générer et le type de connaissance considéré. Des instanciations en sont discutées et proposées dans la section 3.6 et les chapitres 4 et 5. Pour alléger les notations, dans la suite, nous notons $P_x(e, f)$ la pénalité et $I_x(e, E)$ l'incompatibilité.

La solution du formalisme proposé peut prendre la forme d'une unique explication ou un ensemble d'explications. Dans les chapitres 4 et 5, nous considérons que la solution est une unique explication choisie arbitrairement parmi l'ensemble des explications qui minimisent la fonction de coût. Dans le chapitre 7, nous étudions la génération de plusieurs explications.

3.3 Fonction de pénalité

La fonction de coût des méthodes d'explications telles que discutées dans le chapitre 2 mesure la qualité d'une explication, celle-ci est liée à la notion d'explicabilité du modèle considéré. Comme précisé dans la section 2.5.4, il existe de nombreux critères qui peuvent intervenir dans la définition ; ainsi la qualité est définie de différentes manières selon le type d'explication considéré et la définition choisie. Nous souhaitons minimiser la fonction de coût que nous appelons la fonction de pénalité, elle prend en argument le modèle à expliquer f et l'explication candidate e. Ainsi, elle a pour but de pénaliser les explications qui n'expliquent pas le modèle considéré. Plus la pénalité est faible, meilleure est l'explication candidate e. Elle peut dépendre de l'instance étudiée x dans le cas d'une méthode locale.

Dans cette thèse, nous discutons de cette fonction de pénalité pour deux formes d'explication différentes : les explications contre-factuelles sont étudiées dans le chapitre 4 et la section 5.2, et les vecteurs d'importance des attributs dans la section 5.1. Comme rappelé dans le chapitre 2, dans le premier cas, la qualité d'une explication est souvent définie par la notion de proximité, alors que dans le second cas, elle s'exprime plutôt comme sa fidélité au modèle. Pour ces deux types d'explication, nous nous appuyons sur les définitions données par les méthodes existantes pour mesurer la qualité d'une explication, telles que rappelées dans les sections 2.5 et 2.6.

3.4 Fonction d'incompatibilité

Dans le but d'intégrer des connaissances utilisateur, nous proposons d'ajouter une nouvelle fonction, appelée incompatibilité, qui mesure à quel point l'explication est en accord avec les connaissances considérées. La notion d'incompatibilité peut être comprise de nombreuses manières selon le contexte étudié. Nous discutons dans cette section de deux principes différents qui peuvent être rencontrés dans deux cas distincts : (i) proposer une explication dans le langage des connaissances et (ii) proposer une explication complémentaire aux connaissances. Après avoir présenté un exemple illustratif

fictif et rappelé les notations, nous illustrons ces deux cas et présentons les fonctions d'incompatibilité qui permettent de les implémenter.

3.4.1 Exemple et notations

Tout d'abord, nous présentons l'exemple fictif sur lequel s'appuient les sections suivantes pour décrire les deux principes : le classifieur considéré prédit le type de légumes à partir des caractéristiques telles que la couleur, la forme, le goût et de caractéristiques nutritives comme le taux de provitamine A, de saccharose, de fibres, etc. On étudie un échantillon qui est prédit comme une carotte. Pour expliquer cette prédiction, deux explications sont possibles : la première, notée e_1, est : "La couleur est orange et le goût est sucré". La seconde, notée e_2, est : "L'échantillon a un taux élevé de provitamine A et de saccharose". Nous considérons pour cet exemple deux utilisateurs : un non-expert et un expert du domaine, qui expriment tous les deux la même connaissance $E = \{couleur, goût\}$. Même si la connaissance est la même, nous montrons ci-dessous qu'elle doit être interprétée différemment.

La fonction d'incompatibilité discutée dans les sections suivantes pour illustrer les deux interprétations de la connaissance utilisateur repose sur deux ensembles d'attributs. Le premier noté, A_e, désigne les attributs qui interviennent dans l'explication, sa définition dépend de la forme de l'explication considérée. Ainsi, pour les exemples contre-factuels, A_e désigne les attributs à modifier pour obtenir la classe souhaitée ; pour les modèles de substitution sous forme de modèles linéaires, A_e désigne les attributs dont le coefficient est non nul. Le second ensemble d'attributs, noté A_E, exprime les attributs impliqués dans la connaissance utilisateur. Dans l'exemple précédent, on a directement $A_E = E$ puisque la connaissance utilisateur est exprimée comme un ensemble d'attributs. Pour une connaissance représentée sous la forme d'un graphe de causalité, l'ensemble A_E désigne respectivement l'ensemble des attributs exogènes et les attributs présents dans la prémisse de la règle.

3.4.2 Explication dans le langage des connaissances

Tout d'abord, nous présentons le cas où l'explication doit être en accord avec les connaissances de l'utilisateur, qui correspond au cas de l'utilisateur non-expert. La connaissance est interprétée comme les caractéristiques que l'utilisateur est en mesure de comprendre et qu'il accepte donc de trouver dans une explication : leur présence lui permet de comprendre l'explication, ce qui peut augmenter sa confiance dans le modèle d'apprentissage et sa volonté de l'utiliser. Ainsi, un utilisateur non-expert souhaite une explication dans son langage pour qu'elle soit plus compréhensible. Dans le cas de l'exemple de classification de légumes introduit ci-dessus, il préfère l'explication e_1 : "La couleur est orange et le goût est sucré", étant donné qu'elle contient seulement les attributs présents dans sa connaissance.

La fonction d'incompatibilité vise alors à minimiser le nombre d'attributs intervenant dans l'explication qui ne font pas partie de la connaissance de l'utilisateur :

$$I_x(e, E) = Card(A_e \setminus A_E)$$

Lorsqu'il y a désaccord total entre la connaissance et l'explication, les attributs présents dans l'explication candidate e ne font pas partie de la connaissance E. Cela correspond au cas où $A_e \cap A_E = \emptyset$, qui conduit à une incompatibilité maximale, égale au nombre d'attributs dans l'explication. En revanche, l'incompatibilité est minimale, c'est-à-dire qu'elle vaut 0, lorsque l'ensemble des attributs présents dans l'explication sont des attributs connus par l'utilisateur, c'est-à-dire lorsque $A_e \subseteq A_E$.

3.4.3 Explication complémentaire aux connaissances

Un autre principe considère une définition de l'incompatibilité contraire à la précédente et qui a du sens dans un scénario différent, pour un autre type d'utilisateur avec des objectifs différents. Cette seconde configuration peut être observée dans le cas d'un utilisateur expert : la connaissance qu'il exprime représente les attributs dont il connaît déjà l'impact et qu'il souhaite compléter. Ainsi, il est intéressé par une explication qui est en fait orthogonale, ou complémentaire, à ses connaissances, susceptible de lui fournir de nouvelles informations dont il ne disposait pas. En d'autres termes, son objectif dans ce cas est d'acquérir de nouveaux éléments de connaissance, enrichissants, qui ne doivent pas être redondants avec ce qu'il sait déjà. Cette absence de redondance avec les connaissances qu'il exprime va à l'encontre du principe d'intégration des connaissances du cas précédent.

Pour l'exemple de la classification de légumes introduit ci-dessus, l'utilisateur expert préfère l'explication e_2 : "L'échantillon a un taux élevé de provitamine A et de saccharose", car elle exclut les variables dont il connaît déjà l'impact. Cet exemple illustre le fait qu'avec une même connaissance, deux explications différentes peuvent être privilégiées selon l'interprétation que l'on fait de la connaissance.

Dans le second cas, l'explication proposée doit être complémentaire à la connaissance, c'est-à-dire qu'elle doit minimiser la redondance. Étant donné que nous considérons ici une définition opposée à celle présentée ci-dessus, on pourrait envisager a priori de définir l'incompatibilité comme $Card(A_E \setminus A_e)$. Cette incompatibilité est minimale lorsque $A_E \subseteq A_e$, ce qui signifie que tous les attributs connus sont utilisés par l'explication. Cette incompatibilité ne définit pas une mesure de redondance mais évalue que toutes les notions connues par l'utilisateur sont exprimées par l'explication.

La redondance est plutôt vue comme le nombre d'attributs qui ne sont pas communs à la connaissance considérée et l'explication proposée. Avec les notations introduites ci-dessus, ce principe peut par exemple être associé à la fonction suivante :

$$I_x(e, E) = Card(A_e \cap A_E)$$

Lorsqu'il y a accord total entre la connaissance et l'explication, les attributs présents dans E n'interviennent pas dans l'explication candidate e, c'est-à-dire $A_e \cap A_E = \emptyset$ alors on obtient une incompatibilité minimale qui vaut 0. En revanche, l'incompatibilité est maximale, lorsque tous les attributs présents dans l'explication sont connus par l'utilisateur, c'est-à-dire $A_e \subseteq A_E$, ainsi elle vaut $Card(A_e)$. On remarque que les cas extrêmes des deux définitions de la fonction d'incompatibilité sont inversés.

Dans cette thèse, nous nous concentrons sur la première définition de l'incompatibilité, selon laquelle une explication compatible est une explication dans le langage des connaissances. Dans cette section, nous avons tout d'abord présenté le principe général de la fonction d'incompatibilité. Puis, nous avons donné une première instanciation de cette fonction dans le cas où on peut extraire un ensemble d'attributs à partir de la connaissance. Dans la suite de la thèse, nous étudions la notion d'incompatibilité dans deux cas qui correspondent à des connaissances utilisateur facilement exprimables : d'une part un ensemble d'attributs dans le chapitre 4 et la section 5.1, et d'autre part les règles expertes dans la section 5.2.

3.5 Fonction d'agrégation

La troisième fonction à définir dans le cadre de l'équation (3.1) que nous proposons est la fonction qui combine les deux valeurs de pénalité et d'incompatibilité en une valeur de qualité globale de l'explication : il s'agit d'une fonction d'agrégation que nous discutons ici de façon générale. Le chapitre 6 l'étudie plus en détail. Cette fonction relève du domaine de l'agrégation multicritère qui étudie la combinaison de plusieurs valeurs en une seule. Ce vaste domaine a donné lieu à une abondante littérature, voir Calvo et al., 2002 ou Grabisch et al., 2009.

Cette section propose d'abord quelques rappels sur les fonctions d'agrégation en général en présentant des comportements classiques de l'agrégation. Il transpose ensuite certains d'entre elles au cas de la fonction définie dans l'équation (3.1), c'est-à-dire le cas où l'agrégation porte sur deux critères, qui correspondent aux valeurs de pénalité et d'incompatibilité.

3.5.1 Rappels sur les opérateurs d'agrégation

En nous basant sur les définitions données par Calvo et al., 2002, nous présentons tout d'abord trois comportements classiques des opérateurs d'agrégation : conjonctif, disjonctif et compromis. Puis, nous présentons deux propriétés classiques que les opérateurs d'agrégation peuvent vérifier.

Dans cette section, nous notons l'opérateur d'agrégation g et nous considérons qu'il combine n valeurs : $c_1, c_2, ..., c_n$.

Comportement conjonctif Un opérateur est considéré comme ayant un comportement conjonctif si son résultat prend une valeur élevée uniquement lorsque toutes les valeurs qu'il agrège sont élevées : c_1 est élevée *et* c_2 est élevée *et* c_3 est élevée, etc. Formellement, une fonction conjonctive g est définie comme :

$$g(c_1, ..., c_n) \leq min(c_1, ..., c_n)$$

Par conséquent, une fonction conjonctive vérifie :

$$g(c_1, ..., c_n) \leq c_i$$

Parmi les exemples d'opérateurs conjonctifs, on peut citer le minimum, ou la fonction produit définie sur $[0, 1]$. Une famille générique qui regroupe ces deux exemples est la famille des t-normes $\top$ (Klement et al., 2000) qui du fait de leur propriété de symétrie, monotonie et d'élément neutre vérifient $\top(c_1, ..., c_n) \leq min(c_1, ..., c_n)$.

Comportement disjonctif Un opérateur est considéré comme ayant un comportement disjonctif si son résultat prend une valeur élevée lorsqu'au moins une des valeurs qu'il agrège est élevée : c_1 est élevée *ou* c_2 est élevée *ou* c_3 est élevée, etc. Formellement, une fonction disjonctive g est définie comme satisfaisant :

$$max(c_1, ..., c_n) \leq g(c_1, ..., c_n)$$

Par conséquent, une fonction disjonctive vérifie :

$$c_i \leq g(c_1, ..., c_n)$$

La valeur de g est élevée lorsqu'au moins une des valeurs est élevée. Parmi les exemples d'opérateurs disjonctifs, on peut citer le maximum, ou la somme. Une famille générique qui regroupe les deux exemples précédents est la famille des t-conormes $\perp$ (Klement et al., 2000) qui vérifie $max(c_1, ..., c_n) \leq \perp (c_1, ..., c_n)$.

Comportement de compromis Un opérateur est considéré comme ayant un comportement de compromis si son résultat permet de compenser les valeurs faibles par les valeurs élevées. Formellement, une fonction de compromis g vérifie :

$$min(c_1, ..., c_n) \leq g(c_1, ..., c_n) \leq max(c_1, ..., c_n)$$

Si on considère un ensemble contenant des valeurs faibles et élevées, l'agrégation de cet ensemble n'est ni élevée ni faible. Un exemple de fonction de compromis est la fonction moyenne arithmétique.

Il faut noter que les opérateurs d'agrégation n'appartiennent pas nécessairement à une des familles. Ils peuvent avoir des comportements hybrides, c'est-à-dire qu'ils

peuvent adopter différents comportements. Dans la section 6, nous étudions un tel opérateur hybride, l'intégrale de Gödel.

Propriétés Il existe un très grand nombre de propriétés (Grabisch et al., 2009) que peuvent vérifier les opérateurs d'agrégation. Les propriétés les plus classiques sont la monotonie, la commutativité, l'associativité ou encore l'inégalité triangulaire. La section 6.1 discute de plusieurs propriétés des opérateurs.

3.5.2 Mis en œuvre pour le formalisme proposé

Dans le cas de l'IA explicable, plus précisément dans le cas de la fonction de coût donnée dans l'équation (3.1). La fonction d'agrégation doit combiner deux valeurs correspondant aux critères de pénalité et d'incompatibilité. Contrairement au cas classique d'optimisation par maximisation de la fonction considérée, ici nous cherchons à minimiser la fonction de coût. Il est alors nécessaire de redéfinir les comportements des opérateurs d'agrégation présenté ci-dessus pour la pénalité et l'incompatibilité, en particulier dans le cas des opérateurs conjonctifs et disjonctifs. Par contre pour l'opérateur de compromis, le comportement reste le même. Nous présentons ici la différence de comportement des opérateurs conjonctifs et disjonctifs.

Premièrement, un opérateur conjonctif renvoie une valeur faible si toutes les valeurs sont faibles : $P_x(e,f)$ faible *et* $I_x(e,E)$ faible. Par exemple, la fonction maximum vérifie $P_x(e,f) < max(P_x(e,f), I_x(e,E))$ et $I_x(e,E) < max(P_x(e,f), I_x(e,E))$. La valeur maximale est faible si les deux critères sont faibles. Aussi, dans ce cas et contrairement au cas classique, ce sont les fonctions de la famille des t-conormes qui ont un comportement conjonctif.

Deuxièmement, un opérateur disjonctif renvoie une valeur faible si l'une des valeurs est faible : $P_x(e,f)$ faible *ou* $I_x(e,E)$ faible. Par exemple, la fonction minimum vérifie $min(P_x(e,f), I_x(e,E)) < P_x(e,f)$ et $min(P_x(e,f), I_x(e,E)) < I_x(e,E)$. La valeur minimale est faible si l'un des deux critères est faible. Aussi, dans ce cas et contrairement au cas classique, ce sont les fonctions de la famille des t-normes qui ont un comportement disjonctif.

Pour combiner les critères de pénalité et d'incompatibilité, nous proposons dans les chapitres 4 et 5 de considérer une fonction classique qui a un comportement de compromis, une somme pondérée. Dans la section 6, nous focalisons notre étude sur l'agrégation des critères en choisissant une fonction d'agrégation plus complexe, les intégrales de Gödel.

3.6 Illustration du formalisme général proposé

Afin d'illustrer la généralité du cadre proposé, nous montrons dans cette section comment il peut être instancié pour exprimer l'approche de l'état de l'art proposée

par Ustun et al., 2019 présentée dans la section 2.7.4, en mettant en évidence la définition des trois fonctions impliquées.

Dans l'approche d'Ustun et al., 2019, la connaissance de l'utilisateur E est locale et dépend de l'instance considérée : elle est notée $A(x)$ et est définie comme l'ensemble des modifications autorisées, qui peuvent être appliquées à l'instance considérée x. Cette connaissance peut être vue comme un ensemble d'attributs associés à une plage de valeurs dépendant de x. Son intégration permet de générer une explication contre-factuelle actionnable. Pour un classifieur f, cette explication est définie comme la solution au problème d'optimisation suivant :

$$\eta^* = \underset{\eta \in A(x)}{\operatorname{argmin}} \, cost\_fct(\eta, x) \text{ avec } f(x+\eta) \neq f(x) \tag{3.2}$$

Avec les notations de la section 2.7.4, la fonction de coût $cost\_fct$ est définie comme la distance entre x et $x+\eta$. L'explication générée η^* exprime les modifications à effectuer à partir de l'instance considérée x pour obtenir l'instance $x+\eta^*$ la plus proche dans la classe opposée.

Une première étape consiste à considérer l'explication finale sous une autre forme. Nous posons : $e^* = x + \eta^*$. Le problème d'optimisation peut être réécrit comme suit :

$$e^* = \underset{|e-x| \in A(x)}{\operatorname{argmin}} \; cost\_fct(e-x, x) \text{ avec } f(e) \neq f(x)$$

Une deuxième étape consiste à définir différemment l'espace de recherche. Au lieu de le définir à partir des modifications réalisables, nous le définissons comme l'ensemble des instances prédites différemment de x, ainsi il s'écrit $\mathcal{E}_{x,f} = \{x' \in \mathcal{X} \mid f(x') \neq f(x)\}$. Le problème d'optimisation défini dans l'équation (3.2) ci-dessus, peut être réécrit avec une nouvelle fonction de coût comme suit :

$$e^* = \underset{e \in \mathcal{E}_{x,f}}{\operatorname{argmin}} \, cost\_fct(e-x, x) + \mathbb{1}_{|x-e| \notin A(x)} \times Z$$

où Z est un nombre réel arbitrairement grand. On remarque que l'ancienne fonction de coût $cost\_fct$ devient une composante de la nouvelle fonction de coût.

Cette expression, équivalente à la précédente, permet d'identifier les fonctions P_x, I_x et agg, respectivement définies comme :

$$\begin{aligned} P_x(e, f) &= cost\_fct(e-x, x) \\ I_x(e, E) &= \mathbb{1}_{|x-e| \notin A(x)} \times Z \\ agg(u, v) &= u + v \end{aligned}$$

où Z est une valeur arbitraire élevée.

La fonction de pénalité est égale à $cost\_fct$ définie par Ustun et al., 2019. La fonction d'incompatibilité est égale à $\mathbb{1}_{|x-e| \notin A(x)} \times Z$ et représente la présence ou l'absence d'un attribut modifié dans les connaissances de l'utilisateur ; elle ne prend que deux valeurs

0 ou Z. Une explication incompatible a donc une fonction de coût très élevée, ce qui fait que seules les explications compatibles sont prises en compte. Enfin, l'agrégation est réalisée par une somme. Cependant, il faut noter que comme la fonction d'incompatibilité est binaire, (0 si incompatible et Z si compatible) seuls les contre-factuels compatibles sont pris en compte. Ainsi, le contre-factuel résultant est à la fois compatible et de bonne qualité, c'est-à-dire que l'agrégation choisie ici a un comportement conjonctif.

Cette section montre que la méthode de Ustun et al., 2019 étudie un problème d'optimisation qui s'écrit sous le formalisme général que nous proposons. Une différence importante entre cette méthode et les méthodes que nous proposons dans les chapitres suivants est qu'elle se limite à un modèle f linéaire, alors que nos travaux se placent dans un cas agnostique au modèle, ils ne sont pas spécifiques à un modèle en particulier.

3.7 Bilan

Cette section a proposé un formalisme général qui intègre des connaissances utilisateur dans le but de générer des explications personnalisées. Ce formalisme agrège deux critères qui sont la pénalité et l'incompatibilité. Le premier mesure la qualité d'une explication par rapport au modèle et le second mesure la fidélité d'une explication par rapport à l'utilisateur. Ainsi, la fonction de coût mesure l'adéquation de l'explication proposée aux deux composantes de référence : le modèle à expliquer et la connaissance utilisateur.

Dans les prochains chapitres, nous étudions les termes présentés dans ce formalisme en proposant des instanciations pour différents types d'explications et de connaissances. Les chapitres 4 et 5 se concentrent sur la notion d'incompatibilité. Le chapitre 4 étudie la génération d'explications contre-factuelles en intégrant des connaissances sous forme d'ensemble d'attributs. Le chapitre 5 propose deux instanciations, une première qui étudie une autre forme d'explications, les vecteurs d'importance des attributs, et une seconde qui étudie une autre forme de connaissances, un système de règles. Après avoir étudié la notion d'incompatibilité, nous nous focalisons sur l'agrégation de la pénalité et de l'incompatibilité dans le chapitre 6.

Chapitre 4

Knowledge Integration in Counterfactual Explanation (KICE)

Ce chapitre présente une instanciation du cadre général défini dans le chapitre précédent dans le cas particulier d'explications pour des données tabulaires sous forme d'exemples contre-factuels et de connaissances sous forme d'ensemble d'attributs. L'objectif de ce chapitre est de proposer une explication contre-factuelle dans le langage de l'utilisateur, c'est-à-dire les modifications s'effectuent selon les attributs connus par l'utilisateur.

Nous proposons des instanciations pour les trois fonctions qui interviennent dans le formalisme général : P_x, I_x et agg en focalisant la discussion sur la définition de l'incompatibilité. La fonction de pénalité est définie par la proximité de l'exemple contre-factuel à l'instance étudiée, car comme évoqué dans la section 2.5.2, une bonne explication contre-factuelle est souvent associée à peu de modifications pour obtenir la classe souhaitée. Pour l'opérateur d'agrégation, nous considérons ici une moyenne pondérée, le choix de cet opérateur est étudié dans le chapitre 6. Ainsi, l'objectif de ce chapitre est de proposer une explication proche de l'instance étudiée qui est associée à des modifications selon les attributs connus de l'utilisateur.

Premièrement, nous présentons l'instanciation du formalisme général proposé dans le chapitre 3 dans le cadre étudié. Puis, pour résoudre le problème d'optimisation obtenu, nous proposons une nouvelle méthode nommée *Knowledge Integration in Counterfactual Explanation* (KICE). Les sections 4.3 et 4.4 présentent l'étude expérimentale effectuée dans le but d'évaluer cette méthode. Cette étude présente des exemples illustratifs, ainsi qu'une comparaison de différentes méthodes sur plusieurs jeux de données. Enfin, nous discutons des perspectives du formalisme proposé.

Ce travail a été présenté dans l'article *Integrating Prior Knowledge in Post-hoc Explanations* publié à la conférence IPMU 2022 (Jeyasothy et al., 2022a) et à la conférence LFA 2022 (Jeyasothy et al., 2022b).

4.1 Instanciation des critères pour les exemples contre-factuels

Cette section décrit l'instanciation de la fonction de coût générale que nous définissons dans l'équation (3.1) pour le cas d'une explication sous forme d'exemples contre-factuels, de connaissances utilisateur sous forme d'ensemble d'attributs et d'une explication exprimée dans le langage de l'utilisateur. Tout d'abord, nous présentons le type de connaissances considéré. Puis, nous décrivons les trois fonctions : P_x, I_x et agg dans le cadre étudié.

4.1.1 Caractéristiques des types de connaissances

Nos travaux considèrent des données tabulaires, décrites par d attributs numériques, c'est-à-dire l'espace des données $\mathcal{X} \subseteq \mathbb{R}^d$. Nous proposons de considérer des connaissances utilisateur sous la forme d'un ensemble d'attributs comme dans la section 3.4, $E = \{X_i, i = 1 \dots m\}$ avec $m \leq d$. Comme décrit dans la section 2.7.2.1, les attributs de cette connaissance peuvent représenter différentes sémantiques, ils peuvent être considérés comme des préférences de l'utilisateur, des attributs compréhensibles par l'utilisateur ou encore des attributs actionnables comme utilisés par Ustun et al., 2019. Nous considérons que la connaissance définit les attributs compréhensibles par l'utilisateur, mais nos travaux peuvent s'appliquer également aux autres sémantiques.

4.1.2 Fonction de pénalité

Pour la fonction de pénalité, nous considérons le cas classique où elle est définie par la proximité de l'exemple contre-factuel à l'instance étudiée. Comme discuté dans la section 2.5.4, la pénalité est souvent définie par le carré d'une distance euclidienne (Lash et al., 2017).

$$P_x(e, f) = \| x - e \|^2 \tag{4.1}$$

Il faut noter que dans le cas particulier des explications contre-factuelles, la fonction de pénalité ne dépend pas du modèle f ; dans la suite nous notons la fonction $P_x(e)$.

Une perspective aux travaux menés dans la thèse est de considérer des fonctions plus riches, comme celles présentées dans la section 2.5.4, incluant par exemple un critère de parcimonie de l'explication générée. La prise en compte de telles autres fonctions n'impacte pas le principe que nous proposons pour l'intégration de la connaissance.

4.1.3 Fonction d'incompatibilité

Un des objectifs est de proposer une explication contre-factuelle en accord avec les connaissances de l'utilisateur, ce qui signifie qu'idéalement, les modifications contre-factuelles ne doivent être effectuées qu'en fonction des attributs apparaissant dans E. Cependant, comme discuté dans la section 3.1.2, le fait de ne se concentrer que sur un

sous-ensemble des attributs augmente le risque de ne pas pouvoir atteindre la frontière de décision de f, ce qui ferait qu'aucune explication contre-factuelle ne serait générée.

Nous proposons donc d'assouplir cette contrainte en pénalisant les modifications en fonction des attributs de $\overline{E}$, c'est-à-dire des attributs qui ne sont pas présents dans la connaissance E. Cela permet de s'assurer de l'existence d'une solution. Nous proposons donc de définir la fonction d'incompatibilité comme le carré de la distance euclidienne uniquement selon les attributs absents :

$$I_x(e, E) = \| x - e \|^2_{\overline{E}} = \sum_{i \notin E} (x_i - e_i)^2 \tag{4.2}$$

Aussi, l'incompatibilité vaut 0 lorsque l'explication contre-factuelle ne modifie aucun attribut inconnu. Par contre, la valeur est élevée si les modifications selon les attributs non présents dans E sont importantes. La minimisation de cette incompatibilité permet d'éviter de générer des explications contre-factuelles qui modifient considérablement les caractéristiques inconnues.

4.1.4 Fonction d'agrégation

Dans l'idéal, une bonne explication a à la fois une faible incompatibilité et une faible pénalité. Toutefois, il n'est pas nécessairement possible d'atteindre ces deux objectifs simultanément : le gain selon un critère n'implique pas toujours le gain selon le second critère. Au contraire, dans certains cas plus une explication est compatible, plus elle est éloignée de l'instance étudiée dans l'espace des données, il y a alors une hausse de la pénalité. C'est pourquoi, nous proposons de considérer les deux critères en effectuant un compromis. Plus précisément, nous proposons de considérer une somme pondérée :

$$agg(u, v) = u + \lambda v \tag{4.3}$$

où $\lambda \in \mathbb{R}^+$ est un hyperparamètre défini par l'utilisateur : λ contrôle l'importance associée à l'incompatibilité aux connaissances par rapport à la pénalité. Ceci peut par exemple représenter l'ouverture de l'utilisateur à accepter des modifications selon les attributs qu'il ne pourrait pas comprendre. Une valeur élevée de λ implique que l'exemple contre-factuel peut être situé plus loin de l'instance de référence, ce qui signifie que l'utilisateur peut avoir besoin d'effectuer plus de modifications. En contrepartie, ces modifications ne s'appliquent qu'aux éléments dont il comprend la signification.

Fonction globale Nous obtenons alors le problème d'optimisation suivant :
Soit $\mathcal{E}_{x,f} = \{x' \in \mathcal{X} \mid f(x') \neq f(x)\}$ et $\lambda \in \mathbb{R}^+$,

$$\begin{aligned} e^* &= \underset{e \in \mathcal{E}_{x,f}}{\operatorname{argmin}} \, cost_{x,E}(e) \\ \text{avec } cost_{x,E}(e) &= \| x - e \|^2 + \lambda \| x - e \|^2_{\overline{E}} \end{aligned} \tag{4.4}$$

Algorithm 1 KICE : Knowledge Integration in Counterfactual Explanation

Require: $f : \mathcal{X} \longrightarrow \{0, 1\}$, le modèle à expliquer
Require: $x \in \mathcal{X}$, l'instance considérée
Require: E, la connaissance utilisateur
Require: Paramètres : $\nu_0, \epsilon, n, \lambda$
Ensure: $e^* = \underset{e \in \mathcal{E}_{x,f}}{\text{argmin}} \parallel x - e \parallel^2 + \lambda \parallel x - e \parallel_{\overline{E}}^2$

1: Générer $\mathcal{Z} = \{z_i\}_{i \leq n} \sim \mathcal{U}(\mathcal{EL}(x, 0, \nu_0, \lambda, E))$ avec GCE (algorithme 2)
2: **while** $\exists e \in \mathcal{Z}, f(e) \neq f(x)$ **do**
3: $\quad \nu_0 \leftarrow \nu_0 / 2$
4: $\quad$ Générer $\mathcal{Z} = \{z_i\}_{i \leq n} \sim \mathcal{U}(\mathcal{EL}(x, 0, \nu_0, \lambda, E))$ avec GCE (algorithme 2)
5: **end while**
6: $a_0 \leftarrow \nu_0$
7: $a_1 \leftarrow \nu_0 + \epsilon$
8: **while** $\nexists e \in \mathcal{Z}, f(e) \neq f(x)$ **do**
9: $\quad$ Générer $\mathcal{Z} = \{z_i\}_{i \leq n} \sim \mathcal{U}(\mathcal{EL}(x, a_0, a_1, \lambda, E))$ avec GCE (algorithme 2)
10: $\quad a_0 \leftarrow a_1$
11: $\quad a_1 \leftarrow a_0 + \epsilon$
12: **end while**
13: $e^* = \underset{e \in \mathcal{Z}, f(e) \neq f(x)}{\text{argmin}} \parallel x - e \parallel^2 + \lambda \parallel x - e \parallel_{\overline{E}}^2$
14: **return** e^*

4.2 Description de l'algorithme KICE

Dans cette section, nous décrivons l'algorithme KICE, *Knowledge Integration in Counterfactual Explanation*, que nous proposons pour résoudre le problème d'optimisation défini par l'équation (4.4). Tout d'abord, nous présentons le principe utilisé par cet algorithme. Ensuite, nous détaillons l'étape de génération des instances.

Principe KICE dont le pseudo-code est donné dans l'algorithme 1, utilise le principe de génération itérative d'instances. Ce dernier correspond à la première étape mise en œuvre par l'algorithme Growing Spheres (Laugel et al., 2018a) présenté dans la section 2.5.5. Toutefois, KICE utilise des couches générées différemment. En effet, comme détaillé ci-dessous, les lignes de niveaux de la fonction de coût ne sont pas sphériques.

A l'initialisation, KICE génère des instances dans une couche ellipsoïdale de centre x et de rayon $\sqrt{\nu_0}$ selon les attributs de E et $\sqrt{\frac{\nu_0}{1+\lambda}}$ selon les attributs de $\overline{E}$ (ligne 1) où ν_0 un paramètre choisi en entrée. Si les instances générées appartiennent à une autre classe que x alors des couches de plus en plus petites sont générées jusqu'à ce qu'aucun exemple contre-factuel ne soit trouvé (ligne 2 à 5). Enfin, KICE génère des instances dans des espaces de plus en plus grands autour de x jusqu'à ce qu'il trouve une instance prédite différemment par f (ligne 8 à 12). Ainsi, Growing Spheres et KICE diffèrent sur le type de couche généré, sphérique pour le premier et ellipsoïdale pour le second (ligne 1, 4 et 9). La forme de la couche est modifiée car l'intégration de E dans le terme d'incompatibilité supplémentaire de la fonction de coût nécessite une génération non uniforme : l'espace est déformé, les attributs étant associés à des poids différents.

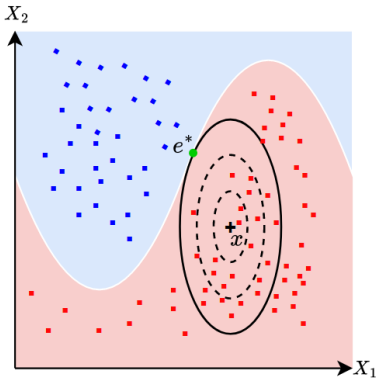


FIGURE 4.1 – Illustration de l'étape de génération des couches ellipsoïdales de KICE pour un jeu de données 2D : les prédictions de f sont représentées par les régions colorées et $E = \{X_2\}$.

A chaque étape, les instances sont générées dans des couches associées à la fonction de coût définie dans l'équation (4.4). Pour tout ν, l'équation $cost_{x,E}(e) = \nu$ définit une ellipse de centre x et de rayon $\sqrt{\frac{\nu}{1+\lambda}}$ par rapport aux attributs de $\overline{E}$ et $\sqrt{\nu}$ par rapport aux attributs de E.

Ce principe est illustré par la figure 4.1 qui présente un ensemble de données 2D, la prédiction du classifieur, représentée par les régions bleues et rouges, et une instance x. Les connaissances de l'utilisateur sont définies comme $E = \{X_2\}$. Par conséquent, il est moins coûteux de modifier l'attribut X_2 que l'attribut X_1. Des instances sont donc générées itérativement dans les ellipses "verticales" jusqu'à ce que l'explication e^* soit trouvée.

Génération uniforme des couches ellipsoïdales Pour deux valeurs a_0 et a_1, nous définissons la couche ellipsoïdale $\mathcal{EL}(x, a_0, a_1, \lambda, E)$ comme :

$$\mathcal{EL}(x, a_0, a_1, \lambda, E) = \{z \in \mathcal{X}, a_0 \leq \| x - z \|^2 + \lambda \| x - z \|^2_{\overline{E}} \leq a_1\}$$

Afin de générer des candidats uniformément dans ces couches ellipsoïdales, KICE s'appuie sur une version modifiée GCE (algorithme 2), de la procédure HLG (Muller, 1959). Cette dernière génère des instances uniformément dans la couche sphérique $SL(x, a_0, a_1)$ définie comme l'ensemble des points situés à une distance au carré supérieure à a_0 et inférieure à a_1 de x. La génération d'instances par la méthode HLG repose sur une variable aléatoire $U \sim \mathcal{U}([\sqrt{a_0}, \sqrt{a_1}])$.

La procédure GCE que nous proposons distingue les attributs de E et de $\overline{E}$: pour les attributs de E, la procédure est similaire à HLG, en considérant $U \sim \mathcal{U}([\sqrt{a_0}, \sqrt{a_1}])$. Pour les attributs de $\overline{E}$, la génération est pondérée par $\frac{1}{\sqrt{1+\lambda}}$, c'est-à-dire qu'elle considère $U \sim \mathcal{U}([\sqrt{\frac{a_0}{1+\lambda}}, \sqrt{\frac{a_1}{1+\lambda}}])$. Ces valeurs sont normalisées sur le même principe que HLG pour qu'elles soient uniformément dans une couche ellipsoïdale.

Globalement, l'algorithme KICE couvre l'espace en générant des instances de manière itérative : dans un premier temps, n instances sont générées dans l'ellipse de

Algorithm 2 Génération des Couches Ellipsoïdales (GCE)

Require: x, centre de la couche ellipsoïdale
Require: E, la connaissance utilisateur
Require: a_0 et a_1 les limites de la couche
Require: n, nombre de points souhaités
Require: λ, poids
Ensure: $Z = \{z_i\}_{i \leq n} \sim \mathcal{U}(\mathcal{EL}(x, a_0, a_1, \lambda, E)$
$Y = \{y_i\}_{i \leq n} \sim \mathcal{N}(0, 1)$
$Y \leftarrow \frac{Y}{\|Y\|_2}$
$U \leftarrow \{u_i\}_{i \leq n} \sim \mathcal{U}([(\sqrt{a_0})^{dim(x)}, (\sqrt{a_1})^{dim(x)}])$
$R \leftarrow U^{1/dim(x)}$
$W \leftarrow R^T Y + x$
$R' \leftarrow \frac{1}{\sqrt{1+\lambda}} U^{1/dim(x)}$
$W' \leftarrow R'^T Y + x$
for $j \leftarrow 1, ..., dim(x)$ **do**
    if $j \in E$ **then**
        $Z_j \leftarrow W_j$
    else
        $Z_j \leftarrow W'_j$
    end if
end for
return Z

centre x et de rayon $\sqrt{\frac{\nu}{1+\lambda}}$ par rapport aux attributs de $\overline{E}$ et $\sqrt{\nu}$ par rapport aux attributs de E. Si aucune de ces instances n'est prédite différemment de x, KICE génère des instances dans la couche ellipsoïdale entre ν et $\nu + \epsilon$ où $\epsilon > 0$ est un hyperparamètre.

Paramètres Dans cette section, nous résumons les arguments d'entrée considérés. Nous distinguons parmi ces arguments deux types : ceux définis par l'utilisateur à qui l'explication est fournie et ceux définis par l'informaticien qui fournit l'explication.

L'utilisateur définit l'instance étudiée x et sa connaissance E qui ne dépend ni du formalisme étudié ni de l'algorithme KICE. L'utilisateur choisit également le poids λ qui représente l'importance qu'il accorde au respect des connaissances par rapport à la proximité de l'explication à l'instance étudiée. La section 4.5.1 discute du choix de ce paramètre.

L'informaticien choisit les paramètres de la méthode KICE. La variable n correspond au nombre d'instances générées à chaque étape : plus n est élevé, plus l'explication obtenue sera précise mais le coût de calcul sera élevé. Le paramètre ν_0 définit le rayon de la première couche, si cette valeur est faible, le nombre d'étape pour atteindre la frontière de décision sera élevé. Une valeur élevée de ν_0 est alors intéressante lorsque l'instance étudiée est loin d'une frontière de décision. ϵ représente l'intervalle entre deux couches successives : plus la valeur est faible plus l'explication obtenue sera précise, par contre le coût de calcul sera élevé.

Jeux de données	Nombre d'instances	Nombre d'attributs	|E|	Précision
Californie	20 640	7	4	0.84
Half-Moons	1000	2	1	0.99
Boston	506	13	7	0.98
Breast-cancer	569	30	15	0.93

TABLE 4.1 – Caractéristiques des jeux de données

4.3 Protocole expérimental

Nous présentons dans cette section le protocole expérimental utilisé pour évaluer la méthode KICE. Tout d'abord, nous présentons les quatre jeux de données considérés. Puis, nous détaillons le protocole considéré en présentant les valeurs des paramètres de KICE. Ensuite, nous présentons les compétiteurs auxquels nous comparons les résultats de la méthode proposée. Enfin, nous présentons les trois métriques considérées pour évaluer la méthode.

4.3.1 Jeux de données

Nous menons des expérimentions sur quatre ensembles de données tabulaires de référence : Californie [1], Half-Moons [2], Boston [3] et Breast Cancer [4] dont les caractéristiques sont résumées dans le tableau 4.1. Ces jeux de données ont été choisis car ce sont des jeux de données classiques de l'état de l'art. De plus, ils sont de dimensions différentes ce qui permet d'observer les résultats dans différents cas de figure. Half-Moons est un jeu de données synthétiques en deux dimensions générées par le package sklearn [5]. Pour le jeu de données Californie, le tableau 4.2 détaille le sens des sept attributs. Le jeu de données Boston est souvent associé à un problème de régression. Dans notre contexte, nous étudions les problèmes de classification, ainsi la valeur de régression est transformée en une classe binaire au moyen d'une étape de discrétisation : le prix est "cher" s'il est supérieur à 21 000$, et "bon marché" dans le cas contraire. De plus, nous normalisons les jeux de données afin de considérer la même échelle pour tous les attributs.

4.3.2 Protocole

Nous présentons dans cette section le classifieur à expliquer, les connaissances utilisateur considérées et les paramètres choisis pour la méthode KICE.

Classifieur Chaque jeu de données est normalisé de manière standard, c'est-à-dire que chaque valeur est calculée en soustrayant à la valeur de départ la moyenne des valeurs et ensuite en divisant par l'écart-type. Puis, les ensembles de données sont divisés en

1. https://scikit-learn.org/stable/datasets/real_world.html#california-housing-dataset
2. https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_moons.html
3. https://scikit-learn.org/0.15/modules/generated/sklearn.datasets.load_boston.html
4. https://scikit-learn.org/0.21/modules/generated/sklearn.datasets.load_breast_cancer.html
5. https://scikit-learn.org/stable/

a_0	longitude
a_1	latitude
a_2	âge médian des logements
a_3	nombre moyen de pièces par ménage
a_4	nombre moyen de chambres par ménage
a_5	population totale
a_6	revenu médian du logement

TABLE 4.2 – Caractéristiques du jeu de données Californie, chaque instance représente un groupe d'appartements

	λ	ϵ	n	ν_0
Californie	5	0.05	2000	0.1
Half-moons	4	0.01	200	0.1
Boston	3	0.02	1000	0.2
Breast Cancer	6	0.3	2000	5

TABLE 4.3 – Valeurs des paramètres λ, ϵ, n et ν_0 de KICE choisies pour quatre jeux de données

sous-ensembles de données d'entraînement et de test (80%-20%). Dans le cadre de l'explication post-hoc envisagée, le choix du classifieur n'a pas d'importance. Nous ne cherchons pas à avoir le meilleur classifieur mais à expliquer tout type de classifieur. Nous appliquons un classifieur SVM avec un noyau gaussien qui atteint globalement une bonne précision sur les trois jeux de données comme présenté dans la dernière colonne du tableau 4.1.

Connaissances utilisateur Nous considérons que l'utilisateur connaît moins d'attributs que ceux considérés par le classifieur pour prédire. Pour le jeu de données Californie, nous considérons une connaissance de trois attributs : $\{a_3, a_4, a_6\}$. Ils représentent les caractéristiques les plus compréhensibles parmi les sept attributs. Pour les autres jeux de données, dans le but de construire une connaissance plausible, nous entraînons un arbre de décision de profondeur faible sur les données d'entraînement. Plus précisément, nous fixons la profondeur maximale à la moitié du nombre total d'attribut. L'ensemble des attributs E que nous considérons contient alors les attributs présents dans les différents nœuds de cet arbre.

Paramètres Les valeurs fixées pour les paramètres λ, ϵ, n et ν sont indiquées dans le tableau 4.3. Elles sont choisies en fonction de la dimension des ensembles de données et du nombre d'attributs considérés pour E. Plus le jeu de données a d'attributs, plus la valeur des paramètres ϵ, n et ν est élevée. Quant à la valeur de λ, elle est choisie de manière à avoir un bon compromis, c'est-à-dire dans cette section une faible perte en pénalité et un gain élevé en incompatibilité. Le choix de ce paramètre est discuté en détail dans la section 4.5.1.

Les explications e^* sous forme d'exemples contre-factuels sont ensuite générées pour chaque instance x de l'ensemble de données de test à l'aide de KICE.

4.3.3 Compétiteurs

Nous comparons les résultats obtenus à deux méthodes générant des explications associées aux valeurs extrêmes de λ.

λ = 0 La première méthode résout le problème d'optimisation de référence qui minimise uniquement la fonction de pénalité, son explication générée est notée e_{ref}. Cela correspond à un cas extrême d'agrégation de l'équation (4.4) où le terme d'incompatibilité est ignoré et où le coût est égal à la pénalité. Cette méthode correspond à l'algorithme Growing Spheres (Laugel et al., 2018a) présenté dans la section 2.5.5.

λ arbitrairement grand Un second concurrent est proposé en imposant le respect strict de la connaissance. La fonction de coût qui lui est associée se restreint aux explications totalement compatibles, parmi ces explications l'instance la plus proche est choisie. Elle correspond à une façon simple d'intégrer la connaissance dans l'explication. Nous notons e_{user} l'exemple contre-factuel qui résout le problème associé :

$$e_{user} = \underset{e \in \mathcal{E}_{x,f}}{\operatorname{argmin}} \parallel x - e \parallel^2 \qquad \text{sous contrainte} \qquad \parallel x - e \parallel_E^2 = 0$$

Ce problème d'optimisation peut être vu comme un cas particulier de l'équation (4.4) avec λ arbitrairement grand, où on s'assure que l'incompatibilité est nulle. Par construction, s'il existe une explication totalement compatible, prendre une valeur de λ arbitrairement grand implique une compatibilité totale.

4.3.4 Métriques

Afin d'analyser les explications obtenues, nous comparons les résultats selon trois métriques quantitatives qui correspondent aux trois fonctions : P_x, I_x et $cost_{x,E}$. La fonction de pénalité définie par l'équation (4.1) permet de comparer l'explication proposée à celle de référence e_{ref}. Ensuite, la fonction d'incompatibilité définie par l'équation (4.2) permet de comparer l'explication proposée à celle totalement compatible e_{user} quand elle existe. Enfin, la fonction de coût définie par l'équation (4.4) permet de vérifier que l'explication proposée est bien celle souhaitée, c'est-à-dire qu'elle résout le problème d'optimisation étudié.

4.4 Étude expérimentale

Cette section présente les résultats menés selon le protocole détaillé dans la section précédente, pour évaluer l'algorithme KICE. Tout d'abord, nous présentons deux exemples illustratifs : le premier dans un cas réaliste avec le jeu de données Californie et

le second dans un cas artificiel avec les données synthétiques Half-Moons. Ensuite, nous nous appuyons sur différentes métriques pour comparer la méthode KICE aux deux méthodes extrêmes. Puis, nous étudions les temps d'exécution de chacune des méthodes. Enfin, nous comparons en détail la valeur associée à la fonction de coût obtenue par KICE et les compétiteurs.

4.4.1 Exemple de résultats de KICE sur la base Californie

Cette première expérimentation a pour but d'observer les explications obtenues pour un exemple concret, qui est le jeu de données Californie. Étant donné que ce jeu de données sera étudié dans les expérimentations des prochains chapitres, nous présentons dans le tableau 4.2 les sept attributs considérés. Le tableau 4.4 présente les résultats obtenus pour une instance de référence, indiquée dans la première ligne, et le tableau 4.5 les valeurs des métriques pour chaque explication.

Tout d'abord, nous analysons les résultats obtenus dans le tableau 4.4. Le but de la méthode proposée est de pénaliser les modifications selon les attributs absents dans E, ainsi nous mettons en valeur les modifications selon les attributs inconnus, en rouge s'il y a des modifications et en vert s'il n'y a pas de modifications. On remarque comme attendu que e_{user} ne modifie que les attributs de E, c'est-à-dire ceux connus par l'utilisateur. Quant à e_{ref}, l'explication sans prise en compte des connaissances propose des modifications selon les quatre attributs non présents dans E, c'est-à-dire ceux que l'utilisateur ne connaît pas. Cette explication n'est donc pas souhaitée. L'explication fournie par KICE modifie deux des quatre attributs, elle est alors plus adaptée que e_{ref}, même si deux attributs inconnus, l'âge médian des logements et la population totale sont modifiés. De plus, on remarque que la modification selon cet attribut n'est pas très grande par rapport aux modifications effectuées par e_{ref}, ce qui signifie que l'explication e^* demande peu d'efforts sur les attributs inconnus. Ainsi, e^* est une bonne alternative pour l'utilisateur, elle nécessite seulement de connaître deux attributs supplémentaires à ceux qu'il connaît déjà.

Ensuite, on compare ces résultats selon les trois métriques considérées dans le tableau 4.5. L'explication proposée e^* a une incompatibilité égale à 0.15, ce qui est très faible comme souhaité. Sa pénalité vaut 5.23. Par rapport à e_{ref} il y a une perte de pénalité de 2.08, e_{user} quant à lui est associé à une perte de 3.46. Ainsi, on remarque que la perte de pénalité de e^* est presque quatre fois plus faible que celle de e_{user}. KICE permet donc de générer une explication de compatibilité élevée qui n'a pas une grande perte de qualité.

4.4.2 Exemples illustratifs en deux dimensions sur Half-Moons

Cette section a pour but d'observer en deux dimensions les explications dans l'espace des données. Pour cela, on note X_1 et X_2 l'abscisse et l'ordonnée. La figure 4.2 présente les exemples contre-factuels e_{ref}, e_{user} et e^* obtenus pour trois instances différentes x (symbole +). Les points bleus et rouges représentent les données d'entraînement

	a_0	a_1	a_2	$\boldsymbol{a_3}$	$\boldsymbol{a_4}$	a_5	$\boldsymbol{a_6}$
x	-122.0	37.2	28	7.70	1.00	1085	10.68
e_{ref}	-1.6	-0.8	+6	0	+0.22	-61	-1.08
e_{user}	0	0	0	-0.04	-0.65	0	-4.08
e^*	0	0	-3	0	-0.69	+315	-3.23

TABLE 4.4 – Exemples e_{ref}, e_{user} et e^* pour l'instance x du jeu de données Californie donnée dans la première ligne et les métriques associées avec les attributs E en gras (vert : pas de modification, rouge : modification)

	$P_x(e)$	$I_x(e, E)$	$cost_{x,E}(e)$
e_{ref}	**3.15**	2.62	16.25
e_{user}	6.61	**0.0**	6.61
e^*	5.23	0.15	**5.98**

TABLE 4.5 – Valeurs des métriques : $P_x(e)$, $I_x(e, E)$ et $cost_{x,E}(e)$ pour l'instance x du jeu de données Californie donné dans la première ligne du tableau 4.4

et les différentes régions représentent les classes prédites. La frontière de décision du classifieur SVM est représentée en blanc, sa précision est de 0.99. Pour la connaissance experte, nous entraînons un arbre de décision de profondeur égale à 1, deux règles sont alors obtenues :

$$\begin{cases} X_2 > 0.1 \implies classe = bleu \\ X_2 \leq 0.1 \implies classe = rouge \end{cases}$$

Un seul attribut est présent dans ce système de règles, la connaissance experte considérée est $E = \{X_2\}$.

Nous observons que l'exemple contre-factuel e_{ref} est le point le plus proche appartenant à une autre classe. Comme attendu, e_{user} est plus éloigné que e_{ref} et ne modifie que l'attribut X_2 qui est l'attribut présent dans la connaissance. De plus, nous remarquons que e^* est un compromis entre e_{ref} et e_{user}. Il nécessite moins de modifications selon X_1 que e_{ref}, il est donc plus compatible. Il est également plus proche de l'instance étudiée que e_{user}. Sur la figure de droite, nous remarquons que l'exemple contre-factuel e_{user} n'apparaît pas : dans ce cas, il n'existe aucun contre-factuel totalement compatible, c'est-à-dire qu'il est impossible d'appartenir à une autre classe en modifiant uniquement l'attribut X_2. Ainsi, la méthode proposée est utile car elle permet de générer une explication plus compatible que e_{ref} sans être très éloignée.

4.4.3 Évaluation comparative de la méthode KICE

Dans cette section, nous comparons notre méthode KICE aux compétiteurs selon les métriques considérées : la pénalité, l'incompatibilité ainsi que la fonction de coût.

Nous appliquons les trois méthodes décrites dans la section 4.3.3 sur les données test des trois jeux de données : Half-moons, Boston et Breast cancer. Parmi les données de l'ensemble test, certaines instances n'ont aucune explication contre-factuelle qui ne modifie que les attributs présents dans E, c'est-à-dire qu'il n'existe pas d'explication e_{user}

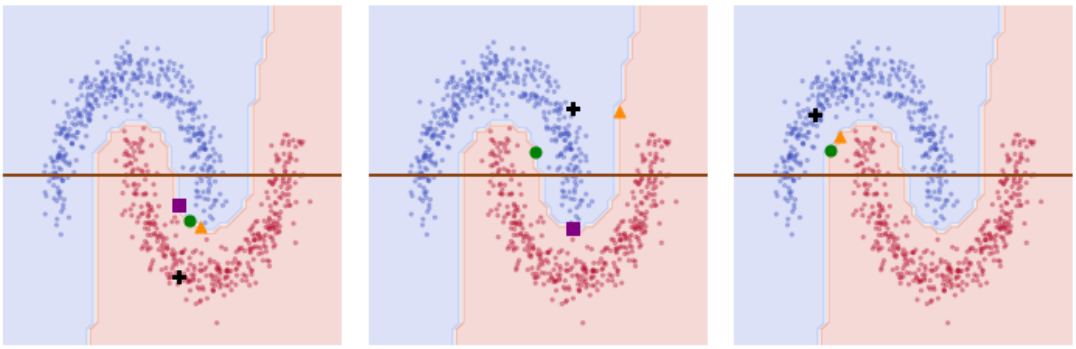


FIGURE 4.2 – Exemples des résultats e_{ref}, e_{user} et e^* pour trois instances x (+ : x, ▲ : e_{ref}, ■ : e_{user}, ● : e^*), $E = \{X_2\}$

		$P_x(e)$	$I_x(e,E)$	$cost_{x,E}(e)$
Half-moons	e_{ref}	**0.32** $\pm$ **0.21**	0.14 $\pm$ 0.13	0.86 $\pm$ 0.56
	e_{user}	1.48 $\pm$ 1.3	**0.0** $\pm$ **0.0**	1.48 $\pm$ 1.3
	e^*	0.42 $\pm$ 0.29	0.08 $\pm$ 0.11	**0.73** $\pm$ **0.52**
Boston	e_{ref}	**1.48** $\pm$ **1.75**	0.7 $\pm$ 1.03	3.57 $\pm$ 4.72
	e_{user}	2.26 $\pm$ 2.71	**0.0** $\pm$ **0.0**	2.26 $\pm$ 2.71
	e^*	1.72 $\pm$ 2.09	0.13 $\pm$ 0.19	**2.12** $\pm$ **2.54**
Breast cancer	e_{ref}	**8.82** $\pm$ **9.22**	7.27 $\pm$ 8.25	52.41 $\pm$ 58.63
	e_{user}	22.42 $\pm$ 24.87	**0.0** $\pm$ **0.0**	22.42 $\pm$ 24.87
	e^*	10.74 $\pm$ 9.85	1.25 $\pm$ 1.33	**18.24** $\pm$ **16.83**

TABLE 4.6 – Moyenne et écart-type des métriques $P_x(e)$, $I_x(e,E)$ et $cost_{x,E}(e)$, respectivement définies dans les équations (4.1), (4.2) et (4.4) pour les trois méthodes considérées et les trois jeux de données pour les instances telles que les trois explications existent.

associée à ces instances comme nous avons pu l'observer dans la section précédente. Cela concerne 20% des cas pour les données Half-moons, 0% des cas pour les données Boston, et 11% des cas pour les données Breast cancer. Pour les autres instances, c'est-à-dire les instances pour lesquelles les trois exemples contre-factuels e_{ref}, e_{user} et e^* existent, le tableau 4.6 présente la moyenne et l'écart-type des valeurs de la pénalité, de l'incompatibilité et de la fonction coût associée aux trois approches.

Nous observons, comme attendu, que l'exemple contre-factuel proposé e^* a une valeur de pénalité supérieure à celle de e_{ref} mais inférieure à celle de e_{user}. De plus, la valeur de l'incompatibilité est plus faible que celle de e_{ref}. Enfin, la fonction de coût associée à e^* est la plus faible. Il est intéressant de noter que l'écart-type est élevé, cela est dû au fait que les instances des données test sont très variées. Elles appartiennent à différentes classes, se situent dans différentes zones de l'espace des données et se situent à différentes distances de la frontière. Les plages de valeurs pour les deux critères sont alors grandes.

	e_{ref}	e_{user}	e^*
Half-moons	0.06 ± 0.04	0.25 ± 0.11	0.15 ± 0.10
Boston	0.17 ± 0.23	0.22 ± 0.19	0.30 ± 0.47
Breast Cancer	0.69 ± 0.65	0.19 ± 0.09	1.61 ± 1.57

TABLE 4.7 – Temps d'exécution moyen (en s), des trois approches considérées pour obtenir e_{ref}, e_{user} et e^* pour les trois jeux de données

4.4.4 Temps de calcul

Le tableau 4.7 montre les temps d'exécution pour obtenir les trois exemples contre-factuels. Comme attendu, pour les trois ensembles de données, nous remarquons que le temps associé à e^* est plus élevé que celui de e_{ref} . Puisqu'un seul attribut est pris en compte, les exemples contre-factuels de l'utilisateur sont plus éloignés et nécessitent plus d'itérations pour être identifiés, ce qui augmente le temps d'exécution. Pour les deux ensembles de données Boston et Breast Cancer, le temps associé à e^* est plus élevé que les deux autres. Plus la dimension du jeu de données est élevée, plus le temps d'exécution est long. Comme dans l'évaluation des métriques, nous notons également que les écarts-types sont élevés car les instances étudiées sont associées à différentes classes, se situent dans différentes zones de l'espace des données et se situent à différentes distances de la frontière. Le nombre d'étapes nécessaires pour atteindre la frontière de décision peut varier selon l'instance considérée.

4.4.5 Comparaisons du coût

Dans les expérimentations précédentes, nous avons montré dans les tableaux 4.5 et 4.6 que la valeur moyenne associée à la fonction de coût est minimale pour e^*. L'écart-type obtenu ne permet pas de montrer que KICE permet bien de minimiser la fonction de coût considérée. Cette section compare pour l'ensemble de données Half-Moons, les valeurs de la fonction de coût pour chaque explication associée à chaque instance de l'ensemble test. Pour vérifier expérimentalement que KICE minimise la fonction de coût considérée par rapport aux deux autres méthodes, la figure 4.3 présente la valeur de la fonction de coût associée à e^* par rapport à la valeur prise pour e_{ref} (à gauche) et e_{user} (à droite), pour chacune des instances de test. Sur les figures, nous représentons les droites $y = x$. Sur la figure de gauche, si les points sont au-dessus de cette droite alors la fonction de coût est plus élevée pour e_{ref} que e^* et inversement si les points sont en-dessous. De même, sur la figure de droite, si les points sont au-dessus de cette droite alors la fonction de coût est plus élevée pour e_{user} que e^* et inversement si les points sont en-dessous. Sur les deux figures, on remarque comme attendu que tous les points sont au-dessus de la droite $y = x$, la fonction de coût est alors plus faible pour e^* pour toutes les instances de l'ensemble test.

Sur le graphique de droite, les points sont plus dispersés mais restent au-dessus de la ligne, ce qui montre que de nombreuses explications e_{user} ont un coût élevé par rapport à celui associé à e^*. Nous remarquons que les instances contre-factuelles générées sont

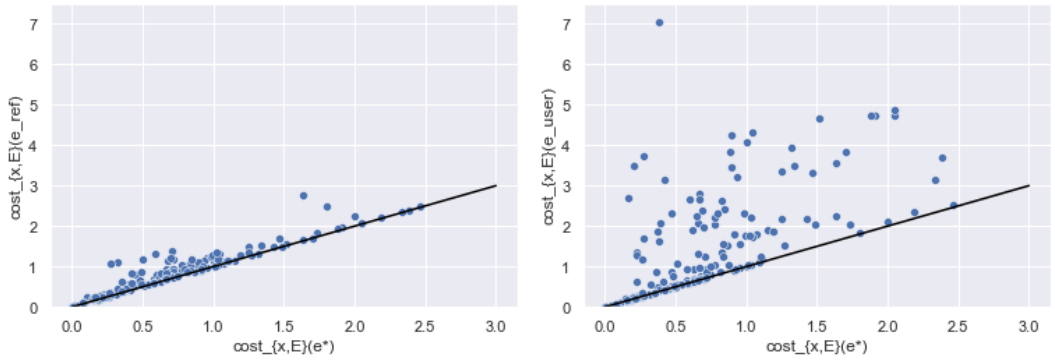


FIGURE 4.3 – Fonctions de coût *cost* définies par l'équation (4.4) de e_{ref}, e_{user} et e^* pour les 80% des données test pour lesquelles les trois explications contre-factuelles sont définies.

plus proches de la fonction de coût de e_{ref} que de e_{user}. La connaissance de l'utilisateur contient un seul attribut, il est difficile d'obtenir une instance contre-factuelle proche uniquement en modifiant cet attribut. Il est alors nécessaire de s'éloigner davantage de l'instance étudiée pour en obtenir une explication compatible à 100%. Cette expérimentation montre que la méthode KICE propose une explication avec une fonction de coût plus faible que les deux autres compétiteurs pour toutes les instances de l'ensemble test.

4.5 Discussion

Avec la méthode KICE, nous proposons d'effectuer un compromis entre la pénalité et l'incompatibilité. Comme évoqué dans la section 3.1.2, la prise en compte de plusieurs critères présente des risques, notamment proposer une explication qui n'optimise pas tous les critères. Dans cette section, nous présentons deux perspectives à notre proposition. Tout d'abord, nous discutons du choix du paramètre λ, qui est lié à la problématique de l'agrégation, cette question sera discutée plus en détail avec des sémantiques plus riches dans le chapitre 6 et 7. Ensuite, une seconde discussion porte sur l'impact de l'accord entre le modèle et les connaissances.

4.5.1 Choix du paramètre λ

Dans ce chapitre, nous proposons d'effectuer un compromis entre la pénalité et l'incompatibilité avec une moyenne pondérée : un poids λ est associé à l'incompatibilité. Il y a alors une question importante du choix de λ que nous avons fixé arbitrairement dans les expérimentations. Le choix de ce paramètre peut être effectué par l'utilisateur, mais concrètement cette tâche est difficile car l'impact de la valeur de λ sur l'explication finale est inconnu. Dans cette section, nous présentons les impacts du choix de ce paramètre dans l'espace des critères et dans l'espace des données.

Exemples illustratifs Nous étudions pour deux instances données les exemples contre-factuels associés à différentes valeurs de λ. La figure 4.4 représente dans l'espace des

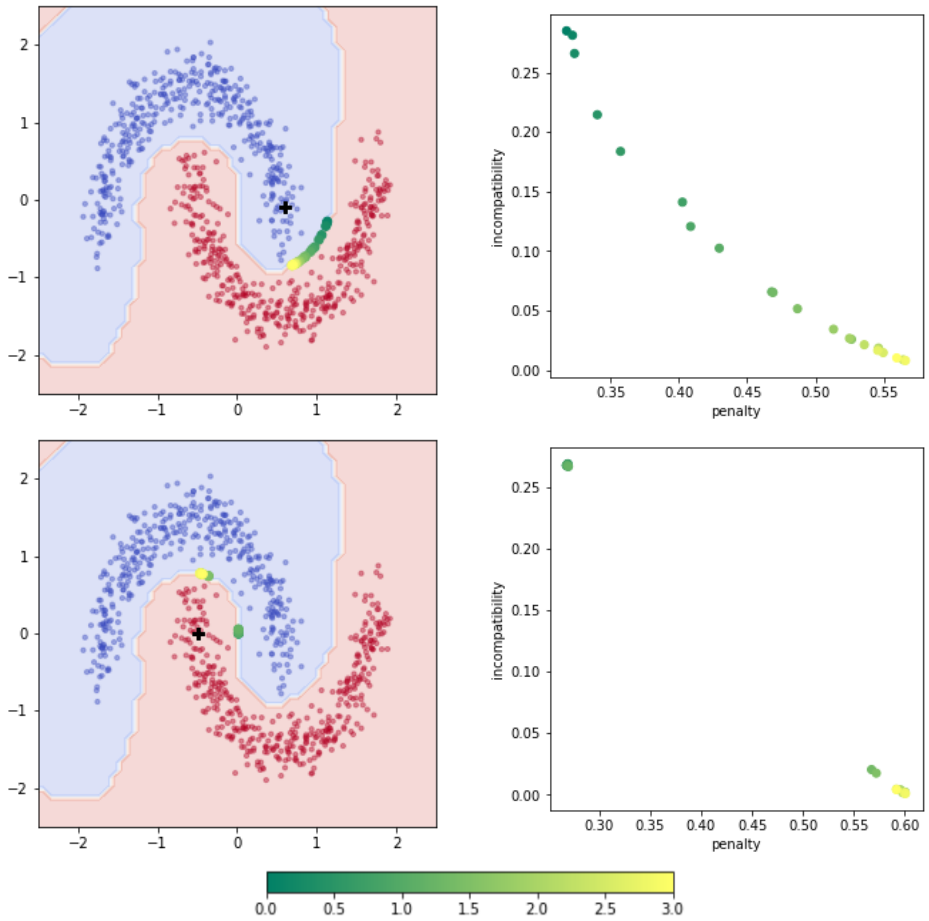


FIGURE 4.4 – Exemples contre-factuels pour différentes valeurs de λ pour deux instances x. A gauche dans l'espace des données, à droite dans l'espace des critères.

critères les exemples contre-factuels obtenus. Les figures de gauche et de droite présentent respectivement les explications dans l'espace des données et dans l'espace des critères. Nous considérons pour chaque instance onze valeurs de λ à intervalle régulier entre 0 et 3. Cet intervalle est délimité par les explications extrêmes : l'explication la plus proche (e_{ref}) associée à $\lambda = 0$ et l'explication totalement compatible (e_{user}) associée à $\lambda = 3$. Ces explications sont générées avec la méthode KICE.

Sur les figures du haut, pour la première instance x, on remarque que les explications sont faiblement espacées dans l'espace des données et dans l'espace des critères. On observe que plus la valeur de λ augmente plus les explications obtenues sont proches de l'instance de référence. Sur les deux figures, on observe que l'intervalle entre les explications se réduit lorsque la valeur de λ augmente même si l'écart entre les valeurs de λ reste le même. Cela montre que la modification de la valeur de λ n'a pas toujours le même impact sur l'explication finale, ains'il est difficile de connaître l'impact de ce paramètre.

Sur la figure du bas, nous générons les explications pour une autre instance. On remarque ici une grande différence entre les valeurs de 1.2 et 1.5 aussi bien dans l'espace

des données que dans l'espace des critères. Les explications associées à une valeur inférieure à 1.2 sont proches, elles se situent à droite de l'instance étudiée dans l'espace des données et en haut à gauche dans l'espace des critères. Pour les explications associées à une valeur supérieure à 1.5, elles sont très proches : en haut de x dans l'espace des données et en bas à droite dans l'espace des critères. Ainsi, seules deux explications contre-factuelles associées à deux intervalles de λ sont possibles.

Frontière de décision / Front de Pareto Pour expliquer ces résultats, il faut étudier en détail le lien entre la frontière de décision autour de l'instance étudiée et le front de Pareto selon les deux critères. Dans l'espace des données, on remarque que la frontière de décision autour de la première instance est convexe alors qu'autour de la seconde instance, elle est concave. Étant donné qu'on considère un espace en deux dimensions, il est possible de montrer que la convexité dans l'espace des données est conservée dans l'espace des critères, de la même manière pour la concavité. Ainsi, nous obtenons dans l'espace des critères, un front de Pareto convexe dans le premier cas et concave dans le second. Or, Wang et al., 2020 montrent que l'approximation du front de Pareto par les sommes pondérées est intéressante si celui-ci est convexe et très peu conseillé s'il est concave. C'est pourquoi dans les exemples présentés sur la figure 4.4, nous arrivons à donner une approximation du front de Pareto seulement pour la première instance. Il serait intéressant de voir si la conservation de la concavité et de la convexité est valide pour des données de dimension supérieure à deux.

Bien que le choix du paramètre λ ne soit pas simple, dans certains cas l'utilisateur peut le spécifier. En particulier, dans le cas où on a un front de Pareto concave, on obtient seulement deux explications possibles : l'explication la plus proche et l'explication la plus compatible. Si la valeur de λ est faible, l'explication la plus proche est obtenue et si elle est élevée l'explication la plus compatible est obtenue. Un autre cas de figure où il est possible de choisir la valeur du λ est lorsque nous considérons un utilisateur qui a des notions techniques. Il peut alors comprendre la variation de la pénalité et de l'incompatibilité en fonction du modèle et de l'instance étudiée, et aussi l'impact de λ. Il peut alors à partir de ces connaissances, émettre des hypothèses sur l'impact des paramètres. Ainsi, il est possible d'estimer une valeur du paramètre intéressante. Une autre possibilité est de ne pas se restreindre à une seule explication, mais proposer plusieurs explications associées à différentes valeurs de λ. La génération de plusieurs explications est discutée dans le chapitre 7.

4.5.2 Connaissances et modèle en désaccord

Problème Le but de la méthode proposée KICE est de proposer une explication fidèle à la fois au modèle et aux connaissances. Cela est problématique si le modèle et les connaissances ne partagent pas des caractéristiques similaires. A titre illustratif, considérons un exemple en deux dimensions où la frontière de décision est une droite horizontale $X_2 = k$ avec $k \in \mathbb{R}$ et la connaissance utilisateur est $E = \{X_2\}$. Dans ce cas, la seule solution pour changer de classe est de faire varier la valeur de X_2, or cet attribut

pour l'utilisateur est inconnu. L'explication proposée n'est alors pas compatible, elle modifie principalement les attributs inconnus par l'utilisateur plutôt que les attributs connus.

Mesure d'accord Une étape importante est de prévoir dans quel cas de figure on peut observer un désaccord entre le modèle et les connaissances. Nous proposons ici trois définitions possibles pour la notion d'accord.

Une première définition est : "le modèle et la connaissance sont en accord si le modèle se base sur les mêmes attributs que l'utilisateur pour donner sa prédiction". Une manière d'extraire les attributs importants du modèle est d'utiliser une méthode générant des explications sous forme de vecteurs d'importance des attributs comme LIME (Ribeiro et al., 2016) ou SHAP (Lundberg and Lee, 2017). A partir du vecteur obtenu, il est possible de mesurer une similarité entre les attributs ayant un poids important et les attributs donnés par l'utilisateur.

Une deuxième définition est : "le modèle et la connaissance sont en accord s'il existe une explication totalement compatible". Pour mesurer cet accord, une solution est de générer une explication totalement compatible qui modifie uniquement les attributs connus par l'utilisateur, notée e_{user} dans ce chapitre. Si cette solution existe alors il y a accord.

Une dernière définition proche de la précédente est : "le modèle et la connaissance sont en accord si l'explication la plus compatible est proche de l'instance étudiée". Pour mesurer cet accord, une solution est de comparer les deux explications extrêmes : l'explication la plus proche e_{ref} et l'explication la plus compatible e_{user}. Si e_{user} a une pénalité élevée ou que e_{ref} a une incompatibilité élevée, il y a désaccord.

Procédure choisie Une fois que la mesure d'accord est définie, s'il y a désaccord, il faut considérer une autre procédure que KICE pour générer l'explication. Deux solutions sont possibles. La première consiste à demander à l'utilisateur d'apprendre de nouvelles connaissances avant de lui proposer à nouveau une explication. La seconde consiste à choisir entre le modèle et la connaissance, afin de proposer une explication fidèle au modèle ou à la connaissance.

Cette section a discuté de la difficulté à définir l'accord entre le modèle et la connaissance. L'étude ne se résume pas à définir cette mesure, il faut également étudier quelle explication sera la plus adaptée. Cette étude représente une perspective très intéressante pour de futurs travaux.

4.6 Bilan

Dans le chapitre 3, nous avons introduit un formalisme pour intégrer des connaissances utilisateur. Ce chapitre en a proposé une instanciation pour des explications contre-factuelles et des connaissances sous formes d'ensemble d'attributs. Le but est de générer une explication personnalisée qui favorise la modification des attributs connus

par l'utilisateur. Pour résoudre le problème d'optimisation considéré nous avons proposé un nouvel algorithme nommé *Knowledge Integration in Counterfactual Explanations* (KICE).

Plusieurs améliorations et extensions à ce chapitre existent, elles sont notamment étudiées dans les chapitres 5 et 6. Une première est l'étude de ce formalisme pour le même type de connaissances mais des explications sous forme de vecteurs d'importance des attributs. Une seconde extension considère le même type d'explications que KICE mais avec des connaissances sous forme de système de règles. Le chapitre 6 quant à lui se concentre sur le choix de la fonction d'agrégation pour avoir un comportement plus expressif que la moyenne pondérée considérée par KICE.

Chapitre 5

Instanciation des critères dans de nouveaux cadres

Ce chapitre propose d'instancier le formalisme général d'intégration des connaissances dans la génération d'explications proposé dans le chapitre 3 pour deux configurations différentes de celle présentée dans le chapitre 4. La première considère un type d'explications différent des exemples contre-factuels, les vecteurs d'importance des attributs, pour le même type de connaissance exprimée par un ensemble d'attributs : le but est de fournir une explication qui accorde des poids importants aux attributs connus par l'utilisateur, afin d'avoir une explication qui soit, autant que possible, dans le langage de l'utilisateur. La seconde configuration considère une autre forme de connaissances, exprimées par des règles pour le même type d'explication, les exemples contre-factuels : l'objectif est de proposer un exemple contre-factuel dans la zone de l'espace où l'utilisateur a une connaissance associée à l'instance x. Le chapitre décrit les deux approches proposées, respectivement appelées *Knowledge Integration in Surrogate Models* (KISM) et *Rule Knowledge Integration in Counterfactual Explanation* (rKICE). Le chapitre présente également les études expérimentales menées pour évaluer ces approches.

Ce travail a été présenté dans l'article *A General Framework for Personalising Post Hoc Explanations through User Knowledge Integration* publié dans le journal IJAR 2023 (Jeyasothy et al., 2023a).

5.1 KISM : Knowledge Integration in Surrogate Models

Dans cette section, nous présentons une instanciation du formalisme général proposé dans le chapitre 3 pour un autre type d'explications que les explications contre-factuelles considérées dans le chapitre 4 : les vecteurs d'importance des attributs. Nous considérons le même type de connaissances utilisateur que KICE, c'est-à-dire un ensemble d'attributs. Nous considérons le cas où les vecteurs d'importance sont extraits d'un modèle de substitution, selon le processus utilisé par la méthode LIME (Ribeiro et al., 2016) rappelée dans la section 2.6.3. Après avoir présenté la configuration étudiée dans cette section, nous décrivons dans la section 5.1.2 la fonction de coût étudiée. Puis, nous présentons dans la section 5.1.3 l'algorithme proposé, nommé Knowledge

Integration in Surrogate Models (KISM) qui optimise cette fonction de coût. Enfin, nous détaillons les expérimentations effectuées pour valider l'algorithme proposé.

5.1.1 Configuration étudiée

Dans cette section, le classifieur à expliquer est une fonction $f : \mathcal{X} \longrightarrow [0,1]^k$, où $f(x)$ est un vecteur de dimension k qui donne la probabilité de chacune des classes pour toute instance x. Nous considérons une classification binaire où $k = 2 : f(x) \in [0,1]$ donne la probabilité que la classe de x soit 1. La probabilité que la classe de x soit de la classe 0 est de $1 - f(x)$. De plus, nous considérons des données tabulaires : $\mathcal{X} = \mathbb{R}^d$.

Nous notons $\mathcal{E}$ l'ensemble des modèles de substitution candidats $e : \mathcal{X} \longrightarrow [0,1]$. Des définitions pour $\mathcal{E}$ sont par exemple un ensemble de modèles linéaires ou un ensemble d'arbres de décision. Pour tout $e \in \mathcal{E}$, et pour toute instance $z \in \mathcal{X}$, $e(z)$ désigne la probabilité de prédire la classe 1 par le modèle de substitution pour z.

A partir du modèle de substitution, des explications sous forme de vecteurs d'importance sont extraites, notées $\widetilde{g} = (w_1, w_2, ..., w_d)$. Dans le cas où les modèles de substitution sont des modèles linéaires, c'est-à-dire si $\mathcal{E} = \{\sum_i w_i x_i + w_0, W \in \mathbb{R}^{d+1}\}$, les poids du vecteur d'importance correspondent directement aux coefficients de chaque attribut. Lorsque l'on considère les arbres de décision, ces poids ne sont pas explicites. Une possibilité proposée par Kazemitabar et al., 2017 pour les définir est de sommer les réductions d'impureté sur les nœuds de l'arbre où la séparation se fait selon l'attribut étudié.

5.1.2 Instanciation du cadre général : fonction de coût proposée

Cette section présente successivement les trois composantes de la fonction de coût générale, à savoir les fonctions de pénalité, d'incompatibilité ainsi que l'opérateur d'agrégation, en utilisant les mêmes notations que la section 4.1.

Fonction de pénalité Comme pour le cas des explications contre-factuelles, la pénalité est mesurée par la définition classique des modèles de substitution sans connaissances utilisateurs. Ainsi, comme décrit dans la section 2.6, le problème d'optimisation associé aux modèles de substitution est le suivant (Ribeiro et al., 2016; Jia et al., 2019) :

$$e^* = \underset{e \in \mathcal{E}}{\operatorname{argmin}}\, L(f, e, \pi_x) + \Omega(e) \tag{5.1}$$

Pour notre étude, nous choisissons la fonction de coût considérée par LIME qui est une méthode de référence. Cette fonction mesure l'erreur quadratique du modèle de substitution e par rapport au modèle f à expliquer sur un ensemble $\mathcal{Z}_x$:

$$penalty_x(e, f) = \sum_{z \in \mathcal{Z}_x} (f(z) - e(z))^2 + \Omega(e) \tag{5.2}$$

Le premier terme traduit la fidélité locale du modèle de substitution à f et le second terme $\Omega(e)$ traduit la complexité de l'explication candidate e, comme décrit dans la section 2.6. Ce dernier dépend du type de classifieurs de substitution. Dans le cas des modèles de régression linéaire tels que l'approche LIME (Ribeiro et al., 2016), la complexité $\Omega(e)$ peut être définie par les normes l_0, l_1 ou l_2 des coefficients de régression, selon qu'une régression simple, lasso ou ridge est effectuée. Dans la suite, nous considérons le cas de la norme l_2, c'est-à-dire $\Omega(e) = \sum_{i=1}^{d} w_i^2$.

Bien que la méthode LIME propose une explication locale, elle génère des instances globalement dans l'espace des données, puis intègre la localité en associant des poids différents aux instances générées selon leur proximité à l'instance étudiée. Plutôt que de considérer une fonction π_x qui associe des poids aux instances, nous définissons la localité en générant les instances proches de l'instance étudiée, c'est-à-dire que nous considérons un rayon faible de la boule autour de x.

Fonction d'incompatibilité Nous interprétons la connaissance de l'utilisateur E comme sa définition d'attributs importants. Nous proposons alors d'interpréter l'intégration de la connaissance pour la formulation d'explication dans le langage de l'utilisateur comme l'objectif de favoriser la pertinence de ces attributs dans l'explication, c'est-à-dire d'augmenter les coefficients associés à ces variables : le principe de cette intégration consiste à exploiter les corrélations entre les attributs. Lorsque deux attributs sont corrélés et que l'un d'eux a un coefficient élevé, alors on cherche à attribuer un coefficient important à l'attribut connu par l'utilisateur.

Considérons, pour illustrer ce principe, l'exemple du classifieur de légumes et le cas de la prédiction d'une carotte présenté dans la section 3.4. Deux caractéristiques sur lesquelles la classification est basée sont liées : le taux de provitamine A et la couleur, car une présence importante de provitamine A implique une couleur orange. On peut alors comparer les deux explications e_1 : "le taux de provitamine A a un coefficient élevé" et e_2 : "la couleur a un coefficient important". e_1 peut ne pas être compréhensible par un utilisateur non expert pour lequel $E = \{couleur\}$, car elle met tout le poids sur un attribut que l'utilisateur ne connaît pas. Au contraire, e_2 constitue une explication qui est équivalente à e_1, étant donné la corrélation entre les attributs et sa compatibilité avec la connaissance de l'utilisateur. Nous souhaitons alors favoriser e_2, c'est-à-dire l'importance de l'attribut dans l'explication. Dans l'apprentissage, cela n'a pas d'importance étant donné que ces deux attributs sont corrélés, ainsi un modèle qui accorde de l'importance à la provitamine A accorde de l'importance à la couleur orange.

Pour favoriser les attributs E, nous proposons de pénaliser les coefficients associés aux attributs qui n'appartiennent pas à E en utilisant une pénalité l_2. Pour faciliter la résolution du problème, nous avons choisi la même norme que celle qui intervient dans la fonction de pénalité dans la fonction Ω. Ainsi, nous proposons de définir la fonction d'incompatibilité comme :

$$I_x(e, E) = \sum_{i \in \overline{E}} w_i^2 \tag{5.3}$$

avec w_i les poids associés aux attributs et $\overline{E}$ les attributs non présents dans la connaissance de l'utilisateur. L'incompatibilité vaut 0 si le modèle de substitution n'accorde aucune importance aux attributs non présents dans la connaissance E. Inversement, l'incompatibilité prend une valeur élevée si le modèle de substitution accorde peu d'importance aux attributs E et une grande importance aux attributs inconnus.

Fonction d'agrégation Comme dans le cas des explications contre-factuelles personnalisées (chapitre 4), nous proposons de formuler le nouvel objectif d'explication comme un compromis entre la fidélité et la compatibilité de l'explication, par le biais d'une somme pondérée.

Fonction globale Par conséquent, le problème d'optimisation proposé est le suivant :

$$e^* = \underset{e \in \mathcal{E}}{\operatorname{argmin}}\, cost_{x,E,f}(e) \tag{5.4}$$

$$\text{avec } cost_{x,E,f}(e) = \sum_{z \in \mathcal{Z}_x} (f(z) - e(z))^2 + \sum_{i=1}^{d} w_i^2 + \lambda \sum_{i \in \overline{E}} w_i^2 \tag{5.5}$$

où λ est un réel positif qui définit l'importance relative de la fidélité et de la compatibilité.

5.1.3 Algorithme proposé

Dans cette section, nous présentons l'algorithme proposé pour résoudre le problème défini par l'équation (5.5), dans le cas où les modèles de substitution considérés sont linéaires. Tout d'abord, nous présentons le principe général de la méthode en décrivant les principales étapes utilisées. Puis, nous détaillons l'étape d'entraînement du modèle.

Principe Pour résoudre le problème d'optimisation nous proposons l'algorithme KISM qui utilise le principe d'approximation locale proposé par LIME autour de l'instance donnée, mais il ne considère pas tous les attributs au même niveau : il accorde plus d'importance aux attributs de l'utilisateur.

KISM est constitué de trois étapes comme LIME. Tout d'abord, un ensemble d'instances est généré autour de l'instance étudiée x suivant le même principe que LIME, comme rappelé dans la section 2.6.3. Dans la seconde étape, un modèle de substitution linéaire est entraîné : comme détaillé ci-dessous, nous proposons une version modifiée de LIME qui tient compte du terme additionnel de l'équation 5.5. Enfin, la troisième étape est triviale, étant donné que nous considérons un modèle linéaire, l'explication finale est l'ensemble des coefficients.

Entraînement du modèle de substitution L'algorithme KISM s'inspire de l'algorithme LIME. La différence intervient lors de l'étape d'entraînement du modèle de régression.

La méthode LIME détaillée dans la section 2.6.3 résout le problème :

$$W = (X^t X + \alpha I)^{-1} X^t y$$

où α est un paramètre de complexité de la régression, X représente les instances générées dans le voisinage $\mathcal{Z}_x$, y les classes $f(x)$ pour tout $x \in \mathcal{Z}$ et I la matrice d'identité.

La fonction de coût définie dans l'équation (5.5) se présente comme l'équation présentée ci-dessus, avec une pénalisation supplémentaire pour intégrer la connaissance utilisateur, associée à un coefficient λ qui pondère les attributs qui ne sont pas connus par l'utilisateur. Ainsi, en notant $I_{\overline{E}}$ la matrice diagonale contenant 1 uniquement sur les lignes associées aux attributs inconnus, le problème d'optimisation considéré peut être réécrit :

$$(X^t X + \alpha I + \lambda I_{\overline{E}})^{-1} X^t y$$

Cette équation montre explicitement l'intégration des connaissances au sein de la méthode par l'ajout de la pénalité λ par rapport à la connaissance E. En particulier, il est facilement observable qu'elle n'est pas coûteuse, car le nouveau terme associé à la connaissance n'induit pas de changement dans la complexité du problème.

5.1.4 Étude expérimentale

Cette section décrit trois expériences menées pour étudier les propriétés de la méthode KISM. Nous présentons le protocole expérimental qui décrit les paramètres, en particulier les connaissances utilisateurs, et les compétiteurs étudiés. Comme dans le chapitre 4, nous considérons tout d'abord un exemple illustratif dans le cas du jeu de données Californie, puis, dans le cas des données Half-Moons. Enfin, nous évaluons la méthode proposée selon différentes métriques quantitatives pour les mêmes jeux de données que le chapitre 4.

5.1.4.1 Protocole expérimental

Nous considérons pour ces expérimentations un protocole similaire à celui de la section 4.3 avec les mêmes jeux de données. Nous définissons dans cette section les connaissances utilisateur choisies et les compétiteurs considérés.

Paramètres L'impact de l'intégration des connaissances utilisateur est évalué en examinant les évolutions du rang des attributs selon la valeur absolue du poids qui leur est associée, afin d'étudier s'il diminue pour les attributs qui font partie de E. En effet, comme discuté lors de la définition de la fonction d'incompatibilité, il est attendu que les attributs de E soient les attributs les plus importants d'après les valeurs $|w_i|$. Pour faciliter l'interprétation des résultats, nous considérons pour tous les jeux de données autres que Californie que l'utilisateur fournit un seul attribut et que nous examinons le rang de cet attribut. Généraliser cette expérience à plusieurs attributs utilisateur rendrait difficile l'agrégation des rangs qui leur sont associés.

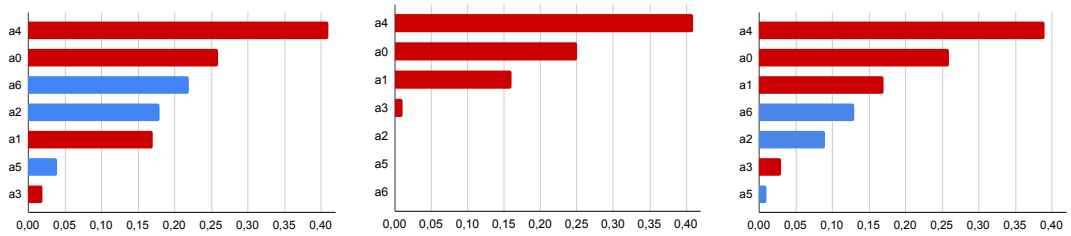


FIGURE 5.1 – Poids des vecteurs d'importance fournis par e_{ref} (gauche) qui correspond à LIME, e_{user} (milieu) et e^* (droite) qui correspond à l'approche proposée KISM pour une instance x du jeu de données Californie. Les attributs connus E sont représentés en rouge et les attributs inconnus $\overline{E}$ en bleu.

Ainsi, pour Half-Moons, à l'exception des exemples illustratifs donnés dans la section 5.1.4.3, nous fixons $E = \{1\}$, pour Boston, $E = \{7\}$ et pour Breast Cancer, $E = \{3\}$. Par contre, pour le jeu de données Californie, nous choisissons $E = \{longitude, latitude, nombre\ de\ pieces, nombre\ de\ chambres\}$.

En ce qui considère le choix du paramètre λ, il est guidé par l'observation des échelles des valeurs des deux termes de pénalité et d'incompatibilité de l'équation (5.5) : la première est très élevée par rapport à la seconde. Ainsi, pour obtenir des valeurs de même ordre de grandeur nous fixons $\lambda = 500$. Les paramètres LIME sont fixés à leurs valeurs par défaut et nous considérons les mêmes classifieurs que décrits dans le protocole expérimental mis en œuvre dans la section 4.4.

Compétiteurs Comme dans le chapitre précédent, nous comparons l'explication proposée à deux concurrents. Le premier noté e_{ref} correspond au cas extrême où $\lambda = 0$: il résout le problème d'optimisation de référence qui minimise uniquement la fonction de pénalité. Aussi e_{ref} correspond à l'explication obtenue par LIME (Ribeiro et al., 2016). Le second concurrent correspond à l'autre cas extrême, de l'explication totalement compatible, définie par le modèle de substitution qui ne considère que les attributs utilisateur : les attributs non connus $\overline{E}$ ont un coefficient forcé à 0. Nous notons e_{user} l'explication qui résout le problème d'optimisation suivant :

$$e_{user} = \underset{e \in \mathcal{E}}{\operatorname{argmin}} \sum_{z \in \mathcal{Z}_x} (f(z) - e(z))^2 + \sum_{i=0}^{d} w_i^2 \ \text{ avec } \ \forall i \in \overline{E},\ w_i = 0$$

Les poids w_i des attributs inconnus sont forcés à prendre la valeur 0, nous considérons donc des modèles de substitution uniquement entraînés sur les attributs E. Ce problème revient à l'équation (5.5) avec une valeur de λ arbitrairement grande.

5.1.4.2 Exemple de résultats de KISM sur la base Californie

La première expérience vise à observer les résultats obtenus par la méthode proposée et les compétiteurs pour un exemple réaliste. La figure 5.1 montre les résultats obtenus pour une instance choisie du jeu de données Californie. Nous considérons

	$P_x(e,f)$	$I_x(e,E)$	$cost_{x,E,f}(e)$
e_{ref}	377.94	0.35	1069.44
e_{user}	480.27	0.26	1001.63
e^*	417.36	0.28	962.62

TABLE 5.1 – Métriques associées aux résultats e_{ref}, e_{user} et e^* pour une instance x du jeu de données Californie.

pour cet exemple une connaissance différente du chapitre 4 : $E = \{longitude, latitude, nombre\ de\ pieces, nombre\ de\ chambres\}$. Ainsi, le but de l'utilisateur est d'avoir une explication qui dépend de la localisation, du nombre de pièces et de chambres de la maison car ce sont des attributs qu'il connaît et qu'il comprend. Il attendu que ces caractéristiques aient avoir un poids élevé dans l'explication.

La figure 5.1 présente les résultats associés aux explications e^*, e_{user} et e_{ref} obtenues pour une instance x prise aléatoirement dans le jeu de données test. Chaque graphique présente en ordonnée les attributs qui sont classés dans l'ordre décroissant selon la valeur absolue de leur poids. Les attributs de E et de $\overline{E}$ sont distingués par couleur, rouge pour les premiers et bleue pour les seconds. Le tableau 5.1 présente les valeurs des métriques associées à chacune de ces explications.

Tout d'abord, sur la figure de gauche qui représente l'explication de référence obtenue avec la méthode LIME, les attributs de E se situent à la première, seconde, cinquième et dernière place. On remarque que la longitude de la maison (a_0) et le nombre de chambres (a_4) sont bien prises en compte dans la prédiction, ils se situent à la première et seconde place. Cependant, les autres variables importantes sont le revenu des habitants (a_6) et l'âge médian des logements (a_2), ce qui dans notre cas ne correspond pas à des variables connues par l'utilisateur.

La figure du milieu représente l'explication la plus compatible e_{user} qui considère uniquement les attributs présents dans les connaissances. On remarque bien sur le graphique que les poids associés aux attributs inconnus sont nuls, les autres attributs ont un poids nul. Le classement proposé est alors le suivant : nombre de chambres, longitude, latitude et nombre de pièces. Ce classement reprend celui obtenu avec e_{ref} sans les attributs inconnus. Cela ne convient pas car cette personnalisation s'effectue au prix d'une diminution de la fidélité du modèle, comme indiqué dans le tableau 5.1. Comme discuté dans la section 3.1.2, nous obtenons ici un cas où seule la notion de personnalisation est considérée, on propose à l'utilisateur une explication qui convient à ses connaissances indépendamment de la fidélité au classifieur à expliquer.

Enfin, la figure de droite présente l'explication e^* fournie par KISM. Le compromis entre les deux explications précédentes est bien visible, notamment les attributs a_1 et a_3, c'est-à-dire la latitude et le nombre de pièces sont remontés dans le classement, le premier est passé de la cinquième à la troisième place et le second est monté d'une place.

Nous comparons dans le tableau 5.1 les valeurs des trois métriques : $P_x(e,f)$, $I_x(e,E)$ et $cost_{x,E,f}(e)$ associées aux explications présentées sur la figure 5.1. Comme attendu les explications e_{ref}, e_{user} et e^* minimisent respectivement la pénalité, l'incompatibilité et la

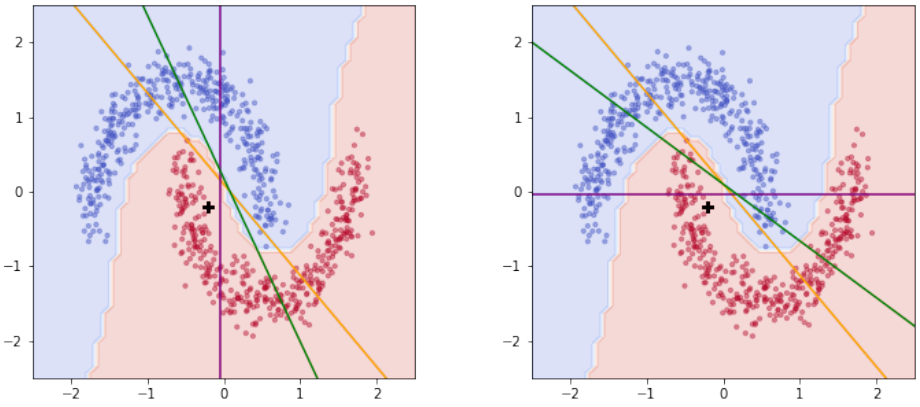


FIGURE 5.2 – Exemples des résultats e_{ref}, e_{user} et e^* pour une instance x et différentes connaissances utilisateur : à gauche $E = \{X_1\}$, à droite $E = \{X_2\}$. (+ : x, — : e_{ref}, — : e^*, — : e_{user})

fonction de coût. La perte de fidélité entre e_{ref} et e_{user} est de 21% alors qu'elle est de 10% entre e_{ref} et e^*. Pour l'incompatibilité, la perte entre e_{user} et e^* est de 7% contre 26% entre e_{user} et e_{ref}. Ainsi, on remarque que l'explication e^* est associée à une perte faible sur la qualité pour un gain au niveau de la compatibilité, ce qui correspond bien au résultat souhaité.

5.1.4.3 Exemples illustratifs sur la base Half-Moons

Dans cette seconde expérience, nous illustrons le comportement des méthodes sur le jeu de données en deux dimensions Half-Moons également utilisé pour KICE (voir section 4.4.2), pour une instance x représentée par un + et différentes connaissances E. Cette section visualise les explications classiques et lorsque nous intégrons la connaissance avec KISM pour la même donnée $x = (-0.1, -0.1)$. La figure 5.2 montre les frontières de décision (correspondant à une probabilité de 0.5) des modèles de substitution : e^* (en vert) obtenue par KISM, LIME (en orange) qui constitue le modèle de substitution de référence, qui ne tient pas compte de la connaissance utilisateur, et e_{user} (en violet), complètement compatible. Sur le graphique de gauche, la connaissance de l'utilisateur considérée est $E = \{X_1\}$ et sur le graphique de droite, elle correspond à l'autre attribut $E = \{X_2\}$.

Dans les deux graphiques, le modèle de substitution classique e_{ref} (en orange), qui ne prend en compte aucune connaissance de l'utilisateur, fournit le même résultat. L'explication e_{user} dépend d'un seul attribut, elle correspond donc à une droite verticale sur la figure de gauche et à une droite horizontale sur la figure de droite. A gauche, on observe que le poids selon l'attribut X_1 est plus élevé pour la droite verte qui correspond à e^* que pour la droite orange associée à e_{ref}. Sur le graphique de droite, on observe que le poids selon l'attribut X_2 est plus élevé pour la droite verte que pour la droite jaune. Ainsi, on remarque que la droite verte associée à KISM est entre les deux autres droites

	e_{ref}	e^*	Gain de classement	e_{user}
Half-Moons	1.63 ± 0.48	1.55 ± 0.5	0.09 ± 0.28	1 ± 0.0
Boston	6.08 ± 2.89	4.49 ± 2.74	1.58 ± 1.35	1 ± 0.0
Breast Cancer	13.18 ± 5.19	8.51 ± 4.86	4.67 ± 2.46	1 ± 0.0

TABLE 5.2 – Classement moyen, avec écart-type, de l'attribut défini par l'utilisateur, pour les trois méthodes considérées et les trois jeux de données, ainsi que le gain de classement entre e^* et e_{ref}.

sur les deux figures. La méthode KISM proposée est donc utile, car elle permet effectivement d'accorder plus d'importance aux attributs connus qu'aux attributs inconnus dans l'explication. L'explication proposée intègre au mieux la connaissance utilisateur sans sacrifier la fidélité du modèle au classifieur à expliquer.

5.1.4.4 Évaluation de la méthode

La troisième expérience évalue quantitativement la méthode proposée sur les jeux de données Half-Moons, Boston et Breast Cancer en faisant varier l'instance à expliquer x. Tout d'abord, pour les explications obtenues pour chaque instance de l'ensemble test, nous classons les attributs de l'explication par ordre décroissant de la valeur absolue des poids. Ensuite, nous étudions la position dans ce classement de l'unique attribut de l'utilisateur tel que défini dans la section 5.1.4.1. Enfin, nous comparons les explications selon les mêmes métriques que précédemment, c'est-à-dire $P_x(e, f)$, $I_x(e, E)$ et $cost_{x,E,f}(e)$ respectivement définies par les équations (5.2), (5.3) et (5.5).

Les explications sont générées pour l'ensemble des instances du jeu de données test. Le tableau 5.2 présente le classement moyen (avec écart-type) de l'attribut connu par l'utilisateur dans l'explication. Les première, seconde et dernière colonnes concernent respectivement les explications e_{ref}, e^* et e_{user}. Comme dans le chapitre 4, les écarts-types sont très élevés, cela est dû au fait qu'il y a une grande variété des instances x étudiées : elles se situent à différentes positions de la frontière de décision ou les frontières de décision près de l'instance ont des formes différentes.

Les explications e_{user} considèrent uniquement l'attribut utilisateur, comme attendu l'attribut se trouve à la première place ; par contre cette compatibilité maximale se fait au prix d'une faible fidélité au modèle (voir tableau 5.3). Nous nous concentrons alors, la troisième colonne, sur le gain de position de l'explication proposée par rapport à l'explication de référence e_{ref} fournie par LIME qui ne tient pas compte des connaissances utilisateur. On observe que l'attribut étudié remonte en effet dans le classement en moyenne, le gain de classement moyen est compris entre 0.09 pour Half-Moons et 4.67 pour Breast cancer. Cet effet est plus visible pour les jeux de données comportant plus d'attributs : dans le cas de Half-Moons en 2D, seules deux valeurs de rang sont possibles, ce qui rend plus difficile la modification des valeurs. Dans l'ensemble, cela montre que la méthode KISM proposée permet d'augmenter l'importance accordée aux attributs connus par l'utilisateur.

		$P_x(e,f)$	$I_x(e,E)$	$cost_{x,E,f}(e)$
Half-moons	e_{ref}	**247.44 ± 143.96**	0.07 ± 0.11	284.89 ± 169.13
	e^*	255.08 ± 147.65	0.03 ± 0.04	**268.89 ± 156.44**
	e_{user}	340.89 ± 229.55	**0.0 ± 0.0**	340.89 ± 229.55
Boston	e_{ref}	**333.36 ± 183.75**	0.05 ± 0.06	438.25 ± 273.97
	e^*	358.18 ± 201.49	0.01 ± 0.01	**373.58 ± 213.80**
	e_{user}	398.83 ± 235.61	**0.0 ± 0.0**	398.83 ± 235.61
Breast cancer	e_{ref}	**9.45 ± 16.35**	0.04 ± 0.09	80.91 ± 186.09
	e^*	26.32 ± 54.58	0.01 ± 0.01	**36.85 ± 79.60**
	e_{user}	53.99 ± 120.93	**0.0 ± 0.0**	53.99 ± 120.93

TABLE 5.3 – Moyenne et écart-type des métriques $P_x(e,f)$, $I_x(e,E)$ et $cost_{x,E,f}(e)$ respectivement, définies dans les équations (5.2), (5.3) et (5.5) pour les trois méthodes considérées et les trois jeux de données.

Ensuite, nous étudions les trois méthodes selon les trois critères présentés dans la section 3.2 : pénalité, incompatibilité et fonction de coût. Comme attendu, e_{ref}, e^* et e_{user} minimisent respectivement la fonction de pénalité, d'incompatibilité et de coût. Par rapport à e_{ref}, l'explication proposée e^* présente une pénalité plus élevée mais une incompatibilité plus faible. Par rapport à e_{ref}, e^* présente une incompatibilité plus élevée mais une pénalité plus faible. Ce comportement correspond bien à celui qui est souhaité.

5.2 rKICE : Rule Knowledge Integration in Counterfactual Explanation

Cette section examine une troisième instanciation du formalisme général d'intégration de la connaissance, défini dans le chapitre 3. Comme la méthode KICE proposée dans le chapitre 4, nous considérons les explications sous forme d'exemples contrefactuels, mais nous intégrons une connaissance plus riche qu'un ensemble d'attributs qui s'exprime sous la forme de règles. Tout d'abord, nous présentons la configuration étudiée dans ce chapitre. Puis, nous examinons la définition des trois composantes du cadre général, les fonctions de pénalité, d'incompatibilité et d'agrégation. Nous décrivons ensuite l'algorithme proposé *Rule Knowledge Integration in Counterfactual Explanation* (rKICE) pour résoudre le problème d'optimisation induit. Enfin, nous présentons les expérimentations menées pour examiner la pertinence de l'algorithme proposé.

5.2.1 Configuration étudiée

Nous considérons des connaissances sous formes de règles de décision qui constituent une forme de connaissances plus riche qu'un ensemble d'attributs. Une règle de décision est définie par une prémisse qui contient un ensemble de conditions sur les attributs et une conséquence qui prédit une classe. Plus précisément, la prémisse est définie par un ensemble d'attributs A_E où chacun des attributs $i \in A_E$ est associé à

un intervalle $[v^i_{inf}, v^i_{sup}]$. La conclusion C_E est l'une des classes possibles, par exemple $C_E \in \{0, 1\}$ dans le cas binaire. Une règle s'écrit alors, pour tout $x \in \mathcal{X}$:

$$E : \bigwedge_{i \in A_E} x_i \in [v^i_{inf}, v^i_{sup}] \Longrightarrow C(x) = C_E$$

où $C(x)$ est alors la classe prédite pour l'instance x. Nous faisons l'hypothèse que l'ensemble de règles qui constitue la connaissance considérée respecte deux propriétés principales : il ne couvre pas forcément tout l'espace des données et les espaces associés aux règles sont distincts. La seconde propriété signifie qu'au maximum une seule règle est déclenchée pour l'instance x.

Par rapport aux connaissances sous forme d'ensemble d'attributs, examinées jusqu'à présent dans le chapitre 4 et la section 5.1, l'information fournie est plus précise et plus riche : chaque attribut de A_E est associé à un intervalle (éventuellement plusieurs lorsque E contient plusieurs règles). De plus, les attributs ne sont pas considérés individuellement et indépendamment, puisqu'ils sont combinés conjonctivement dans la prémisse. En outre, une information de classe C_E est associée : la connaissance utilisateur peut être vue comme un classifieur, partiel étant donné qu'elle ne couvre pas toujours tout l'espace de données. Ce classifieur peut être considéré comme très interprétable pour l'utilisateur puisqu'il lui est personnel et prend la forme d'une base de règles : ces règles peuvent être vue comme des explications de la prédiction effectuée par l'utilisateur.

La question de l'accord entre la connaissance et le classifieur, introduite dans la section 4.5.2, se pose alors de façon particulière à la fois en terme de prédiction et d'explication, et cet accord a un impact sur l'explication générée. En effet, lorsque le classifieur à expliquer prédit une classe identique à celle de la règle déclenchée pour l'instance considérée par l'utilisateur, on peut considérer que celui-ci ne sollicite pas d'explication supplémentaire, interprétant sa propre règle comme explication. Par contre, s'il y a désaccord il est plausible que l'utilisateur demande une explication car il se demande : "Pour quelles raisons le modèle prédit différemment ?". Si aucune règle utilisateur n'est déclenchée pour l'instance x considérée, on se ramène au cas de figure où il n'y a pas d'intégration de connaissances et l'explication de référence est proposée. Ainsi, un exemple contre-factuel avec intégration de connaissances est demandé uniquement si la prédiction donnée par la règle déclenchée par l'utilisateur pour l'instance considérée existe et que la prédiction associée est différente de celle donnée par le classifieur à expliquer.

5.2.2 Instanciation du cadre général : fonction de coût proposée

Cette section présente successivement les trois composantes de la fonction de coût générale, les fonctions de pénalité, d'incompatibilité ainsi que l'opérateur d'agrégation, en utilisant les mêmes notations que la section 4.1.

Fonction de pénalité Comme pour le cas des explications contre-factuelles avec la connaissance exprimée sous forme d'ensemble d'attributs, nous considérons pour la

pénalité la fonction de coût classique qui mesure la proximité de l'explication candidate à l'instance étudiée. Comme défini dans le chapitre 4, dans le cas des exemples contre-factuels nous considérons la proximité de l'explication à l'instance étudiée : formellement, la fonction de pénalité est définie comme dans l'équation (4.1).

$$P_x(e) = \| x - e \|^2 \tag{5.6}$$

Fonction d'incompatibilité Étant donné une instance x dont nous souhaitons expliquer la prédiction, nous considérons une seule règle déclenchée E, telle que $C_E \neq f(x)$. La prémisse de cette règle décrit une région où l'utilisateur a une connaissance sur la classe associée. Nous souhaitons alors que l'exemple contre-factuel proposé se situe dans la région définie par la prémisse. La règle appliquée à x peut être appliquée à l'exemple contre-factuel généré noté e, la prédiction de l'utilisateur pour x et e est alors la même : $C(x) = C(e)$.

Nous proposons donc de pénaliser les candidats contre-factuels dont les valeurs des attributs ne satisfont pas les contraintes de la règle associée. Formellement, la fonction d'incompatibilité est donc définie comme :

$$I_x(e, E) = \sum_{i \in A_E} (x_i - e_i)^2 \times \mathbb{1}_{e_i \notin [v^i_{inf}, v^i_{sup}]} \tag{5.7}$$

avec A_E l'ensemble des attributs présents dans la prémisse de la règle E et $[v^i_{inf}, v^i_{sup}]$ l'intervalle que la règle associe à l'attribut i. Dans le cas où le candidat contre-factuel appartient à la région définie par la prémisse, pour tout i on a $\mathbb{1}_{e_i \notin [v^i_{inf}, v^i_{sup}]} = 0$, donc l'incompatibilité est nulle. Par contre, si aucun attribut de l'explication candidate ne vérifie les contraintes de la prémisse, l'incompatibilité revient à la distance euclidienne sur les attributs A_E : $\| x - e \|_{A_E}$.

Fonction d'agrégation Comme dans les instanciations précédentes du cadre général, nous proposons d'agréger les deux termes de pénalité et d'incompatibilité à l'aide d'un opérateur de compromis, la somme pondérée.

Fonction globale Le problème d'optimisation pour rKICE peut être écrit comme suit :

$$\begin{aligned} e^* &= \underset{e \in \mathcal{E}_{x,f}}{\operatorname{argmin}}\, cost_{x,E}(e) \\ \text{avec } cost_{x,E}(e) &= \| x - e \|^2 + \lambda \sum_{i \in A_E} (x_i - e_i)^2 \times \mathbb{1}_{e_i \notin [v^i_{inf}, v^i_{sup}]} \end{aligned} \tag{5.8}$$

avec $\mathcal{E}_{x,f} = \{e \in \mathcal{X}, f(e) \neq f(x)\}$.

5.2.3 Description de l'algorithme

Cette section décrit l'algorithme rKICE proposé pour résoudre le problème d'optimisation de l'équation (5.8). Tout d'abord, nous décrivons le principe général de rKICE.

Algorithm 3 Génération des couches pour rKICE

Require: x, centre de la couche
Require: E, la connaissance utilisateur
Require: a_0 et a_1 les limites de la couche
Require: n, nombre de points désirés
Require: λ, poids
Ensure: $Z = \{z_i\}$
$S \leftarrow$ Décomposition en sous-espaces $S_j = C_j \bigcup \overline{C}_j$
for $j \leftarrow 1, ..., 2^{Card(A_E)}$ **do**
    $Z_j = \mathcal{EL}(x, E, \nu, \nu + \epsilon, \lambda, C_j)$ avec GCE
    for $z \in Z_j$ **do**
        if $z \in S_j$ **then**
            $Z = Z \cup \{z\}$
        end if
    end for
end for
return Z

Puis, nous détaillons l'étape de génération des instances. Enfin, nous illustrons les étapes de l'algorithme sur des données en deux dimensions pour deux exemples de connaissances différentes.

Principe L'algorithme rKICE utilise le même principe de génération itérative d'instances que KICE présenté dans le chapitre 4 et Growing Spheres (Laugel et al., 2018a), cf. section 2.5.5 : à chaque étape des instances sont générées dans des couches autour de l'instance étudiée jusqu'à ce qu'une instance d'une autre classe soit trouvée. Cette procédure est identique à celle mise en œuvre par l'algorithme 4.2, mais les couches générées à chaque étape sont différentes. En effet, quel que soit ν, l'équation $cost_{x,E}(e) = \nu$ ne définit ni une sphère, ni une ellipse, mais une forme complexe définie par l'union de couches ellipsoïdales définies avesc des paramètres différents dans des sous-espaces induits par les contraintes de la prémisse de la règle. Aussi dans une première étape, rKICE définit les sous-espaces, une seconde étape consiste à générer les instances dans les couches associées à chacun des sous-espaces.

Génération uniforme des couches Le principe présenté ici est implémenté dans l'algorithme 3. La prémisse de la règle déclenchée pour la donnée x définit un ensemble de conditions $C = \{x_i \in [v^i_{inf}, v^i_{sup}], i \in A_E\}$. La première étape de la méthode consiste à décomposer l'espace des données $\mathcal{X}$ en sous-espace S_j suivant que les instances du sous-espace vérifient ou non chacune des conditions présentes dans C. Chaque sous-espace est alors associé à deux ensembles C_j et $\overline{C}_j$ qui représentent respectivement les attributs de A_E dont les conditions sont satisfaites et ceux dont les conditions ne sont pas satisfaites. Nous obtenons au final $2^{Card(C)}$ sous-espaces.

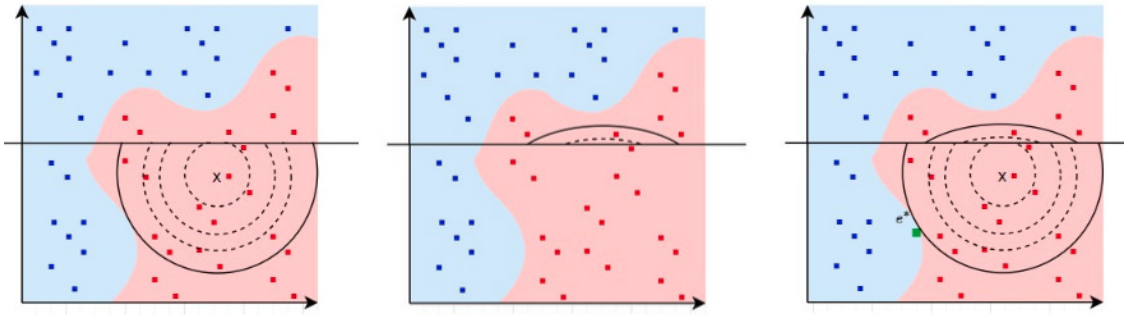


FIGURE 5.3 – Décomposition des couches générées par rKICE. (Gauche) couches associées aux instances qui vérifient les conditions, (Centre) couches associées aux instances qui ne vérifient pas les conditions et (Droite) couches générées à chaque étape.

Pour une instance e dans un sous-espace S_j, la fonction de coût s'écrit :

$$cost_{x,E}(e) = \| x_i - e_i \|^2 + \lambda \sum_{i \in \overline{C}_j} (x_i - e_i)^2$$

A chaque étape, nous souhaitons générer les instances telles que $cost_{x,E}(e) = \nu$ où ν est un paramètre incrémenté itérativement. Or, nous venons de voir que la fonction de coût dépend du sous-espace où se situe l'exemple contre-factuel candidat e, la fonction de coût s'écrit alors comme la combinaison de $2^{Card(A_E)}$ fonctions écrites sous la forme précédente. Chacune de ces fonctions est similaire à l'équation (4.4), lorsque la connaissance E est l'ensemble des attributs qui satisfont les conditions, c'est-à-dire : $E = C_j$. De plus, elle est associée à une couche générée avec la méthode GCE définie dans l'algorithme 2 : la couche ellipsoïdale associée au sous-espace S_j est $\mathcal{EL}(x, \nu, \nu + \epsilon, \lambda, C_j)$.

La couche finale est l'ensemble des instances de chaque couche qui appartiennent au sous-espace associé, comme on peut le voir sur les figures 5.3 et 5.4. Elle est définie comme :

$$Z = \bigcup_j \{z_i \in Z_j | z_i \in S_j\} \text{ avec } Z_j = \mathcal{EL}(x, \nu, \nu + \epsilon, \lambda, C_j)$$

L'algorithme proposé est de complexité exponentielle. Cependant, la connaissance utilisateur considérée est souvent de petite taille avec des règles contenant peu de conditions, $Card(A_E)$ est donc faible.

Exemple Cette section présente sur les figures 5.3 et 5.4, les étapes de l'algorithme pour des données en deux dimensions, notées X_1 et X_2, et deux exemples de connaissances. Sur la première figure 5.3, la règle utilisateur est : "si $X_2 < 0$ alors la classe est bleue". Dans ce cas, une seule condition est présente dans la règle, l'espace est donc divisé en deux zones. Les instances situées au-dessus de la ligne horizontale ne vérifient pas la condition présente dans la prémisse de la règle, au contraire de celles situées en dessous. Ainsi, la fonction de coût associée à la zone du bas est $\| x - e \|^2$, la couche associée est sphérique. Pour la zone du haut, la fonction associée est $\| x - e \|^2 + \lambda \| x - e \|^2_{X_2}$, la couche générée est ellipsoïdale. Dans notre exemple, on obtient l'explication la plus

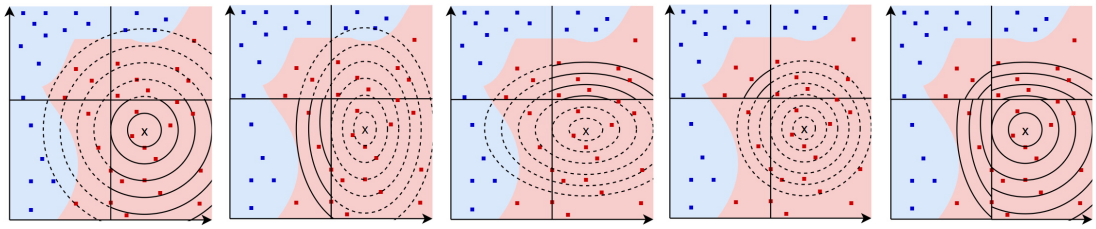


FIGURE 5.4 – Décomposition des couches générées par rKICE. (1) couches associées aux instances qui vérifient les deux conditions, (2 et 3) couches associées aux instances qui vérifient une seule des conditions, (4) couches associées aux instances qui ne vérifient pas les conditions et (5) couches générées à chaque étape.

proche de l'instance étudiée dans la zone souhaitée ; c'est-à-dire celle où l'instance étudiée x est présente.

Sur la figure 5.4, la connaissance utilisateur est : "si $X_1 > 0$ et $X_2 < 0$ alors la classe est bleue". La première étape consiste à décomposer l'espace en quatre sous-espaces, ils sont obtenus selon les conditions de la prémisse qui sont satisfaites : $X_1 > 0$ et $X_2 < 0$ (en bas à droite), $X_1 > 0$ (en haut à droite), $X_2 < 0$ (en bas à gauche) et aucune condition n'est satisfaite (en haut à gauche). Dans chacune de ces zones, la fonction de coût associée est différente, quatre couches différentes sont donc générées. La première figure est associée au cas où les deux conditions sont vérifiées : la fonction de coût se résume à la distance euclidienne, les couches sont sphériques. La seconde figure est associée à $X_2 < 0$, nous pénalisons donc les modifications selon X_1, la couche associée est une ellipse verticale. Sur le même principe, la troisième figure est associée à $X_1 > 0$, les modifications selon X_2 sont pénalisées, la couche associée est une ellipse horizontale. La quatrième figure considère qu'aucune condition est satisfaite, la fonction de coût est la distance euclidienne au carré pondérée par $1 + \lambda$, la couche associée est une sphère de rayon $\sqrt{\frac{\nu}{1+\lambda}}$. Dans toutes ces figures nous représentons en noir la partie de la couche qui nous intéresse, elle correspond aux instances qui sont dans le sous-espace étudié. Enfin, la dernière figure combine les quatre parties des différentes couches, ne formant ni une sphère ni une ellipse.

5.2.4 Étude expérimentale

Cette section décrit deux expérimentations menées avec la méthode rKICE. Comme dans le chapitre 4 et la section 5.1, nous considérons tout d'abord un exemple applicatif basé sur le jeu de données Californie. Nous évaluons ensuite la méthode proposée selon différentes métriques quantitatives.

5.2.4.1 Protocole expérimental

Dans cette section, les expérimentations reprennent le protocole expérimental de la section 4.3. Pour la connaissance utilisateur nous créons des arbres de décision de profondeur 4 pour les trois jeux de données Half-moons, Boston et Breast cancer, afin de garder des règles de prémisses courtes, c'est-à-dire de quatre conditions. Toutefois, avec

	$\boldsymbol{a_0}$	$\boldsymbol{a_1}$	$\boldsymbol{a_2}$	a_3	a_4	a_5	$\boldsymbol{a_6}$
x	-120.7	38.7	13.03	6.12	1.12	2094	4.08
e_{ref}	-0.7	-0.9	-0.94	-0.5	+0.03	0	+0.53
e_{user}	-0.2	-0.6	+4.77	-2.49	+0.19	+600	0
e^*	-0.8	-0.6	+2.77	-0.36	+0.05	0	+0.49

TABLE 5.4 – Exemples e_{ref}, e_{user} et e^* pour une instance x du jeu de données Californie. Les attributs qui font partie de A_E sont indiqués en gras. La valeur qu'ils prennent est notée en vert si elle satisfait la contrainte, en rouge sinon. Signification des attributs donnée dans le tableau 4.2.

	$P_x(e, f)$	$I_x(e, E)$	$cost_{x,E}(e)$
e_{ref}	**0.38**	0.35	0.73
e_{user}	1.40	**0.0**	1.40
e^*	0.40	0.23	**0.63**

TABLE 5.5 – Valeurs des métriques : $P_x(e)$, $I_x(e, E)$ et $cost_{x,E}(e)$ pour l'instance x du jeu de données Californie indiquée dans la première ligne du tableau 5.4 pour chacun des trois exemples contre-factuels.

les règles extraites de ces arbres, on remarque que l'explication la plus proche est dans la zone souhaitée, ce qui ne permet pas d'observer l'utilité de rKICE. Aussi, nous subdivisons arbitrairement chaque zone définie par chaque branche de l'arbre en plusieurs sous zones définies uniformément. Pour chaque instance, la connaissance utilisateur est enfin définie comme la règle extraite de la branche de l'arbre associée à x.

Pour rappel, une explication est souhaitée si les classes prédites par le modèle et la connaissance sont différentes. Ainsi, les trois explications : la plus proche (e_{ref}), la plus compatible (e_{user}) et celle obtenue avec rKICE (e^*) sont générées sur les instances x de l'ensemble test telles que le modèle et la règle aient une prédiction différente associée à une explication de référence en dehors de la zone souhaitée.

5.2.4.2 Exemple de résultats de rKICE sur la base Californie

Dans cette section, nous présentons l'explication obtenue pour une instance spécifique du jeu de données Californie indiquée dans la première ligne du tableau 5.4. Nous considérons la règle utilisateur suivante associée à l'instance x :

$$longitude > -121 \ \ \& \ \ latitude > 38 \ \ \& \ \ age > 13 \ \ \& \ \ revenu \leq 4.5 \Longrightarrow classe = pas\ chere$$

Le tableau 5.4 présente les trois explications associées à x. On remarque que les explications e_{ref}, e^* et e_{user} ne respectent pas respectivement 4, 2 et 0 conditions de la règle considérée. L'explication e_{user} est idéale étant donné qu'elle est dans la zone définie par la règle, par contre on remarque qu'elle propose des modifications importantes. L'explication fournie par l'algorithme proposé rKICE permet de vérifier deux contraintes selon les attributs a_1 et a_2. Ainsi, l'intégration des règles permet de se rapprocher de la zone de localisation et de l'âge médian des appartements souhaités par l'utilisateur.

		$P_x(e)$	$I_x(e,E)$	$cost_{x,E}(e)$
Half-moons	e_{ref}	**0.259 ± 0.11**	0.02 ± 0.1	0.278 ± 0.01
	e_{user}	0.260 ± 0.11	**0.0 ± 0.0**	**0.260 ± 0.0**
	e^*	0.260 ± 0.11	**0.0 ± 0.0**	**0.260 ± 0.0**
Boston	e_{ref}	**2.57 ± 1.50**	0.16 ± 0.36	2.73 ± 1.45
	e_{user}	2.67 ± 1.47	**0.0 ± 0.0**	2.67 ± 1.47
	e^*	2.60 ± 1.51	0.02 ± 0.07	**2.62 ± 1.50**
Breast cancer	e_{ref}	**23.88 ± 22.20**	3.28 ± 5.29	27.17 ± 26.79
	e_{user}	25.91 ± 22.76	**0.0 ± 0.0**	25.91 ± 22.76
	e^*	24.39 ± 22.13	0.22 ± 0.62	**24.61 ± 22.07**

TABLE 5.6 – Résultats obtenus avec les trois approches considérées sur les trois jeux de données pour les métriques : $P_x(e)$, $I_x(e,E)$ et $cost_{x,E,}(e)$ définies dans les équations (5.2), (5.3) et (5.5)

Le tableau 5.5 présente les valeurs des métriques associées à ces explications. Tout d'abord, nous remarquons que e_{user} a une pénalité élevée par rapport à celle de e_{ref} (1.40 vs 0.38). En effet, bien que e_{user} soit dans la zone souhaitée, l'augmentation de la pénalité est importante. Cependant, nous remarquons que les pénalités associées à e_{ref} et e^* sont proches, 0.38 pour le premier et 0.40 pour le second. Cela montre que donner une information sur la zone de l'espace souhaitée permet de guider la recherche vers une zone spécifique de l'espace et de définir une préférence sur la zone où se trouve l'explication.

5.2.4.3 Évaluation de la méthode rKICE

Cette section présente les résultats qualitatifs obtenus avec la méthode rKICE sur les différents jeux de données. Le nombre d'instances à expliquer est de 74, 85 et 68 instances respectivement pour Half-moons, Boston et Breast Cancer. Certaines de ces instances sont prédites différemment par le modèle et le système de règles, ce qui concerne seulement 32 instances pour Half-Moons, 47 pour Boston et 36 pour Breast Cancer.

Nous présentons les valeurs des métriques associés aux explications générées dans le tableau 5.6 et les temps d'exécution nécessaires pour obtenir les explications dans le tableau 5.7. Comme attendu, nous remarquons que pour les trois jeux de données la pénalité, l'incompatibilité et la fonction de coût sont minimisées respectivement par les explications e_{ref}, e_{user} et e^*. Pour ces trois jeux de données on remarque de plus que les différences au niveau de la pénalité entre ces trois exemples contre-factuels sont très faibles. Ceci montre qu'il est possible d'avoir des explications à la fois proches et compatibles. Les explications e^* fournies par rKICE sont proches de e_{user} selon les trois métriques. Dans ces cas les deux explications e_{user} et e^* sont intéressantes.

Le tableau 5.7 présente les temps d'exécution des trois méthodes. Il montre que le temps associé à rKICE est beaucoup plus élevé que celui des compétiteurs. Cela est dû à la complexité de l'algorithme qui dépend du nombre d'ellipses générées.

	e_{ref}	e_{user}	e^*
Half-moons	0.15 ± 0.05	0.19 ± 0.07	1.37 ± 0.49
Boston	1.64 ± 1.30	1.90 ± 1.45	9.09 ± 6.37
Breast Cancer	8.25 ± 7.38	11.33 ± 10.81	47.55 ± 45.88

TABLE 5.7 – Temps d'exécution des trois approches considérées pour obtenir e_{ref}, e_{user} et e^* pour les trois jeux de données.

5.3 Bilan

Nous avons enrichi le cadre d'intégration des connaissances présenté dans le chapitre 3 en considérant un nouveau type d'explications et une nouvelle forme de connaissances. La première instanciation, qui conduit à l'algorithme appelé KISM, considère des explications sous forme de vecteurs d'importance des attributs et des connaissances sous forme d'un ensemble d'attributs. Les conclusions obtenues sont proches de celles de KICE : KISM obtient comme attendu des explications qui accordent plus d'importance aux attributs utilisateurs, ce qui montre que les connaissances sont bien intégrées.

La seconde instanciation considère des exemples contre-factuels avec des connaissances sous forme de règles expertes. Nous proposons une nouvelle méthode rKICE qui se base sur la méthode KICE. rKICE considère un type de connaissances plus complexe que KICE et conduit à des temps de calcul plus longs.

Une des questions soulevées dans ce chapitre est l'agrégation choisie pour combiner la pénalité et l'incompatibilité, qui se pose également dans le chapitre 4 (cf. section 4.5.1). Le choix du paramètre de compromis λ est une étude importante. Il serait intéressant d'observer si la procédure pour choisir λ est la même pour les trois instanciations, KICE, KISM et rKICE ou si une procédure est plus adaptée qu'une autre pour une certaine instanciation.

Chapitre 6

Intégration des besoins utilisateur avec les intégrales de Gödel

Les méthodes proposées dans les chapitre 4 et 5 considèrent pour la fonction d'agrégation une moyenne pondérée, qui offre un comportement classique de compromis entre les critères. Dans ce chapitre, nous proposons une étude plus détaillée de cette fonction qui combine la pénalité et l'incompatibilité. Notre étude est valable pour tout type d'explications et tout type de connaissances, mais nous illustrons ces travaux pour les explications contre-factuelles et les connaissances sous forme d'ensemble d'attributs.

L'une des particularités de l'opération d'agrégation vient de la différence de nature des deux critères qui ne partagent pas la même sémantique : le premier, la pénalité, a pour but de mesurer une notion objective qui est la qualité de l'explication par rapport au modèle. Elle est indépendante de l'utilisateur. Le second, l'incompatibilité, mesure une notion subjective liée à l'utilisateur. Nous étudions en détail dans ce chapitre la combinaison de ces critères sémantiquement différents.

Comme présenté dans la section 3.5.1, il existe de nombreux opérateurs d'agrégation. Très souvent, ces opérateurs suivent un comportement conjonctif, disjonctif ou de compromis. Cependant, ici nous ne désirons pas le même comportement pour toutes les valeurs des critères. Le choix de l'opérateur d'agrégation est alors complexe, deux solutions sont possibles. Une première consiste à considérer un comportement hybride qui n'a pas un unique comportement mais plusieurs comportements selon la valeur des critères. Dans ce cas, le choix des comportements peut être personnalisé selon les besoins utilisateur. Une seconde solution est de générer plusieurs explications, en utilisant plusieurs opérateurs d'agrégation, cette solution est discutée dans le chapitre 7. Dans ce chapitre, nous nous concentrons sur la première solution qui permet d'ajouter un deuxième niveau de personnalisation, en plus de la prise en compte de la fonction d'incompatibilité.

Ce chapitre est structuré comme suit : la section 6.1 présente les propriétés souhaitées pour la fonction d'agrégation. La section 6.2 présente et justifie l'opérateur choisi : les intégrales de Gödel. Dans la section 6.3 nous proposons une nouvelle méthode nommée *Gödel Integrals for Counterfactual Explanation* (GICE) pour générer des explications dans ce cadre. Les sections 6.4 et 6.5 présentent les exemples illustratifs et les expérimentations qui analysent les résultats obtenus.

Une première version de ce travail a été présentée dans l'article *Knowledge Integration in XAI with Gödel Integrals* publié à la conférence Fuzz-IEEE 2023 (Jeyasothy et al., 2023b) et à la conférence LFA 2023 (Jeyasothy et al., 2023c).

6.1 Caractéristiques désirées pour l'agrégation de la pénalité et l'incompatibilité

Nous nous intéressons à la combinaison de deux critères : la pénalité et l'incompatibilité. Pour rappel, le problème d'optimisation considéré est défini dans l'équation (3.1) comme :

$$e^* = \underset{e \in \mathcal{E}}{\operatorname{argmin}}\, agg(P_x(e, f), I_x(e, E))$$

Dans ce chapitre, nous considérons $P_x(e)$ car nous nous étudions principalement les explications contre-factuelles. Cette section examine les propriétés de la fonction agg. Nous étudions successivement quatre propriétés : la monotonie, la commutativité, le comportement et la priorisation qui ont un impact important sur la sélection de l'opérateur.

6.1.1 Discussion sur la monotonie

La première propriété que nous examinons pour l'opérateur d'agrégation considéré concerne le comportement de monotonie qu'il doit satisfaire : nous défendons qu'il doit être croissant en ses deux arguments, pénalité et incompatibilité. Formellement, la croissance en fonction de son premier argument s'écrit :

$$\forall x_1, x_2, x_1' \in [0,1],\ x_1 \leq x_1' \Longrightarrow agg(x_1, x_2) \leq agg(x_1', x_2)$$

De la même manière, la fonction d'agrégation souhaitée doit être croissante en fonction de son second argument. En effet, le coût global doit évidemment augmenter dès que l'un ou l'autre des critères augmente. On peut noter que cette monotonie non stricte signifie qu'il peut y voir des cas d'égalité. Nous ne considérons pas une monotonie stricte car à notre connaissance il n'existe pas de fonction d'agrégation qui vérifie les 3 propriétés suivantes et la monotonie stricte.

6.1.2 Discussion sur la commutativité

L'opérateur d'agrégation considéré peut être commutatif, mais cette propriété n'est pas nécessairement souhaitée. Comme nous l'avons évoqué dans la section 3.2, les deux critères considérés, $P_x(e)$ et $I_x(e, E)$, ont des sémantiques différentes, étant respectivement de nature objective et subjective. Ils ne sont donc pas équivalents, la propriété de commutativité n'est pas attendue : il se peut que $agg(x, y) \neq agg(y, x)$ parce que $y = P_x(e)$ n'a pas la même signification que $y = I_x(e, E)$.

A titre illustratif, considérons à nouveau l'exemple fictif introduit dans la section 3.4.1, d'un classifieur qui reconnaît le type de légumes et une instance prédite comme une carotte. Nous considérons ici une connaissance utilisateur $E = \{poids\}$ et les deux explications suivantes pour que le légume soit prédit comme un panais : l'exemple contre-factuel e_1 explique la prédiction en indiquant que le taux de provitamine et le taux de saccharose doivent diminuer de 1% ; l'exemple contre-factuel e_2 quant à elle indique que le poids doit diminuer de 200g et que le taux de saccharose diminue de 5%. La seconde explication modifie l'attribut utilisateur, elle est donc plus compatible que la première. On considère les valeurs fictives suivantes pour chacune des explications : e_1 a une pénalité de 0.1 et une incompatibilité de 0.8 et e_2 a une pénalité de 0.8 et une incompatibilité de 0.1. Si on considère un opérateur commutatif, ces deux explications sont associées au même coût, or l'exemple présenté montre que ces deux explications ne sont pas similaires : on souhaite leur associer un coût différent. C'est pourquoi la propriété sur la commutativité n'est pas souhaitée.

6.1.3 Discussion sur le comportement des critères

En IA explicable, l'une des difficultés du choix d'une fonction d'agrégation est qu'elle doit être adaptée à tous les types d'utilisateurs, qui ont des motivations et des besoins différents. Nous défendons donc pour résoudre ces problématiques que les opérateurs d'agrégation considérés devraient avoir des comportements différents selon les besoins utilisateur. Nous proposons de définir ces besoins selon les valeurs des critères, ainsi les comportements dépendent des valeurs des critères, par exemple être conjonctifs pour certaines valeurs, disjonctifs pour d'autres et offrir une propriété de compromis pour d'autres encore.

Dans une première étape, il faut définir les intervalles de valeur associées à chacun de ces comportements. Nous proposons de définir quatre zones, à partir de valeurs limites acceptées pour chacun des critères, notés δ_P et δ_I : on a alors les zones où

$$P_x(e) \leq \delta_P \text{ et } I_x(e, E) \leq \delta_I$$
$$P_x(e) \leq \delta_P \text{ et } I_x(e, E) > \delta_I$$
$$P_x(e) > \delta_P \text{ et } I_x(e, E) \leq \delta_I$$
$$P_x(e) > \delta_P \text{ et } I_x(e, E) > \delta_I$$

Les valeurs δ_P et δ_I doivent être choisies par les utilisateurs pour exprimer leurs contraintes personnelles et imposent des valeurs maximales pour la pénalité et l'incompatibilité.

Le choix de ces valeurs peut être difficile. Dans le cas de la pénalité nous proposons d'exprimer le seuil δ_P non pas de façon absolue, mais de façon relative par rapport à un cas de référence. Cette référence peut être définie comme la valeur de pénalité associée à l'exemple contre-factuel e_{ref}, qui est généré en optimisant uniquement le critère de pénalité. La contrainte s'exprime alors en terme de perte de pénalité acceptée par rapport

à e_{ref}, sous la forme :

$$P_x(e) - P_x(e_{ref}) < \delta'_P \tag{6.1}$$

qui est de la forme précédente en considérant $\delta_P = \delta'_P + P_x(e_{ref})$. On peut noter que ce seuil prend des valeurs qui dépendent de x, par le biais de la valeur $P_x(e_{ref})$.

Pour l'incompatibilité, la définition de la valeur de référence pose problème : comme discuté dans la section 3.1.2 l'exemple contre-factuel e_{user} obtenu pour une incompatibilité nulle n'existe pas toujours. Aussi, nous proposons de conserver une définition absolue de la contrainte :

$$I_x(e, E) < \delta_I \tag{6.2}$$

Les contraintes (6.1) et (6.2) divisent l'espace des critères, décrit par les couples $(P_x(e), I_x(e, E))$, en quatre zones différentes comme définies ci-dessus, la décomposition est représentée sur la figure 6.1, selon que les deux, une seule ou aucune contrainte est satisfaite. La zone en bas à gauche (en vert) correspond au meilleur cas, où les deux contraintes sont vérifiées, et la zone en haut à droite est le pire cas, où aucune des deux contraintes n'est satisfaite. Enfin, les deux zones restantes correspondent au cas où uniquement l'un des critères est satisfaisant.

Dans la zone en bas à gauche, les deux critères ont des valeurs faibles, on peut souhaiter une valeur faible qui est la pénalité *ou* l'incompatibilité : un comportement disjonctif est alors plus intéressant. Par contre, dans la zone en haut à droite, les deux critères ont des valeurs élevées, on souhaite alors que la pénalité *et* l'incompatibilité soient aussi faibles que possible : un comportement conjonctif est alors plus intéressant. Enfin dans les deux dernières zones, une des valeurs est faible et l'autre est élevée, pour compenser les deux valeurs un comportement de compromis est souhaitable. Ainsi, il semble souhaitable que la fonction d'agrégation offre des comportements différents dans les zones définies par les contraintes sur les besoins utilisateur qu'on propose de visualiser sur la figure 6.1. L'interprétation dans chacune des zones n'est alors pas la même.

6.1.4 Discussion sur la priorité

Nous avons vu, dans le chapitre 3, une personnalisation à travers une prise en compte de connaissances utilisateur, ici on peut en plus avoir une personnalisation sur les préférences selon chaque critère. L'utilisateur peut avoir une préférence sur les critères, ce qui induit une hiérarchie entre eux, qui peut être interprétée comme un comportement prioritaire souhaité. Cette préférence n'est évidemment pas la même pour tous les utilisateurs.

Une possibilité consiste à intégrer la notion de priorité par le biais du choix des seuils dans les équations (6.1) et (6.2). Dans le cas où la pénalité est préférée à l'incompatibilité,

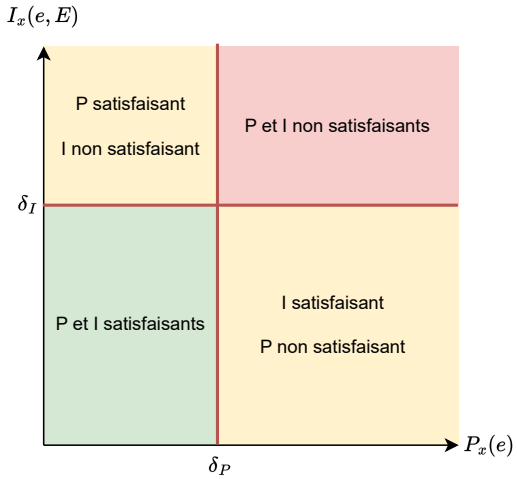


FIGURE 6.1 – Décomposition de l'espace des critères à partir des besoins utilisateur exprimés par les équations (6.1) et (6.2).

	Monotonie	Non commutativité	Comportement	Priorité
$P_x(e)$	✓	✓	×	×
$I_x(e,E)$	✓	✓	×	×
$\min(P_x(e), I_x(e,E))$	✓	×	×	×
$\max(P_x(e), I_x(e,E))$	✓	×	×	×
$\lambda P_x(e) + (1-\lambda) I_x(e,E))$	✓	✓	×	✓

TABLE 6.1 – Propriétés vérifiées par cinq opérateurs classiques.

une condition plus forte sur la pénalité que sur l'incompatibilité est attendue, c'est-à-dire que l'utilisateur choisit un seuil δ_P plus faible associé à la pénalité qu'un seuil δ_I associé à l'incompatibilité.

6.1.5 Conséquences sur le choix de l'opérateur

Dans cette section, nous étudions cinq opérateurs de référence qui vérifient les comportements classiques : conjonction, disjonction et de compromis, présentés précédemment dans la section 3.5.1. Le tableau 6.1 liste parmi les propriétés présentées précédemment, celles qui sont vérifiées par les opérateurs suivants : $P_x(e)$, $I_x(e,E)$, $\min(P_x(e), I_x(e,E))$, $\max(P_x(e), I_x(e,E))$ et $\lambda P_x(e) + (1-\lambda) I_x(e,E)$ qui est équivalent à $P_x(e)$ si $\lambda = 1$ et à $I_x(e,E)$ si $\lambda = 0$.

Tous ces opérateurs vérifient la propriété de monotonie. Par contre, ils proposent tous un unique type de comportement dans l'espace des critères, alors que nous souhaitons des comportements variés, par exemple un comportement disjonctif pour des valeurs faibles et conjonctif pour des valeurs élevées. Enfin, seule la moyenne pondérée intègre la possibilité de prioriser des critères, en permettant d'associer un poids différent à chaque critère.

6.2 Intégrales de Gödel

Les intégrales de Gödel, introduites par Dubois et al., 2017 constituent une famille d'opérateurs d'agrégation avec un comportement plus complexe que les trois comportements présentés dans la section 3.5.1 : comme rappelé ci-dessous, elles vérifient bien les quatre propriétés souhaitées. Cette section rappelle d'abord la définition générale des intégrales de Gödel. Puis, nous discutons de l'utilisation de ces intégrales dans le cadre de l'IA explicable considéré, c'est-à-dire que nous présentons l'instanciation du formalisme général (3.1) qui combine la pénalité et l'incompatibilité dans ce cadre.

6.2.1 Définition des intégrales de Gödel

Nous rappelons dans cette section la définition générale des intégrales de Gödel, qui constituent des variantes d'une intégrale plus générale appelée intégrale de Sugeno (Sugeno, 1974). Après avoir présenté les notations utilisées dans cette section, nous rappelons la définition des intégrales de Sugeno. Puis, les sections 6.2.1.2 et 6.2.1.3 définissent les deux variantes de l'intégrale de Gödel, respectivement basées sur la conjonction et l'implication de Gödel.

Notations L'ensemble des critères d'évaluation est noté $\mathcal{C} = \{1, \cdots, n\}$, ils sont considérés comme étant évalués numériquement, par des valeurs dans $L = [0, 1]$. Nous considérons ici une instance $x = (x_1, ..., x_n)$, où x_i est la valeur associée au critère i. L'agrégation fournit une valeur à partir de ces n valeurs.

Comme les intégrales de Choquet et de Sugeno (Grabisch and Labreuche, 2010), les intégrales de Gödel permettent de modéliser et de prendre en compte le fait que les critères, mais aussi les sous-ensembles de critères, ont des poids différents : elles permettent de représenter l'importance des critères individuels ainsi que leurs interactions.

Formellement, cette importance est modélisée par une fonction, $\mu : 2^{\mathcal{C}} \to [0, 1]$ appelée capacité ou mesure floue qui associe chaque sous-ensemble de critères $A \subseteq \mathcal{C}$ à un poids $\mu(A)$. Par définition, cette fonction est croissante par rapport à l'inclusion d'ensemble et satisfait les conditions limites $\mu(\emptyset) = 0$ et $\mu(\mathcal{C}) = 1$.

6.2.1.1 Intégrales de Sugeno

L'intégrale de Sugeno (Sugeno, 1974) est une intégrale qualitative utilisée dans la prise de décision multicritère pour agréger les scores d'instances évalués selon plusieurs critères. Elle a deux expressions équivalentes, une forme max-min et une forme min-max (Marichal, 2000) respectivement définies comme :

$$\int_{\mu} x = \max_{A \subseteq \mathcal{C}} \left(\mu(A) \wedge \min_{i \in A} x_i \right) \tag{6.3}$$

et :

$$\int_{\mu} x = \min_{A \subseteq \mathcal{C}} \left(\mu^c(A) \vee \max_{i \in A} x_i \right) \tag{6.4}$$

où $\wedge$ désigne un opérateur conjonctif et $\vee$ un opérateur disjonctif et μ^C la capacité conjuguée de μ, définie par $\mu^C(A) = 1 - \mu(\overline{A})$ où $\overline{A}$ est l'ensemble complémentaire de A. Il y a alors une première agrégation qui consiste à combiner la capacité et le minimum d'un ensemble de valeurs.

En généralisant les équations précédentes avec la conjonction ou l'implication de Gödel, on obtient deux opérateurs différents, qui constituent la famille des intégrales de Gödel. Les deux sections suivantes rappellent leur définition formelle.

6.2.1.2 Intégrale de Gödel basée sur la conjonction

L'intégrale de Gödel basée sur la conjonction est définie par la forme max-min de l'intégrale de Sugeno en considérant comme opérateur de conjonction $\wedge$ la conjonction de Gödel. L'équation (6.3) s'écrit :

$$G_\mu^{\otimes}(x) = \max_{A \subseteq \mathcal{C}} \left(\mu(A) \otimes_G \min_{i \in A} x_i \right) \tag{6.5}$$

où la conjonction de Gödel $\otimes_G$ (Dubois and Prade, 1984) est un opérateur défini pour tout $\alpha, \beta \in [0,1]$ par :

$$\alpha \otimes_G \beta = \begin{cases} 0 & \text{si } \beta \leq 1 - \alpha \\ \beta & \text{sinon.} \end{cases}$$

Bien que cet opérateur soit appelé une conjonction, il ne vérifie pas la propriété classique de commutativité. Il est croissant dans ses deux arguments et satisfait les conditions limites suivantes :

— $1 \otimes_G \beta = \beta$,
— $\alpha \otimes_G 1 = 0$ si $\alpha = 0$ et 1 sinon,
— $0 \otimes_G \beta = \alpha \otimes_G 0 = 0$.

Lorsqu'on considère A un singleton, c'est-à-dire $A = \{i\}$, de poids $\mu(A) = \mu_i$, la conjonction $\mu(A) \otimes_G \min_{j \in A} x_j$ devient $\mu_i \otimes_G x_i$ étant donné que $\min_{j \in A} x_j = x_i$. La conjonction de Gödel s'écrit alors comme suit :

$$\mu_i \otimes_G x_i = \begin{cases} 0 & \text{si } x_i \leq 1 - \mu_i \\ x_i & \text{sinon.} \end{cases}$$

Cela signifie que la valeur x_i n'est pas modifiée si elle est supérieure au seuil $1 - \mu_i$ et qu'elle est fixée à 0 dans le cas contraire : les valeurs non satisfaisantes (supérieures à 1-μ_i) ont un coût qui correspond à leurs valeurs, les autres (inférieures à 1-μ_i) ont un coût nul. Le seuil de satisfaction est décroissant par rapport à μ_i : il est petit lorsque μ_i est élevé, c'est-à-dire lorsque le critère i est important. Ainsi, la définition de la notion de satisfaction dépend de l'importance du critère. Une évaluation non satisfaisante sur x_i sur un critère important est modifiée en 0, tandis qu'une petite évaluation sur un critère non important est conservée.

Le principe illustré ici sur les singletons est étendu à tout ensemble de critères A, dont l'évaluation est définie par le minimum de leurs évaluations individuelles. Son

poids est donné par $\mu(A)$, qui permet de représenter l'importance de coalition des critères.

6.2.1.3 Intégrale de Gödel basée sur l'implication

L'intégrale de Gödel basée sur l'implication, proposée par Dubois et al., 2017, est définie par la forme min-max de l'intégrale de Sugeno en considérant comme opérateur de disjonction $\vee$ l'implication de Gödel. L'équation (6.4) s'écrit :

$$G_{\mu}^{\rightarrow}(x) = \min_{A \subseteq \mathcal{C}} \left(\mu^{c}(A) \rightarrow_{G} \max_{i \in A} x_i \right) \tag{6.6}$$

où l'implication de Gödel $\rightarrow_G$ (Baehrens et al., 2010) est l'opérateur défini pour tout α, $\beta \in [0,1]$ par :

$$\alpha \rightarrow_G \beta = \begin{cases} 1 & \text{if } \alpha \leq \beta \\ \beta & \text{sinon} \end{cases}$$

Cette implication satisfait aux conditions limites suivantes :

— $0 \rightarrow_G \beta = 1$,

— $\alpha \rightarrow_G 1 = 1$.

Comme dans le cas de la conjonction, dans le cas où A est un singleton $A = \{i\}$ de poids $\mu(A) = \mu_i$, la valeur x_i du critère i est transformée à l'aide de l'implication de Gödel :

$$\mu_i^c \rightarrow_G x_i = \begin{cases} 1 & \text{si } \mu_i^c \leq x_i \\ x_i & \text{sinon.} \end{cases}$$

Cela signifie que la valeur x_i n'est pas modifiée si elle est inférieure au seuil μ_i^c et qu'elle est fixée à 1 dans le cas contraire : les valeurs non satisfaisantes (supérieures à μ_i^c) ont un coût élevé alors que les autres valeurs satisfaisantes (inférieures à μ_i^c) ont un coût plus faible qui correspond à leurs valeurs. Contrairement, au cas conjonctif où la notion de satisfaction dépend de l'importance du critère étudié, ici elle dépend de l'importance des autres critères. Une évaluation sur un critère non important par rapport aux autres critères devient 1, ce qui offre évidemment une sémantique différente du cas de la conjonction de Gödel.

Le principe illustré ici sur les singletons est étendu à tout ensemble de critères A, dont l'évaluation est définie par le maximum de leurs évaluations individuelles. Son poids est donné par $\mu(A)$, qui permet de représenter l'importance de coalition des critères. Des exemples de ces opérateurs d'agrégation sont fournis dans la sous-section suivante, lorsqu'ils sont appliqués dans le cadre de l'IA explicable.

6.2.2 Intégrales de Gödel appliquées à la pénalité et à l'incompatibilité

Nous venons de rappeler les définitions des intégrales de Gödel dans le cadre général. Dans cette section, nous nous concentrons sur le domaine de l'IA explicable pour tout type d'explications et tout type de connaissances. Tout d'abord, nous instancions les intégrales de Gödel pour les deux critères considérés, pénalité et incompatibilité. Puis,

nous étudions la prise en compte des besoins utilisateur par le biais de ces opérateurs et nous montrons qu'ils satisfont les quatre propriétés discutées dans la section 6.1. Nous analysons les différents comportements de cette intégrale dans l'espace des critères.

6.2.2.1 Instanciation des intégrales de Gödel

Cette section traite de l'application des définitions générales rappelées ci-dessus à l'agrégation des valeurs de pénalité et d'incompatibilité. L'expression formelle des valeurs agrégées, respectivement $G_\mu^{\otimes}(P_x(e), I_x(e,E))$ et $G_\mu^{\rightarrow}(P_x(e), I_x(e,E))$, est donnée ci-dessous, leurs lignes de niveau sont illustrées sur la figure 6.2 et leur interprétation est détaillée dans la section suivante.

On définit ici l'ensemble des critères comme $\mathcal{C} = \{P_x(e), I_x(e,E)\}$, qui sont normalisés et évalués sur l'échelle $L = [0,1]$. La capacité considérée est alors définie sur l'univers $2^{\mathcal{C}}$ dont la taille est égale à $4 : 2^{\mathcal{C}} = \{\emptyset, \{P_x(e)\}, \{I_x(e,E)\}, \{P_x(e), I_x(e,E)\}\}$. Deux valeurs sont fixées en raison des conditions aux limites qui sont $\mu(\emptyset) = 0$ et $\mu(\{P_x(e), I_x(e,E)\}) = 1$, nous notons les deux autres valeurs : $\mu(\{P_x(e)\}) = \mu_P$ et $\mu(\{I_x(e,E)\}) = \mu_I$. Le lien entre les capacités et les seuils δ_P et δ_I introduits dans la section 6.1.3 est discuté dans la section suivante. L'agrégation des deux valeurs $P_x(e)$ et $I_x(e,E)$ par l'intégrale de Gödel basée sur la conjonction est alors égale à :

$$G_\mu^{\otimes}(P_x(e), I_x(e,E)) = \max(\mu_P \otimes_G P_x(e), \mu_I \otimes_G I_x(e,E), 1 \otimes_G \min(P_x(e), I_x(e,E))) \quad (6.7)$$

$$= \begin{cases} \min(P_x(e), I_x(e,E)) & \text{si } P_x(e) \leq 1-\mu_P \text{ et } I_x(e,E) \leq 1-\mu_I \\ \max(P_x(e), I_x(e,E)) & \text{si } P_x(e) > 1-\mu_P \text{ et } I_x(e,E) > 1-\mu_I \\ P_x(e) & \text{si } P_x(e) > 1-\mu_P \text{ et } I_x(e,E) \leq 1-\mu_I \\ I_x(e,E) & \text{si } P_x(e) \leq 1-\mu_P \text{ et } I_x(e,E) > 1-\mu_I \end{cases} \quad (6.8)$$

Comme le montre l'équation ci-dessus et comme l'illustrent les graphiques supérieurs de la figure 6.2, lorsque l'intégrale de Gödel basée sur la conjonction est appliquée à deux critères, elle divise l'espace des critères en quatre régions, en fonction de la position relative de chaque critère $P_x(e)$ et $I_x(e,E)$ dans cet espace et de leurs seuils associés $1-\mu_P$ et $1-\mu_I$.

Dans le cas de l'intégrale de Gödel basée sur l'implication, la définition formelle est :

$$G_\mu^{\rightarrow}(P_x(e), I_x(e,E)) = \min\left((1-\mu_I) \rightarrow_G P_x(e), (1-\mu_P) \rightarrow_G I_x(e,E), 1 \rightarrow_G \max(P_x(e), I_x(e,E))\right) \quad (6.9)$$

$$= \begin{cases} \min(P_x(e), I_x(e,E)) & \text{si } P_x(e) < 1-\mu_I \text{ et } I_x(e,E) < 1-\mu_P \\ \max(P_x(e), I_x(e,E)) & \text{si } P_x(e) \geq 1-\mu_I \text{ et } I_x(e,E) \geq 1-\mu_P \\ I_x(e,E) & \text{si } P_x(e) \geq 1-\mu_I \text{ et } I_x(e,E) < 1-\mu_P \\ P_x(e) & \text{si } P_x(e) < 1-\mu_I \text{ et } I_x(e,E) \geq 1-\mu_P \end{cases} \quad (6.10)$$

Comme dans le cas précédent cette intégrale divise l'espace des critères en quatre sous-espaces. Par contre, les fonctions d'agrégation associées à chaque sous-espace ne sont pas les mêmes.

Propriétés Il est simple de montrer que $G_\mu^\otimes(P_x(e), I_x(e,E))$ et $G_\mu^\rightarrow(P_x(e), I_x(e,E))$ satisfont toutes les propriétés présentées dans la section 6.1 : elles sont monotones en chaque argument, non commutatives, offrent un comportement variable et permettent d'exprimer une hiérarchie de critères. Le comportement variable est visible dans les expressions formelles ainsi que dans la représentation graphique de la figure 6.2 qui correspond bien à la décomposition souhaitée par la figure 6.1.

Cas extrêmes Les fonctions d'agrégation classiques étudiées dans la section 6.1.5 correspondent à des cas extrêmes des intégrales de Gödel, à part la moyenne pondérée. Nous donnons ci-dessous leurs expressions et les valeurs des paramètres μ_P et μ_I qui en font des instanciations de $G_\mu^\otimes$:

$$G_\mu^\otimes(P_x(e), I_x(e,E)) = P_x(e) \quad \text{si} \quad \mu_P = 1 \quad \text{et} \quad \mu_I = 0 \tag{6.11}$$
$$G_\mu^\otimes(P_x(e), I_x(e,E)) = I_x(e,E) \quad \text{si} \quad \mu_P = 0 \quad \text{et} \quad \mu_I = 1 \tag{6.12}$$
$$G_\mu^\otimes(P_x(e), I_x(e,E)) = \min(P_x(e), I_x(e,E)) \quad \text{si} \quad \mu_P = 0 \quad \text{et} \quad \mu_I = 0 \tag{6.13}$$
$$G_\mu^\otimes(P_x(e), I_x(e,E)) = \max(P_x(e), I_x(e,E)) \quad \text{si} \quad \mu_P = 1 \quad \text{et} \quad \mu_I = 1 \tag{6.14}$$

L'équation (6.11) correspond au cas classique où seule la pénalité est prise en compte, l'équation (6.12) considère uniquement l'incompatibilité en ignorant la pénalité ; les équations (6.13) et (6.14) représentent respectivement la conjonction et la disjonction des deux critères. Ces mêmes instanciations sont valables pour $G_\mu^\rightarrow$.

6.2.2.2 Comportement de l'intégrale de Gödel selon les besoins utilisateur

Dans la section 6.1.3, nous avons introduit la prise en compte des besoins utilisateur en décomposant l'espace des critères. Dans cette section, nous étudions comment les intégrales de Gödel vérifient cette propriété. Pour illustrer les comportements, nous nous appuyons sur des explications sous forme d'exemples contre-factuels associées à la pénalité et à l'incompatibilité définies respectivement par les équations (4.1) et (4.2). Pour rappel, la pénalité quantifie les modifications sur l'ensemble des attributs, l'incompatibilité quant à elle quantifie les modifications sur les attributs inconnus. Tout d'abord, nous étudions la décomposition de l'espace des critères par les intégrales de Gödel. Ensuite, nous décrivons les comportements associés dans les différentes zones. Enfin, nous étudions les tailles relatives associées à chacune des zones.

Interprétation des seuils La correspondance entre les paramètres de Gödel, μ_P et μ_I, et les seuils δ_P et δ_I associés aux contraintes discutées dans la section 6.1.3 peut être établie en comparant les régions qu'ils définissent respectivement. Par exemple, pour $G_\mu^\otimes(P_x(e), I_x(e,E))$ et le critère $P_x(e)$, la contrainte est satisfaite lorsque $P_x(e) \leq 1 - \mu_P$, alors que pour $G_\mu^\rightarrow(P_x(e), I_x(e,E))$, la contrainte est $P_x(e) \leq 1 - \mu_I$. En les comparant aux besoins utilisateur exprimées dans l'équation (6.1), on obtient les valeurs suivantes

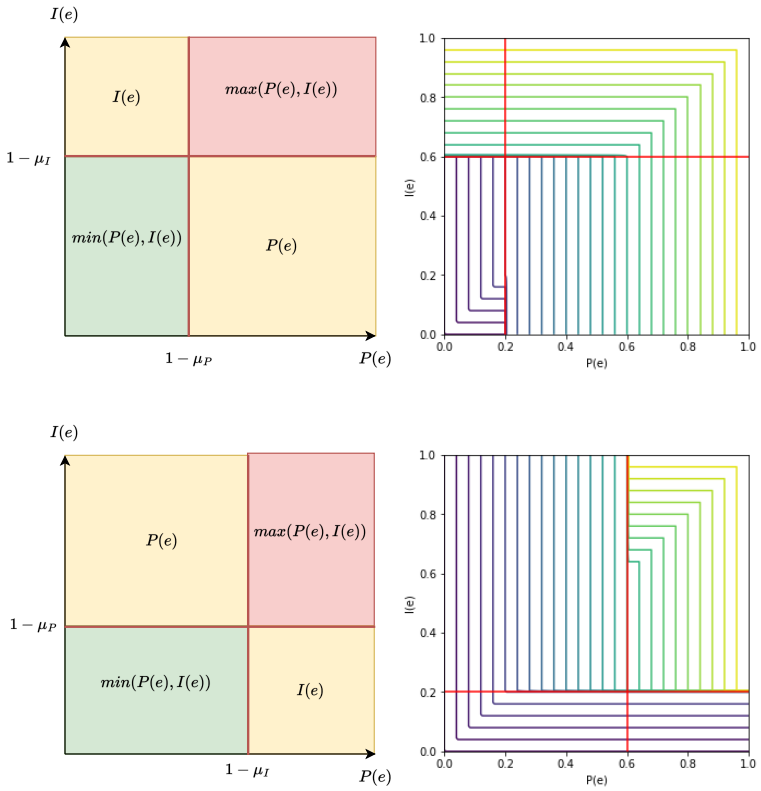


FIGURE 6.2 – Lignes de niveau pour les intégrales de Gödel avec $\mu_P = 0.8$ et $\mu_I = 0.4$: (en haut) $G^{\otimes}_{\mu}(P_x(e), I_x(e,E))$, (en bas) $G^{\rightarrow}_{\mu}(P_x(e), I_x(e,E))$.

des seuils de tolérance pour $G^{\otimes}_{\mu}(P_x(e), I_x(e,E))$:

$$\begin{cases} \delta_P = 1 - \mu_P \\ \delta_I = 1 - \mu_I \end{cases}$$

et pour $G^{\rightarrow}_{\mu}(P_x(e), I_x(e,E))$, on obtient :

$$\begin{cases} \delta_P = 1 - \mu_I \\ \delta_I = 1 - \mu_P \end{cases}$$

La capacité μ peut donc être définie à partir des besoins utilisateur définis par les seuils δ_P et δ_I. Les différences sont commentées ci-dessous, en examinant la différence entre les régions induites.

Interprétation des comportements dans chaque région Pour interpréter des régions, nous nous concentrons sur la représentation graphique donnée dans la figure 6.2 pour $\mu_P = 0.8$ et $\mu_I = 0.4$. On peut observer que $G^{\otimes}_{\mu}(P_x(e), I_x(e,E))$ et $G^{\rightarrow}_{\mu}(P_x(e), I_x(e,E))$ partagent deux régions similaires, en bas à gauche et en haut à droite. La région inférieure gauche correspond aux explications qui satisfont les deux contraintes, les modifications, globales et selon les attributs inconnus, sont faibles, et peuvent donc être considérées

comme satisfaisantes. Leur évaluation ne dépend alors que du meilleur critère, c'est-à-dire le minimum entre $P_x(e)$ et $I_x(e, E)$ (rappelons que le coût global doit être minimisé). Dans ce cas, le comportement considéré est disjonctif. Au contraire, dans la région en haut à droite, les candidats ne satisfont aucune des contraintes, les modifications, globales et selon les attributs inconnus, sont élevées. Afin de pénaliser des explications candidates, leur score est défini comme le maximum entre $P_x(e)$ et $I_x(e, E)$. Ce cas de figure correspond à un comportement conjonctif.

Pour les deux zones restantes, les deux intégrales n'offrent pas la même agrégation, car elles sont basées sur des principes différents. Considérons tout d'abord le cas où la contrainte de pénalité est satisfaite, mais pas celle d'incompatibilité, ce qui correspond à la zone supérieure gauche : les modifications globales sont alors faibles mais elles sont concentrées sur les attributs inconnus. $G^{\otimes}_{\mu}(P_x(e), I_x(e, E))$ adopte un comportement de punition, en pénalisant les candidats de cette région à hauteur de leur critère insatisfait, $I_x(e, E)$, indépendamment de leur valeur de pénalité. L'explication qui est favorisée est celle qui effectue le moins de modifications selon les attributs inconnus. Au contraire, $G^{\rightarrow}_{\mu}(P_x(e), I_x(e, E))$ considère que les candidats sont tous aussi mauvais en ce qui concerne l'incompatibilité et ne les distingue pas par rapport à ce critère, les considérant comme des explications non souhaitées selon ce critère. $G^{\rightarrow}_{\mu}(P_x(e), I_x(e, E))$ évalue alors ces candidats selon leur valeur de pénalité. Dans ce cas, l'explication proposée est celle qui effectue le moins de modifications globalement sur tous les attributs.

Ceci constitue une différence sémantique majeure qui souligne la richesse et la pertinence des intégrales de Gödel en tant qu'opérateur d'agrégation dans le domaine de l'IA explicable. De la même manière, dans le cas où la contrainte d'incompatibilité est satisfaite, mais pas celle de la pénalité, ce qui correspond à la région inférieure droite, $G^{\otimes}_{\mu}(P_x(e), I_x(e, E))$ considère le critère insatisfait $P_x(e)$ et $G^{\rightarrow}_{\mu}(P_x(e), I_x(e, E))$ considère le critère satisfait $I_x(e, E)$.

Interprétation des tailles relatives des régions Nous commentons enfin l'influence des paramètres de Gödel, μ_P et μ_I, sur les tailles relatives des quatre régions, en montrant qu'ils jouent le même rôle pour $G^{\otimes}_{\mu}(P_x(e), I_x(e, E))$ et $G^{\rightarrow}_{\mu}(P_x(e), I_x(e, E))$. Malgré la différence d'interprétation de leurs régions, ils sont basés sur le même principe selon lequel si le poids de capacité associé à un critère est élevé, alors la zone qui minimise uniquement ce critère, en ignorant l'autre critère est grande, ce qui lui donne en effet plus d'importance. Par exemple, si μ_P est élevé, le seuil $1 - \mu_P$ est faible. Les deux intégrales augmentent la zone qui minimise la pénalité : pour $G^{\otimes}_{\mu}(P_x(e), I_x(e, E))$, elle correspond à la zone inférieure droite alors qu'elle est la zone supérieure gauche pour $G^{\rightarrow}_{\mu}(P_x(e), I_x(e, E))$. Dans le cadre des explications contre-factuelles, accorder une grande importance à la pénalité signifie que le nombre d'exemples contre-factuels pénalisés selon les modifications globales qu'ils effectuent est élevé. Dans les deux cas, l'aire de cette région est égale à $\mu_P(1 - \mu_I)$, ce qui montre qu'ils accordent globalement la même importance à la pénalité pour des valeurs données des paramètres. Ainsi,

$G_\mu^{\otimes}(P_x(e), I_x(e,E))$ et $G_\mu^{\rightarrow}(P_x(e), I_x(e,E))$ diffèrent dans la position de cette région, mais pas dans son importance.

6.3 GICE : Gödel Integrals for Counterfactual Explanation

Nous proposons d'utiliser l'intégrale de Gödel dont les définitions et propriétés ont été rappelées dans la section précédente pour agréger les critères de pénalité et d'incompatibilité qui définissent la fonction de coût globale de l'explication comme proposé dans le chapitre 3. En choisissant l'intégrale de Gödel comme fonction d'agrégation, la formalisation générale devient :

$$e^* = \underset{e \in \mathcal{E}_{x,f}}{\operatorname{argmin}} G_\mu^{\otimes}(P_x(e), I_x(e,E)) \tag{6.15}$$

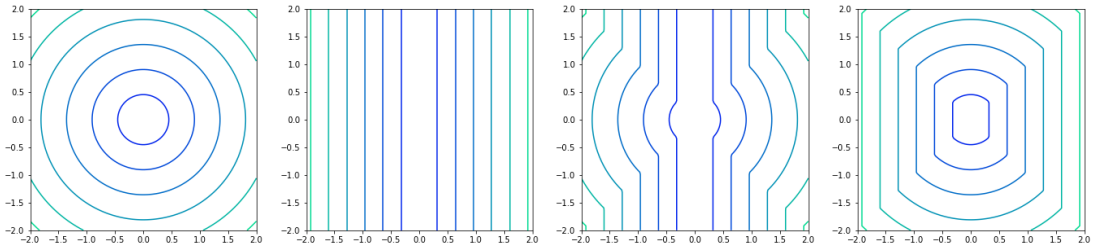
avec μ la capacité associée aux seuils de tolérance δ_P et δ_I de l'utilisateur. Cette agrégation est valable pour tout type de connaissances et tout type d'explications.

Dans cette section, nous présentons tout d'abord l'objectif considéré. Puis, nous décrivons les lignes de niveaux et la procédure de génération associées aux quatre opérateurs classiques utilisés par l'intégrale de Gödel. Enfin, nous présentons l'algorithme *Gödel Integrals for Counterfactual Explanation* (GICE) proposé.

6.3.1 Objectif et principe

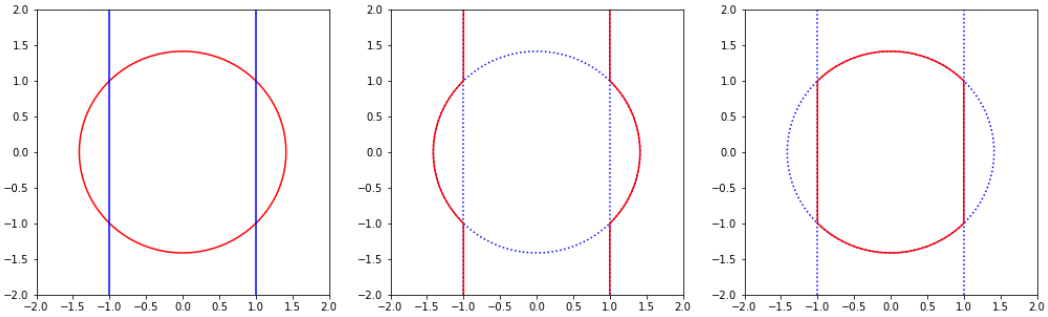
Comme dans le chapitre 4, nous étudions la génération d'une explication contrefactuelle pour un classifieur f et une instance x. L'espace de recherche $\mathcal{E}$ devient alors $\mathcal{E}_{x,f} = \{e \in \mathcal{X} | f(e) \notin f(x)\}$ et la fonction de pénalité s'écrit $P_x(e)$ car la proximité d'un exemple contre-factuel ne dépend pas du modèle. Nous considérons les connaissances utilisateur sous forme d'ensemble d'attributs $E = \{X_i, i = 1 \dots m\}$. Comme défini par les équations (4.1) et (4.2), la pénalité et l'incompatibilité correspondent respectivement à $P_x(e) = \frac{1}{Z_P} \parallel x - e \parallel$ et $I_x(e,E) = \frac{1}{Z_I} \parallel x - e \parallel_{\overline{E}}$. Les paramètres Z_P et Z_I sont des coefficients de normalisation permettant d'obtenir des valeurs dans $[0,1]$ qui correspondent respectivement à la pénalité et à l'incompatibilité maximales dans l'espace des données. Dans la suite du chapitre, nous omettrons les notations Z_P et Z_I.

Pour résoudre ce problème d'optimisation, nous proposons d'utiliser le même principe que dans le chapitre 3, la section 5.2 et la méthode Growing Spheres présentée dans la section 2.5.5 : nous générons des instances dans des couches croissantes autour de l'instance à expliquer, où la forme des couches est définie par les lignes de niveaux de la fonction de coût. Ainsi, nous présentons dans la section suivante les différentes lignes de niveaux associées à la fonction étudiée.



(a) $P_x(e)$ (b) $I_x(e,E)$ (c) $min(P_x(e), I_x(e,E))$ (d) $max(P_x(e), I_x(e,E))$

FIGURE 6.3 – Lignes de niveaux de quatre fonctions de coût : $P_x(e)$, $I_x(e,E)$, $min(P_x(e), I_x(e,E))$ et $max(P_x(e), I_x(e,E))$, dans l'espace des données en deux dimensions avec $E = \{X_2\}$.



(a) $P_x(e)$ et $I_x(e,E)$ (b) $min(P_x(e), I_x(e,E))$ (c) $max(P_x(e), I_x(e,E))$

FIGURE 6.4 – Décomposition des lignes de niveaux associées à la fonction maximum et minimum selon les lignes de niveaux de la pénalité et de l'incompatibilité. Fig. a : $P_x(e)$ (rouge) et $I_x(e,E)$ (bleu), Fig. b et c : $min(P_x(e), I_x(e,E))$ et $max(P_x(e), I_x(e,E))$ (rouge).

6.3.2 Lignes de niveaux

Sur la figure 6.3, nous illustrons en deux dimensions : X_1 en abscisses et X_2 en ordonnées, les lignes de niveaux pour une instance $x = (0,0)$ et une connaissance utilisateur $E = \{X_2\}$. Nous étudions le cas où $G_\mu^\otimes(P_x(e), I_x(e,E)) = \nu$ avec ν la valeur d'un niveau. Comme défini dans l'équation (6.7), $G_\mu^\otimes$ correspond à différentes fonctions d'agrégation selon la zone de l'espace des critères étudiée. Ainsi pour une instance e, $G_\mu^\otimes(P_x(e), I_x(e,E))$ correspond à un des quatre opérateurs de référence.

La première figure 6.3a est associé à la pénalité : les lignes de niveaux de la distance euclidienne forment des cercles concentriques. La figure 6.3b est associée à la fonction d'incompatibilité, les lignes sont verticales étant donné que la connaissance est l'attribut X_2. La figure 6.3c est associée au minimum et la figure 6.3d au maximum.

La figure 6.4 présente la décomposition des couches associées au minimum et au maximum à partir des couches de chacun des critères. La première figure présente une ligne de niveau associée à la pénalité (en rouge) et une ligne de niveau associée à l'incompatibilité (en bleu). Les deux autres figures représentent respectivement en rouge une ligne de niveau associée au minimum et une associée au maximum. Ces lignes sont obtenues en supprimant une partie des lignes de la pénalité et de l'incompatibilité, représentée par des pointillés bleus. On remarque que les couches associées au minimum

et au maximum ont une forme complexe mais la décomposition proposée permet de facilement générer des instances dans chacune de ces couches.

6.3.3 Génération uniforme des couches

Pour définir la forme des couches à chaque étape, nous définissons tout d'abord les deux couches de référence $\mathcal{B}_P$ et $\mathcal{B}_I$ associées respectivement à la pénalité (figure 6.3a) et l'incompatibilité (figure 6.3b) :

$$\mathcal{B}_P(x, \nu, \nu + \epsilon) = \{e \in \mathcal{X} | \nu < P_x(e) < \nu + \epsilon\}$$

$$\mathcal{B}_I(x, \nu, \nu + \epsilon, E) = \{e \in \mathcal{X} | \nu < I_x(e, E) < \nu + \epsilon\}$$

Pour générer des instances dans la couche $\mathcal{B}_P$, nous utilisons l'algorithme GCE (algorithme 2) présenté dans la section 4 avec une connaissance nulle : $E = \emptyset$ et une valeur de $\lambda = 0$. Pour générer des instances dans la couche $\mathcal{B}_I$, nous utilisons un nouvel algorithme en deux étapes. La première étape consiste générer des instances dans une couche sphérique, en se restreignant au attributs $\overline{E}$. La seconde étape consiste à associer à ces instances les valeurs des attributs E uniformément générées.

Comme vu précédemment, les lignes de niveaux associées au minimum et au maximum peuvent être décomposées à partir de celles associées à la pénalité et à l'incompatibilité. Ainsi dans les deux cas pour l'étape de génération, nous générons des instances dans les couches $\mathcal{B}_P(x, \nu, \nu + \epsilon)$ et $\mathcal{B}_I(x, \nu, \nu + \epsilon, E)$. Puis, nous gardons uniquement les instances qui vérifient $\nu < \min(P_x(e), I_x(e, E)) < \nu + \epsilon$ pour la fonction minimum et $\nu < \max(P_x(e), I_x(e, E)) < \nu + \epsilon$ pour la fonction maximum.

6.3.4 Algorithme GICE

Nous présentons dans cette section la méthode GICE, dont le pseudo-code est fourni dans l'algorithme 4. GICE propose de générer des instances dans des couches de plus en plus grandes à chaque étape. Comme évoqué dans la section précédente, quatre cas de figure sont considérés (lignes 4, 9, 12 et 15). Les deux cas les plus simples considèrent qu'un seul des critères est satisfaisant : lignes 9 à 11 lorsque la pénalité est satisfaisante et lignes 12 à 14 lorsque l'incompatibilité est satisfaisante. Dans les deux cas, les instances sont générées dans la couche associée au critère non satisfaisant, $\mathcal{B}_I$ et $\mathcal{B}_P$ respectivement.

Puis, nous considérons le cas où les deux critères sont satisfaisants (lignes 4 à 8), la fonction associée est le minimum : les deux couches associées à la pénalité et à l'incompatibilité sont générées. Seules les instances appartenant à la couche souhaitée sont gardées, c'est-à-dire celles dont le minimum des valeurs des deux critères est compris entre ν et $\nu + \epsilon$.

Enfin, dans le dernier cas de figure les deux critères sont non satisfaisants (lignes 15 à 19), la fonction associée est le maximum : les deux couches associées à la pénalité et à

Algorithm 4 Gödel Integrals for Counterfactual Explanation

Require: $f : \mathcal{X} \longrightarrow \{0,1\}$
Require: $x \in \mathcal{X}$ instance considérée
Require: E, la connaissance utilisateur
Require: δ_P, δ_I seuils
Require: ν_0, ϵ, n paramètres
Ensure: $e^* = \text{argmin}_{e \in \mathcal{E}_{x,f}} G_\mu^\otimes(P_x(e), I_x(e,E))$
1: $\nu = \nu_0$
2: $\mathcal{Z} = \emptyset$
3: **while** $\nexists e \in \mathcal{Z}, f(e) \neq f(x)$ **do**
4: **if** $\nu \leq \delta_P$ and $\nu \leq \delta_I$ **then**
5: Générer $\mathcal{Z}_p = \{z_j^p\}_{j \leq n} \sim U(\mathcal{B}_P(x, \nu, \nu+\epsilon))$ avec GCE, (algorithme 2)
6: Générer $\mathcal{Z}_i = \{z_j^i\}_{j \leq n} \sim U(\mathcal{B}_I(x, \nu, \nu+\epsilon, E))$
7: $\mathcal{Z} = \{z \in \mathcal{Z}_I \cup \mathcal{Z}_P | \nu < \min(P_x(z), I_x(z,E)) < \nu + \epsilon\}$
8: $\mathcal{Z} \leftarrow \{z \in \mathcal{Z} | P_x(z) < \delta_P \ \wedge \ I_x(z,E) < \delta_I\}$
9: **else if** $\nu \leq \delta_P$ **then**
10: Générer $\mathcal{Z} = \{z_j\}_{j \leq n} \sim U(\mathcal{B}_I(x, \nu, \nu+\epsilon, E))$
11: $\mathcal{Z} \leftarrow \{z \in \mathcal{Z} | P_x(z) < \delta_P\}$
12: **else if** $\nu \leq \delta_I$ **then**
13: Générer $\mathcal{Z} = \{z_j\}_{j \leq n} \sim U(\mathcal{B}_P(x, \nu, \nu+\epsilon))$ avec GCE, (algorithme 2)
14: $\mathcal{Z} \leftarrow \{z \in \mathcal{Z} | I_x(z,E) < \delta_I\}$
15: **else**
16: Générer $\mathcal{Z}_P = \{z_j^p\}_{j \leq n} \sim U(\mathcal{B}_P(x, \nu, \nu+\epsilon))$ avec GCE, (algorithme 2)
17: Générer $\mathcal{Z}_I = \{z_j^i\}_{j \leq n} \sim U(\mathcal{B}_I(x, \nu, \nu+\epsilon, E))$
18: $\mathcal{Z} = \{z \in \mathcal{Z}_I \cup \mathcal{Z}_P | \nu < \max(P_x(z), I_x(z,E)) < \nu + \epsilon\}$
19: **end if**
20: $\nu \longleftarrow \nu + \epsilon$
21: **end while**

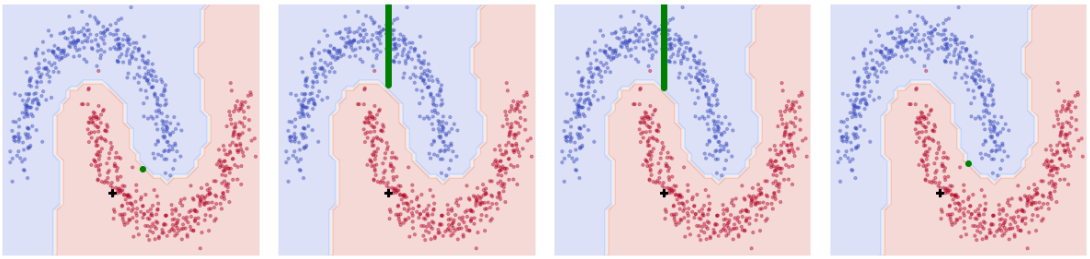
l'incompatibilité sont générées. Cette fois-ci les seules instances gardées vérifient que le maximum des valeurs des deux critères est entre ν et $\nu + \epsilon$.

6.4 Exemples illustratifs

Cette section illustre des exemples contre-factuels générés par l'algorithme GICE sur un jeu de données en deux dimensions en considérant principalement l'intégrale de Gödel basée sur la conjonction $G_\mu^\otimes$. Tout d'abord, elle détaille le protocole expérimental considéré. Puis, elle présente quatre instanciations de l'intégrale de Gödel pour des valeurs extrêmes des paramètres μ_P et μ_I. La troisième section analyse les résultats obtenus pour différentes valeurs des paramètres. Enfin, la dernière section examine les résultats fournis par l'intégrale de Gödel basée sur l'implication $G_\mu^\rightarrow$.

6.4.1 Protocole expérimental

Les expérimentions sont menées avec l'ensemble de données 2D Half-Moons dont les dimensions sont notées X_1 et X_2. Sur les figures 6.5 et 6.6, les régions bleues et rouges



(a) $P_x(e)$ (b) $I_x(e, E)$ (c) $\min(P_x(e), I_x(e, E))$ (d) $\max(P_x(e), I_x(e, E))$

FIGURE 6.5 – Explications contre-factuelles générées pour des cas de référence, définies par les équations (6.11), (6.12), (6.13) et (6.14)

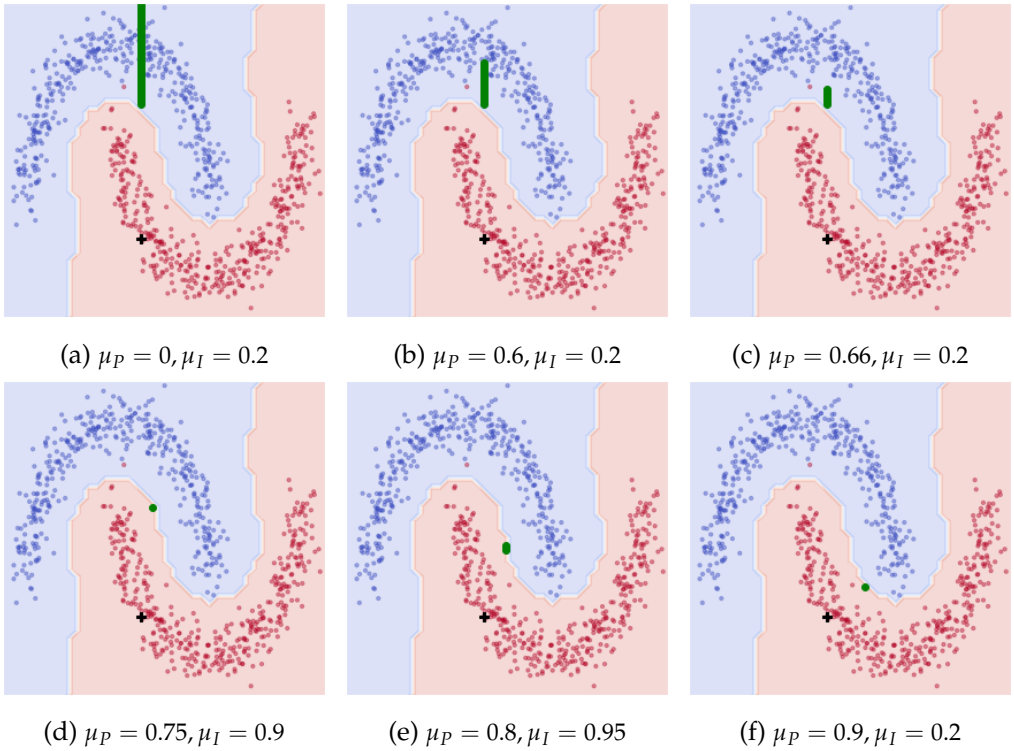
représentent les classes prédites, les points plus foncés les exemples d'apprentissage ; la frontière de décision du classifieur SVM avec un noyau gaussien entraîné est représentée en blanc (précision du test : 0.99). Les connaissances de l'utilisateur considérées sont le singleton $E = \{X_2\}$. Pour permettre des comparaisons visuelles, toutes les expérimentions utilisent la même instance $x = (-0.5, -1)$, représentée par une croix noire. La pénalité $P_x(e)$ est définie comme la distance euclidienne normalisée $P_x(e) = \|x - e\|$, l'incompatibilité $I_x(e, E)$ comme la distance euclidienne normalisée sur l'attribut extérieur à E, $I_x(e, E) = \|x - e\|_{X_1}$. Le problème d'optimisation n'ayant pas de solution unique car nous considérons une fonction d'agrégation qui n'est pas strictement monotone, l'ensemble des solutions est représenté par des points verts. Cela implique que nous n'effectuons pas de différence entre plusieurs explications qui ont la même valeur pour un des deux critères.

6.4.2 Cas de référence

Dans cette section, nous examinons d'abord les quatre fonctions d'agrégation classiques, qui correspondent à des cas extrêmes des intégrales de Gödel comme montré dans les équations (6.11) à (6.14). La figure 6.5 montre les exemples contre-factuels obtenus dans chaque cas, illustrant leur diversité attendue. Ceux-ci sont obtenus en générant tout d'abord des instances dans tout l'espace des données, puis en choisissant ceux qui appartiennent à la classe souhaitée et qui minimisent la fonction de coût considérée.

La figure 6.5a constitue l'explication de référence. La figure 6.5b montre les explications qui minimisent l'incompatibilité. Pour le x considéré, il est possible de trouver des exemples contre-factuels totalement compatibles avec les connaissances de l'utilisateur : les explications générées sont donc des points situés à la verticale de x qui appartiennent à la classe bleue, avec une incompatibilité égale à 0, on peut remarquer qu'une telle fonction de coût réduite à cette incompatibilité, est rarement utilisée.

La figure 6.5c est similaire à la figure 6.5b car on considère un cas particulier où $I_x(e, E)$ peut être égal à 0, alors que $P_x(e)$ ne peut pas être nul. Le minimum de la pénalité et de l'incompatibilité peut alors valoir 0, elle correspond à la valeur possible de l'incompatibilité. Le minimum conduit donc aux mêmes résultats que l'incompatibilité. Enfin, la figure 6.5d est associée à la fonction maximum ; les explications générées sont



(a) $\mu_P = 0, \mu_I = 0.2$ (b) $\mu_P = 0.6, \mu_I = 0.2$ (c) $\mu_P = 0.66, \mu_I = 0.2$

(d) $\mu_P = 0.75, \mu_I = 0.9$ (e) $\mu_P = 0.8, \mu_I = 0.95$ (f) $\mu_P = 0.9, \mu_I = 0.2$

FIGURE 6.6 – Exemples contre-factuels obtenus en minimisant $G^{\otimes}_{\mu}(P_x(e), I_x(e, E))$ pour différentes valeurs de μ_P et μ_I.

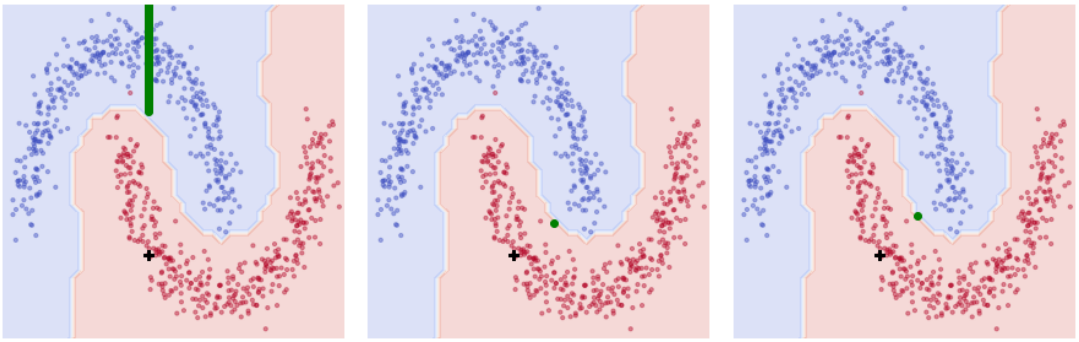
situées à des positions où la pénalité l'emporte sur l'incompatibilité : l'explication finale est plus proche que compatible.

6.4.3 Cas général : Intégrale de Gödel basée sur la conjonction

La figure 6.6 montre les explications générées lors de l'utilisation de $G^{\otimes}_{\mu}(P_x(e), I_x(e, E))$ pour d'autres valeurs, moins extrêmes, des paramètres μ_P et μ_I, choisies pour illustrer la variété des résultats auxquels elles conduisent. Six cas peuvent être distingués, illustrant l'intérêt et l'expressivité de cet opérateur d'agrégation.

Sur la figure 6.6a, identique aux figures 6.5b et 6.5c, l'ensemble des exemples contre-factuels générés est l'ensemble des points totalement compatibles de l'autre classe, c'est-à-dire ceux pour lesquels $I_x(e, E) = 0$. Pour l'instance considérée x, il peut être obtenu chaque fois que $\mu_P < 0.7$, c'est-à-dire $\delta_P > 0.3$. Cela vient du fait que l'exemple contre-factuel le plus proche a une pénalité de 0.3. Lorsque μ_P augmente, au-delà du seuil δ_P, le nombre d'explications générées diminue, comme l'illustrent les figures 6.6b et 6.6c ($\mu_P = 0.6$ et 0.66 respectivement). Ceci montre l'impact de la prise en compte du seuil α dans les intégrales de Gödel : même si les explications sont complètement compatibles, si elles ne satisfont pas la contrainte imposée par la pénalité, elles sont rejetées.

Les figures 6.6d et 6.6e représentent un compromis entre les cas extrêmes des figures 6.6c et 6.6f, c'est-à-dire des compromis entre la pénalité et l'incompatibilité. Nous illustrons ces cas avec un seuil de pénalité élevé, les exemples contre-factuels générés sont les instances les plus compatibles qui satisfont la contrainte de pénalité. Dans les



(a) $\mu_P = 0.8, \mu_I = 0.8$ (b) $\mu_P = 0.8, \mu_I = 1$ (c) $\mu_P = 0.95, \mu_I = 1$

FIGURE 6.7 – Exemples contre-factuels obtenus en minimisant $G_\mu^{\rightarrow}(P_x(e), I_x(e,E))$ pour différentes valeurs de δ_P et δ_I ($+ : x$,$\bullet : e^*$).

figures représentées ici, au moins une des contraintes est satisfaite. La figure 6.5d représente la fonction maximale si aucune des contraintes n'est vérifiée ($\mu_P > 0,9$ et $\mu_I > 0,95$). Ces valeurs sont associées à des contraintes très fortes. La figure 6.6e en est une variante, avec une plus grande tolérance sur la valeur de la pénalité.

Sur la figure 6.6f, qui est identique à la figure 6.5a, un seul exemple contre-factuel est généré, qui correspond au point le plus proche de l'autre classe, c'est-à-dire celui dont la pénalité est la plus faible. Ce cas est obtenu lorsque la contrainte imposée par la pénalité est trop forte, c'est-à-dire lorsque μ_P est trop élevé par rapport à l'incompatibilité. Dans ce cas, il est impossible de trouver une explication compatible, le processus d'optimisation se concentre donc sur la minimisation de la pénalité.

Cette expérimentation montre à quel point l'utilisation de l'intégrale de Gödel permet une diversité des explications selon les besoins utilisateur. Dans cet exemple, nous obtenons six cas différents contre un seul cas de figure dans un cas classique d'agrégation.

6.4.4 Cas général : Intégrale de Gödel basée sur l'implication

Cette section illustre, de la même façon, les résultats obtenus avec l'intégrale de Gödel basée sur l'implication $G_\mu^{\rightarrow}$ au lieu de $G_\mu^{\otimes}$. Pour obtenir les explications associées à cette intégrale, nous utilisons une version modifiée de l'algorithme GICE (algorithme 4) : les lignes 9 et 12 sont inversées.

La figure 6.7 montre les exemples contre-factuels obtenus pour la même instance, seuls trois comportements sont observés : la pénalité, l'incompatibilité et le maximum sont minimisés. Nous considérons ici un cas de figure où il existe un exemple contre-factuel $\tilde{e}$, tel que $I_x(\tilde{e}, E) = 0$. Lorsque $\mu_I = 1$, pour tout $e \in \mathcal{E}_{x,f}$, l'intégrale de Gödel peut s'écrire comme :

$$G_\mu^{\rightarrow}(P_x(e), I_x(e,E)) = \min(P_x(e), I_x(e,E)) \text{ ou } G_\mu^{\rightarrow}(P_x(e), I_x(e,E)) = I_x(e,E)$$

L'explication qui minimise cette intégrale est alors $\tilde{e}$, c'est-à-dire l'ensemble des explications totalement compatible.

Lorsque $\mu_I \neq 1$, deux cas de figures sont étudiés selon que l'explication e_{ref} qui minimise la pénalité vérifie ou non la contrainte de pénalité. Pour tout $e \in \mathcal{E}_{x,f}$, l'intégrale de Gödel s'écrit :

$$\begin{aligned} G_{\mu}^{\rightarrow}(P_x(e), I_x(e,E)) &= P_x(e) && \text{si } P_x(e_{ref}) \leq \delta_P \\ G_{\mu}^{\rightarrow}(P_x(e), I_x(e,E)) &= \max(P_x(e), I_x(e,E)) && \text{si } P_x(e_{ref}) > \delta_P \end{aligned}$$

Nous obtenons ainsi seulement trois cas de figure. On remarque que l'intégrale de Gödel basée sur la conjonction est plus riche que celle basée sur l'implication, elle permet d'obtenir une plus grande diversité des explications. Ces deux intégrales se basent sur des principes différents, qui favorisent ou pénalise un critère selon les cas considérés. Afin de choisir entre ces deux intégrales, une étude plus poussée est nécessaire. Notamment, il serait nécessaire de renforcer une propriété ou de considérer une nouvelle propriété qui permet de choisir une variante de l'intégrale de Gödel.

6.5 Résultats expérimentaux

Cette section présente les résultats de quatre expériences menées pour évaluer quantitativement les explications générées par l'algorithme GICE en le comparant à différents compétiteurs selon différentes métriques. La première expérience illustre les métriques considérées sur des instances particulières du jeu de données Californie. La seconde généralise à l'ensemble des données test, pour les jeux de données Half-Moons et Boston. La troisième vérifie que l'utilisation de l'intégrale de Gödel permet d'obtenir le comportement souhaité qui est de générer des explications qui vérifient les contraintes. La quatrième compare la méthode GICE avec la méthode KICE proposée dans le chapitre 4.

6.5.1 Protocole expérimental

Cette section présente tour à tour la configuration de GICE considérée, les compétiteurs, les métriques et le protocole mis en œuvre pour l'étude de GICE.

Compétiteurs Nous comparons les résultats fournis par GICE, notés e_{GICE}, à ceux de trois concurrents. Le premier compétiteur est l'algorithme Growing Spheres (Laugel et al., 2018a), qui résout le problème d'optimisation considérant uniquement la pénalité définie comme la distance euclidienne normalisée. D'après la caractérisation établie dans la section 6.2, il correspond à un cas extrême de l'intégrale de Gödel, où $\mu_P = 1$ et $\mu_I = 0$. Growing Spheres fournit une unique explication notée e_{ref}.

Le second compétiteur résout le problème d'optimisation considérant uniquement l'incompatibilité, définie comme la distance euclidienne sur les attributs inconnus normalisés qui correspond à un cas extrême de l'intégrale de Gödel où $\mu_P = 0$ et $\mu_I = 1$.

Plusieurs explications peuvent minimiser cette fonction, leur ensemble est noté e_{user}. Il correspond à l'ensemble des explications les plus compatibles.

Le troisième compétiteur résout le problème d'optimisation considérant pour l'agrégation, la moyenne pondérée entre la pénalité et l'incompatibilité. Il correspond au résultat fourni par KICE noté e_{KICE}. Pour normaliser les critères, nous avons choisi $Z_P = \sqrt{32}$ et $Z_I = 4$ qui correspondent à la pénalité et à l'incompatibilité maximale, ainsi nous fixons une valeur de λ égale à 2, pour avoir une moyenne pondérée équivalente à une simple moyenne.

Métriques Afin d'analyser les explications obtenues, nous comparons les résultats selon trois métriques, qui correspondent aux fonctions de coût minimisées par chaque compétiteur : nous étudions la pénalité, l'incompatibilité et leur agrégation par l'intégrale de Gödel.

Paramètres Pour chaque expérimentation, nous fixons des paramètres μ_P et μ_I différents, de manière à illustrer différents types d'explications, celles qui vérifient les deux contraintes, celles qui en vérifient qu'une seule ou celles qui n'en vérifient aucune.

Pour le jeu de données Californie, nous choisissons $\delta_P = 0.02$ et $\delta_I = 0.01$: nous fixons des valeurs faibles parce que les valeurs de pénalité et d'incompatibilité obtenues pour l'instance choisie sont faibles. Nous avons choisi de normaliser les distances euclidiennes pour calculer les deux métriques, ce qui explique les valeurs faibles. Pour la seconde expérimentation qui évalue quantitativement la méthode GICE, nous choisissons d'autres valeurs des couples (μ_P, μ_I), comme détaillé dans la section 6.5.3.

Protocole Chaque jeu de données est divisé en ensembles d'entraînement et de test (80%-20%). Nous entraînons un modèle SVM sur l'ensemble d'entraînement. Pour chaque instance de l'ensemble test, nous générons trois ensembles d'explications e_{ref}, e_{user} et e^*. Chaque exemple contre-factuel généré est évalué par les métriques listées ci-dessus.

Enfin, nous calculons la moyenne et l'écart-type de ces quantités quand on fait varier l'instance dans le jeu de test. Lorsque le résultat fourni contient plusieurs explications c'est-à-dire plusieurs exemples contre-factuels (pour e_{user} et e^*), nous indiquons les valeurs minimale, maximale et moyenne des valeurs obtenues par ces exemples.

6.5.2 Exemple de résultats de GICE sur la base Californie

La première expérience illustre les métriques considérées pour une instance particulière présentée dans la première ligne du tableau 6.2. La deuxième ligne donne la valeur de l'exemple contre-factuel e_{ref}. Les lignes suivantes illustrent les résultats obtenus en tenant compte de la connaissance utilisateur $E = \{a_0, a_1, a_3, a_4\}$ avec le sens des attributs décrit dans le tableau 4.2. Dans cette configuration, e_{user} contient plusieurs exemples contre-factuels, le tableau indique les valeurs de $e_{user}^{min,P}$ et $e_{user}^{min,P}$ respectivement

	a_0	a_1	$\boldsymbol{a_2}$	a_3	a_4	$\boldsymbol{a_5}$	$\boldsymbol{a_6}$
x_0	-122.6	37.9	48	5.62	0.92	641	8.63
e_{ref}	+0.5	+0.5	+2	-0.54	+0.14	-10	-0.31
$e_{user}^{min,P}$	-0.2	-3	0	-3.44	+1.44	+3	+0.01
$e_{user}^{max,P}$	+0.5	-0.3	0	+127.8	+32.94	+3	+0.01
e^*	+0.9	+2.7	0	-2.58	+0.23	-8	+0.01

TABLE 6.2 – Exemples contre-factuels e_{ref}, $e_{user}^{min,P}$, $e_{user}^{max,P}$ et e_{GICE} pour l'instance x_0 indiquée dans la première ligne. Pour les attributs inconnus, notés en gras, la couleur indique le niveau de modification : vert = pas de modifications, orange = faible, rouge = élevée.

définis comme :

$$e_{user}^{min,P} = \underset{e \in e_{user}}{\operatorname{argmin}} P_x(e)$$
$$e_{user}^{max,P} = \underset{e \in e_{user}}{\operatorname{argmax}} P_x(e)$$

Enfin, la dernière ligne du tableau indique l'unique exemple contre-factuel fourni par GICE. Le tableau 6.3 quant à lui présente les valeurs des trois métriques $P_x(e)$, $I_x(e, E)$ et $G_\mu^\otimes(P_x(e), I_x(e, E))$ pour chacun de ces exemples contre-factuels, en indiquant en couleurs s'ils satisfont les contraintes exprimées par le biais des valeurs $\mu_P = 0.98$ et $\mu_I = 0.99$.

Comme attendu, parmi les exemples contre-factuels considérés e_{ref} minimise la pénalité. De plus, sa pénalité est inférieure au seuil $\delta_P = 1 - \mu_P = 0.02$, il vérifie la contrainte selon ce critère. Par contre, son incompatibilité est supérieure à $\delta_I = 1 - \mu_I = 0.01$ et ne satisfait donc pas cette contrainte. En effet, on observe dans le tableau 6.2 qu'il effectue de grandes modifications sur les attributs inconnus alors qu'elles sont faibles sur les attributs connus : e_{ref} ne dispose pas de connaissances, selon ces deux catégories d'attributs et sélectionne des modifications qui permettent une minimisation globale de la pénalité.

Les deux exemples illustrés de e_{user} quant à elles ne modifient pas l'attribut a_2 et effectuent de très petites modifications, presque négligeables, sur les deux autres attributs inconnus. Ainsi, leur incompatibilité est faible et ils vérifient la contrainte. Par contre, leurs pénalités sont supérieures au seuil, ce qui indique qu'elles ne vérifient pas la contrainte de pénalité.

L'explication fournie par GICE est donc la seule à satisfaire les deux contraintes simultanément : e_{GICE} ne modifie pas a_2, elle effectue une petite modification selon a_6 et une modification intermédiaire selon a_5, offrant un meilleur compromis. La combinaison par l'intégrale de Gödel permet de favoriser la génération d'une explication qui vérifie les deux contraintes.

6.5.3 Évaluation de la méthode GICE

Cette section généralise les observations réalisées sur une instance de référence des données Half-Moons, présentées dans la section 6.4, en évaluant quantitativement les

	$P_x(e)$	$I_x(e,E)$	$G_\mu^\otimes((P_x(e), I_x(e,E)))$
e_{ref}	0.0082	0.0189	0.0189
$e_{user}^{min,P}$	0.0372	0.0001	0.0372
$e_{user}^{max,P}$	0.8919	0.0001	0.8919
e^*	0.0184	0.0004	0.0004

TABLE 6.3 – Valeurs des métriques : $P_x(e)$, $I_x(e,E)$ et $G_\mu^\otimes(e)$ pour une instance x du jeu de données Californie avec $\delta_P = 0.02$ et $\delta_I = 0.01$. Représentation en couleurs de la satisfaction (vert) ou non (rouge) des contraintes.

exemples contre-factuels générés selon les trois métriques introduites ci-dessus lorsqu'on fait varier l'instance sur tout l'ensemble de données de test.

Le tableau 6.3 donne la moyenne et l'écart-type des quatre méthodes considérées qui génèrent respectivement e_{ref}, e_{user}, e_{GICE} et e_{KICE}. Comme e_{user} et e_{GICE} peuvent constituer des ensembles d'exemples contre-factuels, et non seulement des singletons comme e_{ref} et e_{KICE}, le tableau indique la moyenne et l'écart-type des valeurs minimale, maximale et moyenne observées sur ces ensembles.

Configurations Pour KICE, nous considérons $\lambda = 2$. Pour GICE, trois valeurs des paramètres (δ_P, δ_I) sont étudiées, elles sont choisies en fonction des valeurs de pénalité et d'incompatibilité observés dans les résultats de e_{ref} et e_{user}. Le premier couple $(0.25, 0.1)$ est du même ordre de grandeur que la pénalité de e_{user} et que l'incompatibilité de e_{ref}. Pour le second couple, nous choisissons $(0.15, 0.1)$ car nous souhaitons une valeur de μ_P entre 0.10 et 0.21, c'est-à-dire entre la valeur moyenne de pénalité de e_{ref} et de e_{user}. Cela permet d'étudier les résultats lorsque la contrainte selon la pénalité est plus forte que pour le premier couple. Enfin, nous considérons le couple $(0.2, 0.04)$, il est choisi à partir des résultats associés aux précédents couples de manière à avoir une contrainte d'incompatibilité non vérifiée en moyenne et une contrainte de pénalité vérifiée.

Résultats Comme attendu, nous remarquons d'abord que les explications e_{ref}, e_{user} et e^* minimisent respectivement la pénalité, l'incompatibilité et la fonction de coût.

La méthode qu'on propose avec l'intégrale de Gödel a pour but d'intégrer des connaissances utilisateur. On remarque que le choix de cette agrégation permet bien d'avoir une meilleure compatibilité que l'explication de référence e_{ref} et une meilleure pénalité que l'explication totalement compatible e_{user}, un compromis entre les deux critères est bien effectué. Par rapport à e_{KICE} qui effectue également un compromis, les explications proposées e_{GICE} sont en moyenne plus compatibles mais plus éloignées. L'inconvénient de KICE est qu'il n'est pas possible de prévoir de le gain en incompatibilité et la perte de pénalité. L'intégrale de Gödel considère des paramètres δ_P et δ_I, ce qui permet de prévoir les valeurs des critères des explications finales.

Nous étudions l'impact des deux seuils δ_P et δ_I sur le résultat proposé. Pour observer l'impact du seuil δ_P, nous comparons les explications e_{GICE} pour δ_P égal à 0.25 et 0.15. On remarque que considérer une contrainte forte sur la pénalité (δ_P faible) implique une

		$P_x(e)$	$I_x(e,E)$	$G_\mu^\otimes((P_x(e),I_x(e,E)))$
e_{ref}		**0.10 ± 0.03**	0.09 ± 0.04	0.08 ± 0.03
e_{user}	min	0.21 ± 0.12	**0.02 ± 0.04**	0.18 ± 0.20
	max	0.44 ± 0.17	**0.02 ± 0.04**	0.49 ± 0.12
	moyenne	0.34 ± 0.32	**0.02 ± 0.04**	0.34 ± 0.13
e_{KICE}		0.11 ± 0.03	0.07 ± 0.05	0.07 ± 0.05
e_{GICE} ($\delta_P = 0.25, \delta_I = 0.1$)	min	0.18 ± 0.06	0.03 ± 0.05	**0.03 ± 0.05**
	max	0.23 ± 0.04	0.03 ± 0.05	**0.03 ± 0.05**
	moyenne	0.21 ± 0.04	0.03 ± 0.05	**0.03 ± 0.05**
e_{GICE} ($\delta_P = 0.15, \delta_I = 0.1$)	min	0.13 ± 0.02	0.06 ± 0.05	0.07 ± 0.06
	max	0.15 ± 0.01	0.06 ± 0.05	0.07 ± 0.06
	moyenne	0.14 ± 0.02	0.06 ± 0.05	0.07 ± 0.06
e_{GICE} ($\delta_P = 0.2, \delta_I = 0.04$)	min	0.16 ± 0.04	0.04 ± 0.05	0.05 ± 0.05
	max	0.19 ± 0.03	0.05 ± 0.05	0.05 ± 0.05
	moyenne	0.17 ± 0.03	0.05 ± 0.05	0.05 ± 0.05

TABLE 6.4 – Résultats obtenus avec les trois approches considérées sur le jeu de données Half-Moons pour les métriques : $P_x(e)$, $I_x(e,E)$ et $G_\mu^\otimes((P_x(e),I_x(e,E)))$ pour différentes valeurs des paramètres δ_P et δ_I.

valeur moyenne de la pénalité qui diminue. De même, l'impact du seuil δ_I est étudié en comparant les explications e_{GICE} pour δ_I égal à 0.1 et 0.04. On remarque comme pour la pénalité que considérer une contrainte forte sur l'incompatibilité (δ_I faible) implique une valeur moyenne d'incompatibilité qui diminue.

6.5.4 Évaluation du respect des contraintes utilisateur

Pour analyser plus finement les résultats fournis par les évaluations par métriques, nous proposons dans cette section une autre étude qui examine la distribution des exemples contre-factuels générés par rapport aux quatre zones définies par le respect ou non des deux contraintes de pénalité et d'incompatibilité.

Protocole Nous considérons ici $\delta_P = 0.2$ et $\delta_I = 0.04$, fixés selon les résultats de l'étude menée dans la section précédente qui montre qu'avec ces valeurs une seule des contraintes est satisfaite pour de nombreuses instances de référence. Nous comptabilisons pour chaque zone de l'espace des critères le nombre d'instances possédant un exemple contre-factuel situé dans cette zone. Le total des quatre valeurs peut alors être supérieur au nombre total d'instances si certaines possèdent des explications contre-factuelles appartenant à plusieurs zones.

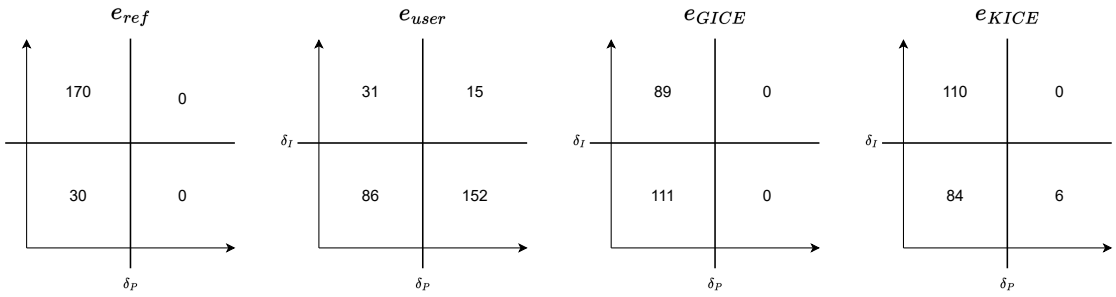


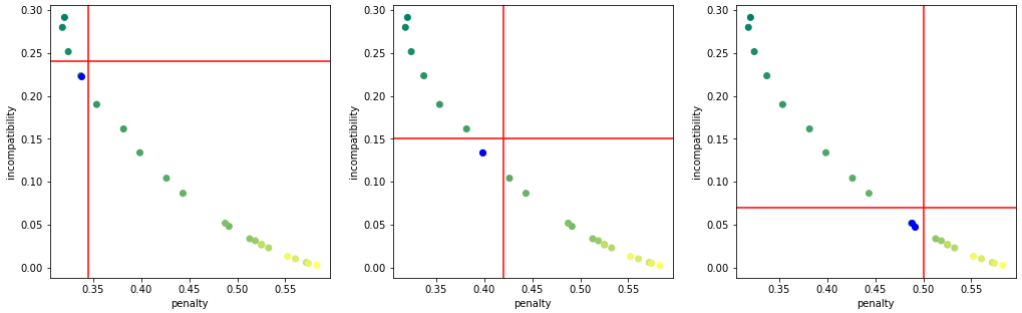
FIGURE 6.8 – Nombre d'instances ayant des explications contre-factuelles dans chacune des zones de l'espace des critères, de gauche à droite : minimisation de la pénalité, de l'incompatibilité, de l'intégrale de Gödel et de la moyenne.

Résultats La figure 6.8 présente les résultats obtenus pour les quatre compétiteurs considérés e_{ref}, e_{user}, e_{GICE} et e_{KICE}. e_{GICE} satisfait majoritairement les deux contraintes. Dans la zone en bas à gauche il comporte le plus d'exemples contre-factuels : 60% contre 15% pour e_{ref}, 30% pour e_{user} et 42% pour e_{KICE}. Cela confirme bien le comportement souhaité par l'opérateur choisi. Dans le pire cas où les explications ne vérifient aucune contrainte (en haut à droite) seul e_{user} a des exemples contre-factuels.

Pour les deux autres cas de figure, nous comparons principalement e_{GICE} à e_{ref} et à e_{user}. Tout d'abord, e_{ref} qui minimise la pénalité satisfait bien la contrainte de pénalité pour toutes les instances de référence : les 200 points sont situés dans les deux régions associées à une pénalité inférieure à δ_P. Ce résultat est compatible avec la valeur moyenne de la pénalité indiquée dans le tableau 6.4. En revanche, comme e_{ref} ne tient pas compte des connaissances utilisateur, il n'est pas surprenant que dans la majorité des cas (85%) la contrainte d'incompatibilité n'est pas vérifiée. Ensuite, e_{user} qui minimise l'incompatibilité satisfait majoritairement la contrainte d'incompatibilité. Par contre e_{user} ne considère pas la pénalité, ainsi la majorité des explications ne vérifient pas la contrainte de pénalité (59%). e_{GICE} considère ces deux contraintes, elle comporte donc plus d'exemples qui vérifient la contrainte d'incompatibilité que e_{ref} (55.5% contre 15%) et plus d'exemples qui vérifient la contrainte de pénalité que e_{user} (41% contre 100%).

6.5.5 Comparaison des paramètres de KICE et de GICE

Cette section propose une comparaison plus fine de KICE et GICE en examinant l'impact de leurs paramètres et l'expressivité qu'ils permettent. Elle considère la même instance de référence des données Half-Moons que la section 4.5.1 et étudie les explications contre-factuelles obtenues pour différentes valeurs de λ, δ_P et δ_I. Les trois graphiques de la figure 6.9 représentent en vert pour une instance donnée les explications contre-factuelles obtenues avec la méthode KICE pour différentes valeurs du paramètre λ dans l'espace des critères et en bleu les explications obtenues avec GICE pour les paramètres δ_P et δ_I. On remarque que les exemples en bleu sont communs aux deux méthodes, or il est plus simple de les obtenir avec la méthode GICE que KICE car déterminer les valeurs de δ_P et δ_I associées à cet exemple est plus facile que de déterminer la



(a) $\lambda = 0.45$, $\delta_P = 0.24$, $\delta_I = 0.34$

(b) $\lambda = 0.9$, $\delta_P = 0.15$, $\delta_I = 0.42$

(c) $\lambda = 1.35$, $\delta_P = 0.5$, $\delta_I = 0.07$

FIGURE 6.9 – Comparaison des paramètres λ, δ_P et δ_I associés à trois explications différentes.

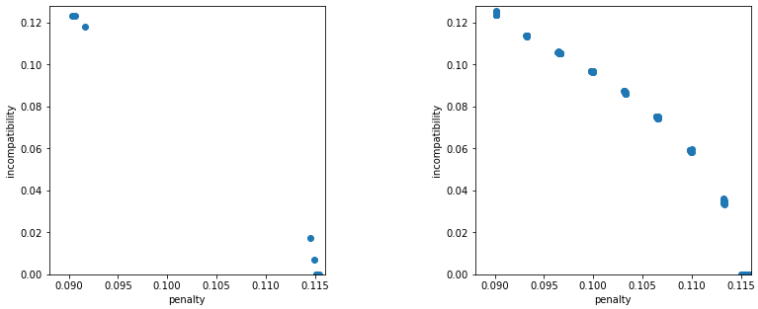


FIGURE 6.10 – Explications obtenues pour une instance donnée avec les méthodes KICE (gauche) et GICE (droite) pour différentes valeurs de λ, δ_P et δ_I.

valeur de λ.

Le graphique de gauche de la figure 6.10 représente les explications obtenues avec KICE pour différentes valeurs de λ dans l'espace des critères et sur le graphique de droite les explications obtenues avec GICE pour différentes valeurs de δ_P et δ_I pour une instance différente de la précédente. Les explications obtenues avec la méthode KICE se situent dans deux zones de l'espace des critères : l'une associée à une pénalité faible et une incompatibilité élevée et une autre associée à une incompatibilité faible et une pénalité élevée, il n'existe aucune explication qui a une pénalité entre 0.09 et 0.112. Quant aux explications obtenues avec la méthode GICE, les espaces entre les explications sont réguliers. Pour cette instance, on remarque bien qu'avec la méthode GICE, on a la possibilité de couvrir beaucoup plus l'espace des critères qu'avec KICE. Un autre avantage de GICE est que ses paramètres sont plus faciles à choisir et sont plus expressifs que ceux de KICE.

6.6 Bilan

Dans ce chapitre, nous avons étudié un second niveau de personnalisation, au delà de l'intégration des connaissances dans une mesure d'incompatibilité, en examinant la

question de l'agrégation de cette mesure avec la mesure classique définie comme la pénalité. Nous avons tout d'abord listé les propriétés qu'une fonction d'agrégation doit vérifier pour obtenir l'explication adaptée à l'utilisateur. Les intégrales de Gödel vérifient l'ensemble des propriétés souhaitées pour agréger des fonctions en IA explicable en intégrant de nouvelles contraintes qui décrivent les besoins utilisateur sur les critères étudiés dans la génération de l'explication. En particulier, elles adoptent des comportements différents selon la satisfaction ou non des besoins utilisateur. Nous avons proposé une nouvelle méthode nommée GICE qui résout le problème d'optimisation avec l'intégrale de Gödel basée sur la conjonction. Comme nous l'avons montré à travers des expérimentations, les intégrales de Gödel permettent bien de générer des explications qui modifient principalement les attributs utilisateurs et favorisent les explications qui satisfont les contraintes.

Chapitre 7

Discussion sur la diversité des explications

Dans les chapitres précédents, nous avons étudié deux niveaux de personnalisation qui sont liés : les chapitres 4 et 5 ont proposé une personnalisation qui intègre les connaissances utilisateur par la mesure d'incompatibilité. Ensuite, le chapitre 6 a proposé une personnalisation qui intègre les besoins utilisateur par le choix de l'opérateur d'agrégation.

Ce chapitre discute d'une autre voie pour personnaliser l'explication, basée sur la notion d'explications diverses : celles-ci constituent une façon plus implicite de proposer des explications adaptées à l'utilisateur, en laissant à sa charge une étape de sélection. Proposer plusieurs explications pour comprendre une notion est un principe souvent utilisé en sciences cognitives, notamment dans l'éducation (Ainsworth, 2008) : pour expliquer une même notion à une classe, il est par exemple courant qu'un enseignant utilise différentes explications. Cela lui permet de s'assurer que tous les élèves comprennent en choisissant l'explication la plus adaptée pour eux.

Parmi les méthodes proposées dans cette thèse, la méthode *GICE* peut fournir plusieurs explications contre-factuelles comme illustré sur la figure 6.6. Toutefois, cette multiplicité n'est pas un objectif imposé, elle est obtenue de façon implicite. Dans la littérature, il existe des méthodes qui cherchent explicitement à générer plusieurs explications simultanément, on parle alors d'explications *multiples*. Très souvent, elles imposent en plus qu'elles diffèrent les unes des autres, les explications sont alors *diverses*. Différentes définitions de diversité ont été proposées.

Ce chapitre propose une discussion générale avec un état de l'art, une étude comparative et une structuration des méthodes de la littérature sur la diversité des explications, en se focalisant ensuite sur le cas des explications contre-factuelles. Tout d'abord, nous détaillons les motivations de la proposition d'un ensemble d'explications diverses. Puis, nous présentons la diversité au sein des explications contre-factuelles en réalisant une étude comparative des méthodes de l'état de l'art et en distinguant trois types de diversité. Nous illustrons ensuite ces types de diversité sur des exemples en deux dimensions. Enfin, nous terminons ce chapitre en discutant des enjeux et des perspectives de la génération d'explications multiples.

Ce travail a été présenté dans l'article *Achieving Diversity in Counterfactual Explanations : a Review and Discussion* publié à la conférence ACM Facct 2023 (Laugel et al., 2023).

7.1 Motivations

Dans cette section, nous présentons les motivations générales de ce chapitre. Tout d'abord, nous présentons les risques de la génération d'une unique explication. Puis, nous discutons des besoins de proposer plusieurs explications à l'utilisateur.

7.1.1 Risques encourus par la génération d'explications uniques

Comme discuté dans le chapitre 2, définir les objectifs de l'interprétabilité et les critères numériques pour mesurer la qualité d'une explication est une tâche difficile et complexe. Cependant, la sélection des critères les plus pertinents pour un problème donné n'est qu'une des questions à prendre en compte, une question supplémentaire est leur agrégation. Comme étudié dans le chapitre 6, le choix de l'agrégation a un impact direct sur l'explication générée.

Tout d'abord, dans cette section nous décrivons les méthodes utilisées par les approches existantes pour combiner des critères plus généraux que la pénalité et l'incompatibilité. Puis, nous discutons de l'impact du choix de l'agrégation sur l'explication finale. Enfin, nous présentons les conséquences indésirables qui peuvent être induites.

7.1.1.1 Méthode de combinaison des critères

Les sections 2.7.3 et 2.7.4 ont détaillé des méthodes de la littérature qui utilisent des méthodes de combinaison pour intégrer la connaissance. Dans le chapitre 6, nous avons étudié en détail la combinaison de la pénalité et de l'incompatibilité qui ont deux natures différentes. Dans cette section, nous ne restreignons pas à des critères sémantiquement différents, nous étudions de façon plus générale la combinaison de différents critères. En effet, la mesure de pénalité peut ne pas être réduite à la proximité, mais combiner différents critères parmi ceux par exemple définis dans la section 2.5.4. Dans ce cas, les critères considérés sont tous objectifs, c'est-à-dire techniques, ils ne dépendent pas d'une composante subjective induite par l'utilisateur. Dans cette section, nous étudions la combinaison de ces critères par les méthodes de la littérature, notamment en utilisant des opérateurs classiques présentés dans la section 3.5.

Comme décrit dans la section 3.1.2, il peut être difficile d'optimiser plusieurs critères en même temps, car par définition certains ne peuvent pas être satisfaits ensemble. Par exemple, l'optimisation de la parcimonie d'une explication contre-factuelle est souvent en contradiction avec la maximisation de sa proximité avec l'instance étudiée, comme le montrent Laugel et al., 2018a. Par conséquent, les opérateurs d'agrégation conjonctifs, qui exigent que tous les critères soient satisfaits simultanément, sont rarement utilisés. Reconnaissant cette impossibilité, certaines approches définissent donc une fonction de

coût qui constitue un compromis explicite entre les différents critères. C'est par exemple ainsi que Mahajan et al., 2019, comme présenté dans la section 2.7.3, proposent d'agréger la pénalité de l'explication (mesurée par la distance l_1) et la satisfaction des contraintes causales considérées (distance causale). De même, nous avons étudié la combinaison de la pénalité et de l'incompatibilité avec les connaissances de l'utilisateur.

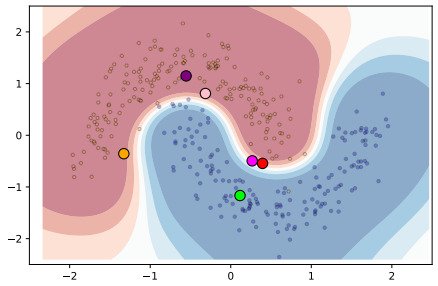
Au lieu de combiner plusieurs critères en un seul objectif, de nombreuses autres approches proposent d'imposer un ordre de priorité entre les critères. Elles considèrent alors une hiérarchie des critères, qui est notamment discutée dans le choix de la fonction d'agrégation dans la section 6.1.4. Ainsi, dans le cas des exemples contre-factuels où la proximité est le critère le moins prioritaire, l'objectif devient la génération de l'exemple contre-factuel le plus proche (c'est-à-dire de la minimisation la fonction de pénalité) dans un sous-espace défini par des contraintes sur d'autres critères. Ce sous-espace d'optimisation peut par exemple être défini par des contraintes sur la densité (Poyiadzi et al., 2020; Artelt and Hammer, 2020), ou l'actionnabilité (Ustun et al., 2019). Ainsi, les méthodes restreignent la recherche aux explications dans une région dense ou actionnable. Inversement, d'autres approches telles que Growing Spheres (Laugel et al., 2018a) ou LORE (Guidotti et al., 2019) optimisent la parcimonie de l'explication contre-factuelle après avoir optimisé sa pénalité.

La définition de cet ordre de priorité peut être liée au processus d'optimisation proposé : un ordre de priorité entre les critères facilite également l'optimisation de la fonction de coût. En effet, le sous-espace satisfaisant la contrainte d'ordre supérieur peut alors être identifié dans une étape de prétraitement. Par exemple, FACE (Poyiadzi et al., 2020) effectue une première étape qui consiste à éliminer dans l'ensemble des instances considérées les explications ne vérifiant pas la contrainte de densité. Puis, parmi ces explications, FACE propose celle minimisant la fonction de coût considérée.

Les méthodes proposées dans cette section permettent d'optimiser les critères sans réellement connaître les besoins des utilisateurs sur ces critères. Dans le chapitre 6, nous nous intéressons à une autre solution qui est l'intégrale de Gödel pour effectuer le bon équilibre entre les deux critères. Cette nouvelle manière d'agréger permet de satisfaire les besoins de l'utilisateur au lieu de vouloir optimiser tous les critères simultanément au risque de n'être satisfaisant selon aucun critère.

7.1.1.2 Impact important de l'opérateur d'agrégation

Bien qu'il semble évident, dans une perspective d'optimisation multicritères, que l'opérateur d'agrégation a un impact direct sur la nature de la solution, à notre connaissance, ce point est rarement abordé dans les approches explicatives. Cela peut paraître surprenant, d'autant plus que certaines approches existantes proposent différentes heuristiques pour les mêmes groupes de critères, et diffèrent donc essentiellement les unes des autres en ce qui concerne la manière de les combiner. Par exemple, Growing Spheres (Laugel et al., 2018a) et LORE (Guidotti et al., 2019) optimisent toutes les deux la proximité de l'explication (mesurée par la distance euclidienne) et sa parcimonie (mesurée par la distance l_0), mais proposent une agrégation différente des deux critères.



	a_0	a_1	a_2	a_3	a_4	a_5	a_6
x	-118.0	33.9	19	6.86	1.02	2374	5.98
e_1	-117.1	35.1	20	5.88	0.89	1953	4.67
e_2	-118.5	33.4	39	2.71	0.98	1257	1.50
e_3	-122.5	37.7	30	4.16	1.18	1577	3.68
e_4	-116.0	34.6	12	6.94	0.63	2344	4.98
e_5	-116.9	38.8	48	5.79	1.29	2099	4.23

FIGURE 7.1 – Illustration de l'impact de l'opérateur d'agrégation : pour l'instance x représentée par le point vert sur la figure de gauche et la première ligne du tableau de droite, cinq exemples contre-factuels différents sont obtenus (définitions décrites dans la section 7.1.1.2) pour différents opérateurs d'agrégation.

Plus généralement, du point de vue de l'optimisation, étant donné deux critères qui ne peuvent pas être optimisés simultanément (comme c'est souvent le cas pour les explications contre-factuelles), l'ensemble des solutions possibles serait le front de Pareto qui amène à une diversité selon les critères considérés, comme le considèrent Dandl et al., 2020.

Exemples illustratifs Nous illustrons l'impact de l'agrégation choisi sur l'explication proposée sur les jeux de données Half-Moons et Californie en générant plusieurs explications contre-factuelles optimisant les mêmes critères avec différents opérateurs d'agrégation. Nous avons proposé dans le chapitre 6, dans la section 6.4.2 une illustration similaire dans le cas où les critères agrégés sont la pénalité et l'incompatibilité.

Nous considérons deux autres critères souhaitables pour les exemples contre-factuels et choisissons ici de prendre en compte : la proximité de l'explication, mesurée par la distance euclidienne, et son appartenance à une région dense de la classe souhaitée, mesurée par la log-vraisemblance de l'exemple contre-factuel estimé par Kernel Density Estimation (KDE) gaussien entraîné sur les données d'entraînement. Ces deux critères sont ensuite combinés à l'aide de plusieurs opérateurs d'agrégation de la littérature : une somme pondérée (e_3, en rose) comme Mahajan et al., 2019 et comme dans les chapitres 4 et 5, la maximisation de la proximité sous contrainte de densité (e_4, en magenta) comme Poyiadzi et al., 2020, la maximisation de la densité sous contrainte de proximité (e_5, en orange) comme Laugel et al., 2018a et Guidotti et al., 2019, ainsi que la pénalité (e_1, en rouge) et la densité (e_2, en violet) indépendamment. Une autre agrégation plus riche sémantiquement et plus expressive qu'on pourrait utiliser est l'intégrale de Gödel comme dans le chapitre 6.

Nous considérons d'abord le jeu de données Half-Moons en deux dimensions, sur lequel un classifieur SVM est entraîné (0.99 de précision). La partie gauche de la figure 7.1 présente les résultats obtenus pour l'instance x représentée par le point vert, dont la prédiction doit être étudiée. On peut observer que les points orange, rouge, rose et violet sont assez dispersés dans l'espace de données. Cela illustre l'importance de l'opérateur d'agrégation pour les explications contre-factuelles.

Nous considérons ensuite le jeu de données Californie présenté dans le tableau 4.2. Nous générons de la même manière les cinq exemples contre-factuels associés à chacune des cinq fonctions d'agrégation présentées précédemment. Les résultats sont présentés dans le tableau de droite de la figure 7.1. Nous remarquons que les explications obtenues sont très variées. Par exemple, les valeurs associées à l'âge des habitations (a_2) varient de 12 à 48 ans, ce qui correspond à un grand intervalle. De même, le nombre moyen de chambres (a_4) varie de 0.63 à 1.29. Pour l'instance étudiée, ce nombre de chambre vaut de 1.02 : certaines explications diminuent ce nombre et d'autres l'augmentent. Cet exemple illustre à quel point utiliser des agrégations différentes implique des modifications différentes pour obtenir la classe souhaitée. En plus d'obtenir différentes explications, nous discutons dans la prochaine section des conséquences non souhaitées des opérateurs d'agrégation choisis.

7.1.1.3 Conséquences indésirables

Nous avons vu que le choix de la fonction d'agrégation influe sur l'explication obtenue, il est possible d'obtenir des résultats très différents. Ce n'est pas la seule question soulevée par ce choix, d'autres questions en termes de pertinence et de besoins de l'utilisateur comme étudié dans le chapitre 6 sont également liées. Par exemple, l'optimisation de critères qui ont des comportements opposés les uns aux autres peut conduire à l'absence totale de solutions ou à des solutions non satisfaisantes. Cela est vrai lorsque les critères concernés appartiennent à des catégories différentes, c'est-à-dire qu'ils sont liés à l'utilisateur, aux données ou à l'instance étudiée. Il semble possible que les utilisateurs, en particulier sans connaissances techniques (qui sont souvent considérés comme des utilisateurs potentiels des méthodes d'IA explicable) ne réalisent pas que l'explication générée ne garantit pas toutes les propriétés souhaitées. Par exemple, dans le cas d'une agrégation réalisée avec une somme pondérée comme pour la méthode proposée par Mahajan et al., 2019 et dans les chapitres 4 et 5, les utilisateurs peuvent ne pas comprendre que la causalité de l'explication est obtenue au détriment d'une plus grande facilité d'action (proximité de l'explication). Plus le nombre de critères est élevé, plus la question de leur agrégation est complexe.

7.1.2 Explications multiples : motivations et discussions additionnelles

Pour résoudre les problèmes d'agrégation discutés dans la section précédente, une solution est de générer *plusieurs* explications et non une seule. Bien qu'une partie des travaux existants proposant multiples exemples contre-factuels exposent certaines motivations pour le faire, ces études sont rarement approfondies. Nous cherchons dans cette section à fournir un argumentaire plus riche en faveur des explications multiples, centré sur deux arguments qui complètent ceux développés précédemment dans la section 7.1.1. Le premier résume des résultats des sciences sociales et cognitives qui ont démontré les avantages considérables de la production d'explications multiples pour

enseigner des concepts complexes. Le second montre que l'existence de plusieurs explications peut aider à surmonter l'une des principales lacunes de l'interprétabilité de l'apprentissage automatique, à savoir satisfaire des besoins des utilisateurs lorsqu'ils ne sont pas formulés explicitement.

7.1.2.1 Sciences sociales : un plus grand nombre d'explications permet une meilleure compréhension

Dans divers domaines scientifiques, proposer plusieurs explications est depuis longtemps considéré comme un facteur clé pour une meilleure compréhension de concepts complexes. Par exemple, dans un contexte médical, Wang et al., 2019 insistent sur le fait que les médecins ont besoin d'explications multiples pour établir de meilleurs diagnostics. Des conclusions similaires sont tirées dans les domaines de l'éducation et de la psychologie : lors de l'utilisation d'analogies pour enseigner des concepts complexes à des étudiants en médecine, il a été démontré que le fait de fournir une seule explication entraîne un risque élevé de fausses interprétations (Spiro et al., 1989), c'est-à-dire une compréhension différente de celle attendue par l'enseignant. Une unique explication présente un seul point de vue qui peut être différent de celui de l'utilisateur, l'explication proposée peut alors être comprise différemment.

De plus, proposer de multiples explications soigneusement sélectionnées est présentée comme une condition nécessaire à une bonne compréhension. De manière plus générale, Miller, 2019 insiste sur le fait que les causes d'un événement doivent être considérées comme multiples et qu'un aspect important de la production d'une bonne explication est la sélection par l'utilisateur de son explication préférée parmi un ensemble d'explications plausibles. L'utilisateur n'est plus passif en acceptant l'explication proposée mais sélectionne l'explication qui lui convient le mieux. Bove et al., 2023 montrent empiriquement les avantages qu'il y a à fournir des explications multiples aux utilisateurs, dans le cadre d'une tâche de classification : ces dernières entraînent une augmentation à la fois de la compréhension objective et de la satisfaction subjective.

7.1.2.2 Satisfaction des besoins non exprimés par l'utilisateur

L'une des principales difficultés identifiées en matière d'interprétabilité en général est la difficulté à déterminer les besoins des utilisateurs. Ainsi, dans le chapitre 6, nous avons fait l'hypothèse que l'utilisateur fournit les seuils δ_P et δ_I sur les critères, mais ces besoins ne sont pas toujours connus et ne se limitent pas aux critères. Une possibilité est de les estimer, mais cela semble difficile étant donné que chaque personne perçoit différemment les interactions et les impacts des attributs (Grgic-Hlaca et al., 2018). Il semble alors illusoire d'espérer qu'une seule explication puisse satisfaire tous les besoins non explicites par les utilisateurs. Générer plusieurs explications et laisser l'utilisateur choisir celle qui lui semble la plus pertinente est un moyen de laisser à l'utilisateur cette tâche "supplémentaire" de compatibilité des explications avec ses besoins et d'écarter les explications qui ne lui conviennent pas (Hilton, 1996; Miller, 2019).

En plus de proposer plusieurs explications, il est nécessaire que ces explications soient diverses, c'est-à-dire qu'elles ne soient pas redondantes, pour couvrir le plus grand nombre de cas de figure. Cette notion de diversité peut être définie de différentes façons, comme nous le discutions dans la section suivante.

7.2 Explications contre-factuelles diverses

Les arguments présentés dans la section 7.1 sont généraux, ils sont valables pour tout type d'explications, nous nous concentrons ici sur les explications contre-factuelles. Les motivations précédentes ont conduit plusieurs approches à expliquer les prédictions par de multiples exemples contre-factuels. La plupart d'entre elles s'appuient sur la notion de *diversité* (voir par exemple Russell, 2019; Mothilal et al., 2020), imposant que les multiples explications diffèrent les unes des autres. Cela permet d'éviter une certaine redondance dans les explications et d'offrir plusieurs alternatives à l'utilisateur.

Comme l'absence de consensus sur les critères définissant les explications, cette notion de diversité a été définie de plusieurs manières. Dans cette section, nous passons en revue ces notions de diversité, en examinant la littérature existante sur les exemples contre-factuels divers. Ces discussions sont résumées dans le tableau 7.1 qui liste les articles que nous avons inclus dans notre étude et les caractérise selon les différents axes que nous proposons ci-dessous. Les trois premières sous-sections détaillent les trois types de diversité que nous proposons de distinguer, respectivement nommés critères, espace des données et actions. Au-delà de la définition de diversité qu'elles considèrent, les méthodes de génération d'explications diverses diffèrent également dans la procédure d'optimisation qu'elles mettent en œuvre. Nous proposons, dans la section 7.2.4, de les distinguer selon trois axes : intégration explicite de la diversité, choix du nombre d'explications ou encore nombre d'exécutions.

7.2.1 Diversité des critères

Un premier type de définition de la diversité dépend des critères de qualité que les exemples contre-factuels doivent optimiser. Le choix de l'agrégation des critères a un impact important sur l'explication finale comme discuté dans la section 7.1.1.2, la diversité des critères consiste à utiliser plusieurs opérateurs d'agrégation au lieu d'en choisir un seul et à générer l'ensemble des explications que chacun induit au lien d'en sélectionner une dans cet ensemble.

Par exemple, Dandl et al., 2020 et Rasouli and Chieh Yu, 2022 s'intéressent à l'optimisation de plusieurs critères comme la proximité ou la densité des explications. Pour générer les explications, ils effectuent différents compromis entre ces critères. Les exemples contre-factuels générés correspondent alors à différentes positions sur le front de Pareto défini par les critères de qualité considérés. La méthode proposée par Rasouli and

Méthode	Diversité		Recherche de contre-factuels		
	Type de diversité	Critère de diversité	Nombre de CF	Explicite	Une exécution
LORE (Guidotti et al., 2019)	Actions	Diverses feuilles d'un arbre de décision	Algo	Oui	Oui
Mahajan (Mahajan et al., 2019)	Données	Stochasticité dans la génération	Utilisateur	Non	Oui
Russell (Russell, 2019)	Actions	Réexécution & exclusion des résultats précédents	Utilisateur + Algo	Oui	Non
CADEX (Moore et al., 2019)	Données	Réexécution & exclusion des résultats précédents	Utilisateur	Non	Non
Ustun (Ustun et al., 2019)	Actions	Réexécution & exclusion des résultats précédents	Utilisateur	Oui	Non
CERTIFAISharma et al., 2020)	Données	Exploration par échantillonnage	Utilisateur	Non	Oui
Tsirtsis (Tsirtsis and Gomez Rodriguez, 2020)	Données	Partitionnement de l'espace de données	Utilisateur	Oui	Oui
MOC (Dandl et al., 2020)	Données & Critères	Front de Pareto dans l'espace des critères	Algo	Oui	Oui
MACE1 (Karimi et al., 2020)	Données	Réexécution & exclusion des résultats précédents	Utilisateur	Oui	Non
DICE (Mothilal et al., 2020)	Données	Mesure de diversité dans l'optimisation	Utilisateur	Oui	Oui
DECE (Cheng et al., 2020)	Données	Considération de différentes contraintes	Utilisateur	Oui	Oui
CRUDS (Downs et al., 2020)	Données	Partitionnement de l'espace de données	Utilisateur	Oui	Oui
DiVE (Rodríguez et al., 2021)	Données	Mesure de diversité dans l'optimisation	Utilisateur + Algo	Oui	Non
OCEANParmentier and Vidal, 2021)	Données	Réexécution & exclusion des résultats précédents	Utilisateur + Algo	Oui	Non
MCCE (Redelmeier et al., 2021)	Données	Exploration par échantillonnage	Utilisateur	Oui	Oui
OrdCEKanamori et al., 2021)	Critères	Front de Pareto dans l'espace des critères	Utilisateur	Oui	Non
MCS (Yang et al., 2021)	Données	Exploration par échantillonnage	Utilisateur	Non	Non
CSCF (Naumann and Ntoutsi, 2021)	Actions	Approche séquentielle pour différentes séquences	Utilisateur	Oui	Oui
MIP-DIVERSE (Mohammadi et al., 2021)	Données	Réexécution & exclusion des résultats précédents	Utilisateur	Oui	Non
Hada (Hada and Carreira-Perpiñán, 2021)	Données	Considération de différentes contraintes	Utilisateur + Algo	Non	Non
Navas (Navas-Palencia, 2021)	Données	Considération de différentes contraintes	Utilisateur	Oui	Oui
Samoilescu (Samoilescu et al., 2021)	Données	Décomposition de l'espace des données	Utilisateur	Oui	Oui
Becker (Becker et al., 2021)	Actions	Diverse feuilles d'un arbre de décision	Utilisateur + Algo	Non	Oui
GeCo (Schleich et al., 2021)	Données	Exploration par échantillonnage	Utilisateur	Non	Oui
FastAR (Verma et al., 2021)	Actions	Stochasticité dans la génération	Utilisateur	Non	Oui
Carreira (Carreira-Perpiñán and Hada, 2021)	Données	Considération de différentes contraintes	Utilisateur	Non	Non
EMC (Yadav et al., 2021)	Critères	Initialisations variées du problème d'optimisation	Utilisateur	Oui	Oui
δ-CLUE (Ley et al., 2022)	Données	Initialisations variées du problème d'optimisation	Utilisateur	Non	Non
CARE1 (Rasouli and Chieh Yu, 2022	Critères	Front de Pareto dans l'espace des critères	Utilisateur	Oui	Oui
MACE2 (Yang et al., 2022)	Données	Exploration par échantillonnage	Utilisateur	Oui	Oui
Smyth (Smyth and Keane, 2022)	Données	Modèle k-NN pour délimiter un groupe de CF	Utilisateur	Oui	Oui
COPA (Bui et al., 2022)	Données	Optimisation utilisant une descente de gradient	Utilisateur	Oui	Oui
FRPD (Nguyen et al., 2023)	Données	Mesure de diversité dans l'optimisation	Utilisateur	Oui	Oui

TABLE 7.1 – Résumé des méthodes existantes de génération d'exemples contre-factuels (CF) divers, examinées dans la section 7.2 qui détaille les colonnes relatives au type de diversité et aux critères.

Chieh Yu, 2022 diffère de celle de Dandl et al., 2020 par l'information supposée disponible sur l'utilisateur : elle considère que l'utilisateur fournit une hiérarchie de l'importance des différents critères, par exemple il préfère avoir une explication proche plutôt que parcimonieuse. Cela permet d'éviter le risque induit par un opérateur de compromis qui peut conduire à une solution qui a en fait une valeur moyenne pour tous les critères considérés. En effet, une hiérarchie définie par l'utilisateur permet de sélectionner les critères à optimiser en premier et ceux à optimiser ultérieurement, un résultat est seulement donné dans le cas où les premiers critères peuvent être optimisés.

Cette diversité des critères peut également être obtenue par les méthodes KICE, KISM et rKICE proposées dans les chapitres 4 et 5. Ces méthodes proposent d'effectuer un compromis en utilisant un paramètre λ. Ainsi, une solution pour générer des explications diverses selon les critères est de fournir des explications associées à différentes valeurs de λ. Un exemple intéressant est notamment montré sur la figure 4.4 où les explications pour différentes valeurs de λ sont présentées dans l'espace des critères. On remarque que les explications ont deux profils différents. Dans le premier cas, il y a une répartition régulière des explications dans l'espace des critères, alors que dans le second cas, les explications ont soit une pénalité et une incompatibilité élevées, soit une pénalité et une incompatibilité faibles.

7.2.2 Diversité dans l'espace des données

Un deuxième type de diversité se concentre sur la position des exemples contre-factuels générés dans l'espace des données. Cette diversité a pour but de proposer des explications, non redondantes, c'est-à-dire qu'elles ne sont pas proches dans l'espace des données. Ceci est illustré dans la figure 7.1 qui montre qu'une même instance peut être associée à différentes explications dans différentes zones de l'espace des données.

Pour obtenir une telle diversité dans l'espace des données, il existe deux types d'approches qui ont été proposées, discutées tour à tour ci-dessous. La première impose des contraintes sur chaque explication individuellement. La seconde optimise directement un ensemble d'exemples contre-factuels en ajoutant une mesure de diversité dans le problème d'optimisation.

Décomposition de l'espace des données Une première approche pour générer des exemples contre-factuels situés dans des zones différentes de l'espace des données consiste à décomposer cet espace en plusieurs sous-espaces et à contraindre la génération d'exemples contre-factuels dans chacun des sous-espaces (Rodríguez et al., 2021; Navas-Palencia, 2021; Carreira-Perpiñán and Hada, 2021). Ainsi, en reprenant l'exemple des appartements, on peut diviser l'espace en deux zones : les appartements de plus de 20 ans et ceux de moins de 20 ans. On propose alors deux exemples contre-factuels différents, l'un associé à un appartement de plus de 20 ans et un autre dans l'autre sous-espace, c'est-à-dire moins de 20 ans.

D'autres méthodes (Russell, 2019; Hada and Carreira-Perpiñán, 2021; Mohammadi et al., 2021) sont basées sur un processus itératif pour générer les explications, où un

exemple contre-factuel est généré à chaque étape. Pour s'assurer que la nouvelle explication est différente des précédentes, elles considèrent une contrainte sur la distance entre la nouvelle explication et les explications déjà générées. Ainsi, la distance entre les exemples n'est pas maximisée mais elle est supérieure à un certain seuil.

Cette diversité dans l'espace des données peut être également obtenue par la méthode rKICE proposée dans la section 5.2. Cette approche considère une règle utilisateur associée à l'instance x, qui définit un sous-espace de l'espace des données. Ainsi, une solution est de considérer un système de règles avec plusieurs règles associées à l'instance étudiée. La méthode rKICE est alors appliquée à plusieurs reprises avec une règle différente, ce qui permet de décomposer l'espace des données en plusieurs sous-espaces.

Mesure de diversité D'autres approches modifient le problème d'optimisation pour produire explicitement un ensemble d'exemples contre-factuels $\{e_1^*, \ldots, e_k^*\}$ en une seule fois, en intégrant un critère de diversité dans la fonction de coût. L'équation 2.2 qui présente un problème d'optimisation qui minimise l'agrégation de plusieurs critères de qualité est modifiée comme suit :

$$\{e_1^*, \ldots, e_k^*\} = \underset{\{e_1,\ldots,e_k\}\subset\mathcal{E}}{\operatorname{argmin}} \; agg\left(\sum_{i=1}^{k} cost_f ct_x(e_k, f, E), div(\{e_1, \ldots, e_k\})\right) \tag{7.1}$$

où k désigne le nombre d'exemples contre-factuels souhaités, $\{e_1, \ldots, e_k\}$ un ensemble d'exemples contre-factuels candidats, div une fonction évaluant leur diversité, considérée comme un nouveau critère de qualité à maximiser et agg un opérateur d'agrégation pour combiner la qualité moyenne des candidats contre-factuels et leur diversité. La fonction de coût peut intégrer différentes composantes, comme la proximité, la parcimonie, la densité ou la compatibilité avec certains des critères supplémentaires discutés dans la section 2.5.4, tels que la parcimonie ou encore la densité.

Par exemple, plusieurs approches définissent la diversité selon les attributs utilisés dans les explications finales (Russell, 2019; Bhatt et al., 2021; Rodríguez et al., 2021), ce qui peut être traduit par la maximisation de la distance l_0 entre les exemples contre-factuels générés. D'autres approches (Mothilal et al., 2020) définissent la diversité comme la distance entre les exemples contre-factuels générés (norme l_1, l_2, ou les deux). Ainsi, les exemples contre-factuels obtenus sont éloignés les uns des autres dans l'espace des données et donc divers dans cet espace. Dans le cas de la classification non binaire, Ley et al., 2022 proposent de prendre en compte, en plus de l'espace des attributs, l'espace des prédictions : ils génèrent des exemples contre-factuels associés à différentes classes autres que celle prédite pour l'instance considérée.

7.2.3 Diversité des actions

Comme décrit dans la section 2.5, une explication contre-factuelle suggère des actions, en tant que modifications de l'instance considérée, qui permettent d'obtenir une

prédiction différente. Un troisième type de diversité vise donc à proposer des explications qui suggèrent des actions différentes. Ces explications sont liées à la diversité en termes d'attributs évoquée ci-dessus, avec une interprétation légèrement différente, davantage liée à la notion de recours algorithmique présentée dans la section 2.5.2 qui consiste à s'intéresser aux actions à effectuer. Outre l'utilisation de la distance l_0, ce type de diversité peut être obtenu dans des contextes plus spécifiques : Guidotti et al., 2019 et Becker et al., 2021 utilisent des arbres de décision pour générer des explications. Le fait d'imposer que ces dernières soient situées dans différentes feuilles de l'arbre implique qu'elles suivent des chemins différents de la racine aux feuilles et, par conséquent, elles sont obtenues en effectuant des étapes différentes.

En plus de proposer des explications qui modifient différents attributs, Russell, 2019 propose des explications qui vont dans des directions différentes, c'est-à-dire que les modifications sont différentes : un premier exemple contre-factuel peut recommander d'augmenter la valeur d'un attribut donné, tandis qu'un autre recommanderait de la diminuer. Les actions induites sont alors complètement différentes. Le même principe s'applique à la proposition de Verma et al., 2021 qui effectue des petites actions successives dans des directions différentes, jusqu'à l'obtention de l'explication finale.

Cette diversité des actions peut être également obtenue par la méthode KICE. Cette méthode considère un ensemble d'attributs selon lesquels les modifications sont favorisées. Une solution pour avoir des explications diverses est de considérer plusieurs ensembles d'attributs, et de générer une explication associée à chacun de ces ensembles. Ainsi, chaque exemple contre-factuel obtenu modifie des attributs.

7.2.4 Caractéristiques de la procédure d'optimisation

Outre le fait qu'ils s'appuient sur différentes définitions de la notion de diversité, qui s'appliquent à différents niveaux, les algorithmes existants pour générer de multiples exemples contre-factuels diffèrent également dans la procédure d'optimisation qu'ils appliquent. Ainsi, la notion de diversité ne se limite à la manière dont elle est représentée mais la procédure utilisée pour l'obtenir est également importante. C'est pourquoi, dans cette section nous étudions de nouvelles dimensions qui caractérisent la procédures d'optimisation. Celles-ci sont résumées dans les trois dernières colonnes du tableau 7.1.

Diversité explicite vs non-explicite La première dimension que nous considérons concerne la prise en compte implicite ou explicite de la notion de diversité. Comme dit précédemment, de nombreuses approches générant des explications contre-factuelles reposent sur des heuristiques, et donc n'expriment pas directement les objectifs optimisés. De la même manière, la notion de diversité est dans certains cas explicitement intégrée comme un objectif, ou bien obtenue via d'autres procédures. Ainsi, nous proposons de distinguer ces deux types d'approches.

Dans un premier temps, nous étudions les méthodes qui n'intègrent pas la diversité de manière implicite : certaines approches sont non déterministes, leur application répétée peut conduire à plusieurs explications. Il faut noter qu'elles peuvent alors conduire à des explications multiples, sans qu'elles soient nécessairement diverses : il y a même une très forte probabilité qu'elles soient similaires, par exemple qu'elles soient proches dans l'espace des données, car la robustesse de ces méthodes a pour but de proposer une explication similaire même avec une légère modification en entrée. D'autres, comme Parmentier and Vidal, 2021 et Russell, 2019, proposent des approches déterministes qui excluent l'explication générée pour en obtenir une nouvelle. La nouvelle explication sera différente de la première mais elle peut être très proche.

Dans un second temps, nous nous concentrons sur les méthodes qui considèrent la diversité de manière explicite : elle peut être incluse en tant que critère de qualité directement dans la fonction de coût modifié comme présenté dans l'équation (7.1) (Mothilal et al., 2020; Dandl et al., 2020). Le problème d'optimisation prend alors une forme différente, son résultat est un ensemble d'explications et non une explication unique. Mothilal et al., 2020 définissent la diversité d'un ensemble de solutions comme la distance moyenne entre les paires d'exemple contre-factuels. Ainsi, le but est d'obtenir des explications distantes dans l'espace des données. Une seconde possibilité consiste à étudier différents problèmes d'optimisation comme Carreira-Perpiñán and Hada, 2021; Hada and Carreira-Perpiñán, 2021, notamment en intégrant différentes contraintes sur les attributs ou sur l'espace des données, ce qui permet d'obtenir des explications qui répondent à différents contextes ou motivations. Dans ce cas, pour chaque problème d'optimisation, l'information considérée comme entrée n'est pas la même, elle est associée à une certaine contrainte et influencera la diversité des explications. Une troisième solution proposée par Dandl et al., 2020 intègre la diversité par le biais de plusieurs fonctions d'agrégation pour combiner les critères de qualité considérés. Les explications obtenues sont alors des solutions de cette agrégation, elles sont réparties régulièrement dans l'espace des critères.

Nombre d'exemples contre-factuels générés Une seconde dimension discute de la manière dont le nombre d'explications finales est déterminé : l'utilisateur peut effectuer ce choix ou il est fixé par l'algorithme. Il est souvent déterminé par l'utilisateur car les méthodes laissent la liberté à celui-ci de choisir le nombre d'explications qu'il souhaite. Or le nombre d'explications a un impact sur la possibilité d'avoir ou non des explications diverses. Lorsque ce nombre est trop élevé, les explications proposées ont de forte chance d'être redondantes. Par exemple, si on considère le jeu de données Half-Moons en deux dimensions et qu'on souhaite 10 explications diverses dans l'espace des données, ces explications risquent d'être proches : localement autour de l'instance étudiée il est difficile d'avoir 10 exemples éloignés. Ainsi, il y a un compromis à effectuer entre le nombre d'exemples contre-factuels et la diversité souhaitée entre ces exemples. C'est pourquoi dans certains cas, le nombre d'explications peut être limité par la méthode elle-même.

Ce nombre d'explications générées peut également être imposé par la méthode utilisée. Par exemple, pour les méthodes proposant des explications dans différentes feuilles d'un arbre, comme LORE (Guidotti et al., 2019) ou Becker et al., 2021, le nombre d'exemples contre-factuels à générer est limité par le nombre de feuilles de l'arbre.

Une vs. plusieurs exécutions Pour générer des explications multiples, il existe deux manières de faire : toutes les explications sont générées en même temps ou plusieurs étapes sont effectuées pour générer itérativement des explications supplémentaires. Les méthodes générant plusieurs explications en une seule étape (Mothilal et al., 2020) optimisent souvent une fonction de coût s'appliquant à des ensembles de candidats, comme indiqué dans l'équation (7.1) (Tsirtsis and Gomez Rodriguez, 2020). Ce cas fait référence aux méthodes qui intègrent la diversité de manière explicite dans la fonction de coût. Ainsi, toutes ces méthodes effectuent une seule exécution. Une autre procédure consiste à explorer simultanément l'espace des données dans différentes directions, comme la méthode LORE (Guidotti et al., 2019) qui étudie plusieurs branches d'un arbre en même temps. Cette procédure est notamment utilisée dans le cas de la diversité selon les actions.

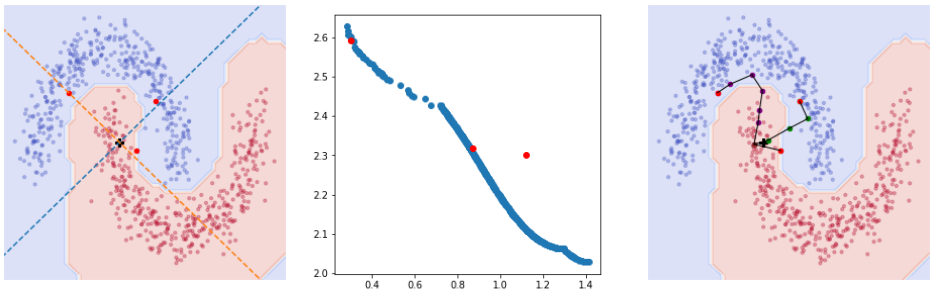
D'autres approches (Samoilescu et al., 2021; Carreira-Perpiñán and Hada, 2021; Ley et al., 2022) génèrent des explications de manière itérative, en utilisant les explications obtenues lors des étapes précédentes pour générer la nouvelle explication. En particulier, pour garantir la diversité des explications, les approches recherchent à chaque étape une explication qui n'est pas similaire aux précédentes. D'autres approches considèrent différentes contraintes à chaque étape pour répondre à des contextes variés. Le problème d'optimisation étudié, parfois défini implicitement, est alors différent pour chaque exemple contre-factuel.

7.3 Illustrations expérimentales

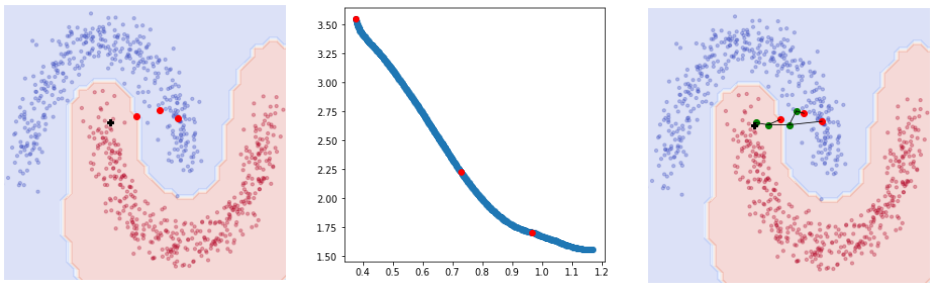
Nous avons présenté dans la section précédente les différentes définitions de la diversité pour les explications contre-factuelles et discuté de l'intégration de cette notion par les méthodes de la littérature. Ici, nous illustrons et discutons des différences entre les trois familles de diversité. Elles sont résumées graphiquement sur la figure 7.2, commentée dans les sections suivantes après la présentation du protocole expérimental adopté.

7.3.1 Protocole

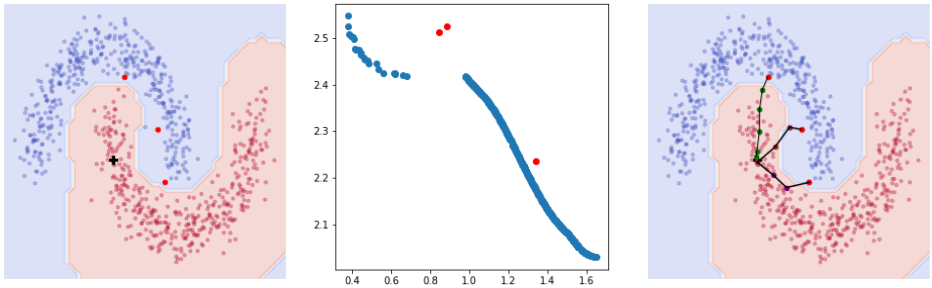
Nous considérons le jeu de données Half-Moons sur lequel nous entraînons un classifieur SVM avec un noyau gaussien. Nous étudions des explications qui optimisent deux critères : la proximité avec la distance euclidienne et la densité mesurée par la log-vraisemblance de l'exemple contre-factuel estimé par Kernel Density Estimation



(a) Diversité dans l'espace des données.



(b) Diversité selon les critères.



(c) Diversité selon les actions : trois chemins pour accéder aux explications : en vert, en violet et en marron.

FIGURE 7.2 – Explications contre-factuelles diverses selon les trois familles décrites dans la section 7.2 : diversité dans l'espace des données, des critères et des actions, chaque ligne correspond à une instance de référence différente. A gauche : représentation dans l'espace des données, au centre : représentation dans l'espace des critères, à droite : représentation dans l'espace des actions.

(KDE) gaussien entraîné sur les données d'entraînement. Pour effectuer une comparaison, nous présentons, pour trois instances différentes, trois explications contre-factuelles diverses, en considérant les trois types de diversité discutés dans la section précédente (espace des données, des critères et des actions). On génère ainsi trois ensembles d'exemples contre-factuels pour chaque instance.

Le premier ensemble associé à la diversité dans l'espace des données est obtenu en divisant l'espace des données en quatre sous-espaces représentés sur la figure gauche de la figure 7.2a. Nous considérons pour ce cas l'instance $x = (-0.2, 0)$, ainsi les droites en bleu et en orange correspondent respectivement aux fonctions $y = x + 0.2$ et $y = -x + 1.6$. Les explications diverses dans l'espace des critères quant à elles sont obtenues en utilisant la méthode de Dandl et al., 2020 qui se base sur l'algorithme NSGA-2 qui génère

des explications à espace régulier sur le front de Pareto. Enfin, pour les explications diverses selon les actions, nous nous basons sur la proposition de Poyiadzi et al., 2020 pour définir les actions. Ainsi, les actions pour obtenir un exemple contre-factuel sont définies par un chemin qui passe par des points proches les uns des autres et qui sont dans une région dense. Pour observer des résultats intéressants, nous considérons une contrainte forte de densité sur l'exemple finale et une contrainte faible sur les instances du chemin.

Les résultats sont présentés sur trois graphiques. Le premier, dans la colonne de gauche de la figure 7.2, représente l'espace des données. Les classes prédites par le classifieur à expliquer sont indiquées en bleu et en rouge. Les points bleus et rouges représentent les données d'entraînement, la croix l'instance étudiée et les points rouges représentent les exemples contre-factuels obtenus. Le second graphique, au milieu, représente l'espace des critères : la pénalité en abscisse et la densité en ordonnée. Sur ce graphique, les points du front de Pareto sont représentés en bleu et les exemples contre-factuels en vert. Enfin, la dernière figure décrit les actions qui peuvent amener aux exemples contre-factuels. Ces actions sont vues ici comme des chemins par lesquels l'utilisateur doit passer pour avoir l'explication.

7.3.2 Analyse des résultats

Diversité dans l'espace des données Nous considérons la diversité dans l'espace des données illustrée sur les graphiques en haut de la figure 7.2. Sur la figure de gauche, on remarque que les explications se situent dans trois sous-espaces différents parmi les quatre sous-espaces considérés. De plus, les explications obtenues sont éloignées les unes des autres.

Sur la figure du milieu, on observe que deux explications parmi les trois se situent sur le front de Pareto. La troisième n'y est pas car il n'y a aucune explication du front de Pareto dans la zone de l'espace de données associée. Les contraintes de diversité conduisent donc à une solution dominée dans l'espace des critères (proximité, densité). En revanche, l'ensemble des trois exemples n'est pas dominé dans l'espace (proximité moyenne, densité moyenne, diversité). On remarque également que deux explications sont très proches dans l'espace des critères étant donné qu'elles sont dans des régions de même densité, les explications ne sont pas diverses dans l'espace des critères.

Sur la figure de droite, on remarque que les chemins proposés pour atteindre les trois exemples sont différents. Ainsi, dans ce cas la diversité dans l'espace des données implique une diversité des actions.

Diversité dans l'espace des critères Les graphiques du milieu de la figure 7.2 montrent les explications générées en tenant compte d'une diversité dans l'espace des critères. Nous remarquons au centre que l'ensemble des points est en effet dispersé sur le front de Pareto. Par contre, ces exemples sont très proches dans l'espace des données, ils se situent tous à droite de l'instance étudiée. De même, on observe sur la figure de droite que les chemins utilisés pour y accéder sont dans la même direction, plusieurs d'entre

eux passent par des points similaires pour accéder aux explications. Pour cet exemple, on remarque que la diversité des critères n'implique pas une diversité dans l'espace des données ou une diversité des actions.

Diversité dans l'espace des actions Enfin, la troisième ligne illustre les exemples contre-factuels générés en tenant compte de la contrainte de diversité selon les actions. Nous remarquons sur la figure de droite que les chemins utilisés pour obtenir les explications sont différents, ils ne passent pas par des points communs. Sur la figure de gauche, les explications ne sont pas proches dans l'espace de données mais elles se situent toutes à droite de l'instance étudiée. On peut remarquer tout de même que la diversité des actions implique une certaine diversité dans l'espace des données. Sur la figure du milieu, aucun des points ne se situent sur le front de Pareto, c'est notamment dû à la façon dont les explications sont obtenues. De plus, deux explications sont très proches dans l'espace des critères car elles sont proches de l'instance étudiée et dans des régions peu denses. On remarque que la diversité des actions n'implique pas une diversité des critères.

Ces exemples montrent les différents ensembles d'explications diverses qu'il est possible d'obtenir. De plus, on remarque qu'un ensemble d'explications obtenues selon un type de diversité ne vérifie pas toujours un autre type de diversité. Selon le contexte étudié et les attentes de l'utilisateur, il est important de choisir le type de diversité le plus adapté. Lorsque le contexte étudié est inconnu, il serait intéressant de combiner les trois types de diversité afin de proposer des explications diverses dans chacun des espaces.

7.4 Discussion et enjeux

La richesse des approches générant des exemples contre-factuels divers présentées dans la section précédente souligne une fois de plus l'importance de motiver le choix des objectifs dans l'explicabilité, y compris la diversité. Comme montré dans la section précédente les résultats peuvent différer grandement d'une définition à l'autre, soulevant la question de la diversité répondant aux besoins utilisateur. Dans cette section, nous proposons une discussion sur le lien entre la notion de diversité et la façon dont la diversité peut aider à mieux répondre aux besoins des utilisateurs.

7.4.1 La diversité comme moyen de répondre aux besoins inconnus des utilisateurs

La section 7.1.2 rappelle l'un des arguments les plus forts en faveur de l'utilisation d'explications contre-factuelles multiples : leur capacité à satisfaire les besoins non fournis directement par les utilisateurs. Une hypothèse implicite à cet effet est que les explications doivent être diverses (Sullivan and Verreault-Julien, 2022) selon différents types de besoins ce qui conduit à des définitions concurrentes de la diversité, décrites dans la précédente section.

Besoins sur les critères La diversité des critères décrit dans la section 7.2.1, répond directement à cet objectif : en proposant de multiples façons de combiner différents critères de qualité, elle permet à l'utilisateur de choisir son ordre de préférence entre les propriétés des explications. Par exemple, demander à un utilisateur de spécifier le niveau minimum de densité de l'explication qu'il souhaite pour un problème peut s'avérer difficile pour lui. Proposer des explications avec différents niveaux de densité et de pénalité pourrait l'aider à comprendre le compromis entre ces deux notions dans la prédiction expliquée et à sélectionner l'agrégation qu'il préfère.

Toutefois, cela suppose que l'utilisateur soit en mesure de comprendre les critères considérés, du moins le fait que les diverses explications proposées varient en fonction de ces critères. Cela remet donc en question l'utilisation de la notion de diversité dans les critères associés aux données (densité locale ou proximité), cette notion est facile à comprendre pour un utilisateur car la notion de besoin sur les critères est une notion intuitive comme discuté dans le chapitre 6. Par contre, l'impact de cette diversité n'est explicite sur les explications contre-factuelles finales. Elles peuvent avoir des valeurs similaires mais être diverses selon les critères.

Les critères associés aux utilisateurs (cf. section 2.7) comme l'actionnabilité, la causalité ou encore la personnalisation, en revanche, ne souffrent pas de ce problème. Par nature, bien que la quantification numérique de ces critères puisse être contestée, on s'attend à ce que l'utilisateur comprenne directement les différences, par exemple qu'il comprenne la différence entre une explication qui est proche et une qui est éloignée. Cela nous amène à penser que pour être pertinente, la diversité d'un ensemble d'explications contre-factuelles doit être observable. Cela remet naturellement en question l'utilité des approches intégrant une diversité non explicite, car elles ne garantissent pas la résolution du problème.

Besoins sur les actions En proposant des explications qui fournissent un ensemble d'actions à modifier pour obtenir la classe souhaitée, les approches intégrant la diversité dans les actions remplissent cet objectif de différences observables. L'utilisateur se voit proposer plusieurs recours alternatifs, parmi lesquels il peut choisir celui qu'il préfère, en fonction de ses préférences personnelles non explicitées. En revanche, la diversité dans l'espace des données ne répond pas explicitement à un besoin formulé par l'utilisateur. Bien qu'elles puissent constituer une approximation de la diversité des actions lorsque les critères de contextualisation de l'utilisateur ne sont pas disponibles, elles ne semblent pas répondre directement aux attentes des utilisateurs, car la diversité dans l'espace des données se focalise plus sur l'exemple final plutôt que l'ensemble des modifications que l'utilisateur doit effectuer. Pourtant, elles constituent le type d'approche le plus représenté (voir tableau 7.1).

7.4.2 La diversité des formes d'explications

Dans la section précédente, nous avons étudié sur la diversité des exemples contre-factuels, mais la diversité peut également considérer d'autres formes d'explications. En

raison de la formulation habituelle des explications contre-factuelles, la pénalité est généralement considérée comme un critère associé à un type d'explication qu'on cherche à minimiser. En la combinant à d'autres critères, il est éventuellement possible de la sacrifier un peu pour satisfaire d'autres critères. Peu de travaux explorent la possibilité de combiner explicitement des explications contre-factuelles locales avec des explications plus globales (un concept proposé par exemple par Rawal and Lakkaraju, 2020). Pourtant, plusieurs travaux ont mis en évidence, dans des contextes appliqués, les avantages de la combinaison d'explications locales et globales sur l'interprétabilité. Ainsi, Collaris et al., 2018 combinent les vecteurs d'importance des attributs locaux et globaux pour expliquer les modèles de détection de fraude ; dans le cadre de la tarification de l'assurance, Bove et al., 2022 proposent des explications locales avec une contextualisation globale. Comme l'illustrent ces travaux, cette perspective semble fortement liée à une autre notion de diversité très intéressante qu'on pourrait appeler diversité des formes d'explication.

7.5 Bilan

Dans le cadre de la thèse, nous nous intéressons à la personnalisation des explications. La complexité de cette tâche peut en partie être résolue par la génération d'explications diverses qui laisse la liberté à l'utilisateur de choisir l'explication souhaitée. Ce chapitre s'est intéressé à la génération de plusieurs explications et particulièrement des explications diverses. Nous avons montré que la combinaison des critères de qualité pour générer des explications contre-factuelles n'est pas une étude triviale. La génération d'une unique explication peut être risquée car elle peut ne pas être adaptée à l'utilisateur, une solution est de proposer des explications diverses. Nous avons proposé de distinguer les approches de la littérature selon différents types de diversité : diversité dans l'espace des données, diversité des critères et diversité des actions. Ces diversités permettent de répondre à différents besoins de l'utilisateur.

Chapitre 8

Conclusion et perspectives

8.1 Conclusion

Dans cette thèse, nous avons étudié la génération d'explications pour enrichir les prédictions faites par des classifieurs dans le cadre de l'IA explicable, en considérant le cas d'explications post-hoc personnalisées. Nous avons étudié différents types d'explications et de personnalisation selon les informations disponibles, nos propositions couvrent trois niveaux de personnalisation. Le premier considère qu'une information sur les utilisateurs, particulièrement sur les connaissances de l'utilisateur, est fournie, donnant des renseignements sur qu'il sait déjà. Le second considère une information différente, qui porte sur les besoins de l'utilisateur, qui induisent des préférences sur les explications possibles. Le troisième cas vise à offrir des possibilités de personnalisation en l'absence d'informations sur l'utilisateur, par le biais de la génération d'explications diverses.

Intégration de connaissances utilisateur Nous avons considéré une information capable de représenter quelque chose pour l'utilisateur, et nous avons proposé une explication dans son langage, c'est-à-dire qui s'aligne sur ce qu'il connaît. L'intégration de connaissances est principalement étudiée dans les chapitres 3 à 5. Pour cette intégration, nous avons proposé un nouveau critère que nous avons appelé incompatibilité, qui mesure à quel point une explication candidate est compatible avec les connaissances considérées.

Dans le chapitre 4, nous avons défini ce critère dans le cadre des explications contrefactuelles avec des connaissances exprimées sous forme d'un ensemble d'attributs. Une explication est alors dite incompatible si elle modifie les attributs que l'utilisateur ne connaît pas, les modifications selon les attributs connus sont ainsi favorisées. Pour l'obtenir, nous avons proposé un nouvel algorithme nommé *Knowledge Integration in Counterfactual Explanation* (KICE).

Dans le chapitre 5, nous avons considéré un autre type d'explications, les vecteurs d'importance des attributs, et un autre type de connaissances, les systèmes de règles. Dans le cas des vecteurs d'importance des attributs, nous avons proposé une définition adaptée d'incompatibilité pour un ensemble d'attributs. Une explication est alors dite incompatible si le vecteur associe une grande importance aux attributs que l'utilisateur

ne connaît pas. Nous avons également examiné une autre forme de connaissances où l'utilisateur possède une connaissance plus riche que les attributs, qui renseigne sur son propre processus de classification sous forme de règles. Une explication incompatible est une explication qui est n'est pas en accord avec la règle de l'utilisateur. Nous proposons deux méthodes *Knowledge Integration in Surrogate Models* (KISM) et *Rule Knowledge Integration in Counterfactual Explanation* (rKICE) qui génèrent les deux explications précédentes. Nous avons mené des expérimentations sur des données de référence pour montrer les propriétés et la pertinence de ces algorithmes.

Intégration de besoins utilisateur Un autre type d'informations que l'utilisateur peut donner concerne les besoins qui induisent ses préférences sur les critères. Nous avons étudié leur intégration dans le problème d'optimisation par le biais de l'agrégation des critères de pénalité et d'incompatibilité. Les chapitres 4 et 5, qui ne disposent pas d'information sur les besoins utilisateur, proposent d'effectuer un compromis entre ces critères par une somme pondérée. Le chapitre 6 étudie plus en détail la question de l'agrégation en intégrant les besoins utilisateur sur les deux critères.

Nous avons proposé d'utiliser l'intégrale de Gödel pour agréger les critères de manière à répondre à un ensemble de propriétés souhaitées dans le domaine de l'IA explicable. Cette agrégation propose un comportement différent selon la satisfaction ou non des besoins utilisateur. Nous avons proposé une nouvelle méthode nommée *Gödel Integrals for Counterfactual Explanation* GICE qui résout le problème d'optimisation associé à l'intégrale de Gödel. Nous avons mené des expérimentations sur les données de référence pour montrer la richesse et l'expressivité de l'intégrale de Gödel dans ce contexte.

Génération d'explications diverses A un troisième niveau, nous avons étudié la personnalisation sans informations sur l'utilisateur. Pour cela, nous avons proposé d'utiliser le cadre de génération d'explications diverses : l'objectif est d'exploiter la multiplicité des explications pour couvrir différentes configurations tout en garantissant la non-redondance par le biais de la diversité. En particulier, nous avons étudié la diversité des explications contre-factuelles en présentant d'abord les différents risques de générer une unique explication. Nous avons effectué une étude comparative des méthodes générant des explications contre-factuelles diverses en les caractérisant selon trois familles de diversité : des données, des critères et des actions. Nous avons présenté une expérimentation illustrant la richesse de chacune des diversités et les différences entre elles.

8.2 Perspectives

Les contributions de cette thèse ouvrent des perspectives variées. Nous les organisons autour de trois axes qui concernent respectivement l'agrégation de critères, la collecte des informations utilisateur et l'évaluation des explications.

8.2.1 Agrégation des critères

Un premier axe porte sur la problématique d'agrégation que nous avons en particulier étudiée dans le chapitre 6, nous présentons trois perspectives associées : la première consiste à étudier d'autres variantes de l'intégrale de Sugeno, un second travail a pour objectif de renforcer une des propriétés souhaitées, la monotonie de l'opérateur en ses deux arguments. La troisième perspective porte sur l'étude de la combinaison de différentes diversités.

Intégrales de Sugeno Dans cette thèse, nous avons étudié la combinaison des deux critères, pénalité et incompatibilité. En particulier, dans le chapitre 6, nous avons utilisé les intégrales de Gödel qui sont des variantes de l'intégrale de Sugeno. Une étude intéressante vise à généraliser les études que nous avons menées aux intégrales de Sugeno qui semblent vérifier les propriétés souhaitées. Ainsi, une perspective est d'effectuer une étude comparative d'autres variantes de l'intégrale de Sugeno, comme l'intégrale de Shilkret, 1971 ou l'intégrale de Dvořák and Holčapek, 2012, et des explications associées pour enrichir les propriétés souhaitées.

Une première étape est de vérifier que les intégrales de Sugeno satisfont les quatre propriétés que nous avons discutées comme étant souhaitables pour combiner la pénalité et l'incompatibilité. Notamment, il est intéressant d'observer de quelle manière la troisième propriété sur le comportement des critères selon différentes zones est considérée par les différentes variantes. Plusieurs interrogations sont alors soulevées : le nombre de zones obtenues, leurs emplacements et la fonction d'agrégation associée à chaque zone. Les zones obtenues sont liées à la définition des besoins utilisateur. Les fonctions quant à elles sont liées au contexte lors duquel cette variante de l'intégrale va être utilisée. La deuxième étape est la mise en œuvre de cette étude théorique et son application dans le contexte de l'IA explicable, pour proposer un algorithme de génération d'explications et analyser expérimentalement les résultats qu'il permet d'obtenir.

Monotonie stricte Dans le chapitre 6, nous avons discuté de quatre propriétés nécessaires pour l'agrégation de la pénalité et l'incompatibilité. Nous avons proposé d'utiliser l'intégrale de Gödel qui vérifie ces propriétés. Nous avons observé que plusieurs explications peuvent être générées, elles diffèrent par la valeur de pénalité ou d'incompatibilité, mais pas les deux : elles ont la même valeur que d'autres explications pour l'un des deux critères. Elles sont donc dominées par d'autres explications selon un des critères considérés, cela est dû au fait que l'intégrale de Gödel n'est pas strictement monotone. Il serait intéressant de considérer un opérateur qui présente une monotonie stricte en plus des quatre propriétés souhaitées.

A notre connaissance, il n'existe pas d'opérateur qui corresponde à ce cas de figure, c'est pourquoi nous avons proposé d'utiliser l'intégrale de Gödel qui vérifie une monotonie non stricte. Pour avoir une monotonie globale stricte, il est nécessaire que les opérateurs associés aux quatre zones de l'espace des critères soient strictement monotones, aussi les opérateurs considérés par l'intégrale de Gödel ($\max(P, I)$, $\min(P, I)$, P et I)

ne peuvent pas être utilisés. Une piste est de proposer un nouvel opérateur d'agrégation qui soit une version modifiée de l'intégrale de Gödel. Plusieurs questions se posent alors. Tout d'abord, il faut se demander à quel niveau la modification peut être effectuée : une première piste est de choisir un opérateur autre qu'une conjonction de Gödel, une seconde piste est de considérer une autre forme que min-max de l'intégrale de Sugeno et une troisième piste utilise une autre famille d'opérateurs. Il faudra alors étudier si les sémantiques proposées par l'opérateur dans chacune des zones correspondent bien au comportement souhaité. Après avoir fait une étude locale de l'opérateur, il sera nécessaire de vérifier que le comportement global satisfait bien les propriétés souhaitées. Une perspective à plus long terme est d'étudier l'opérateur proposé de façon plus générale que pour la pénalité et l'incompatibilité, hors du cadre de l'IA explicable.

Combinaison des diversités Contrairement aux premiers chapitres, le chapitre 7 ne se concentre pas sur l'étude de la pénalité et de l'incompatibilité, il étudie la notion de diversité. Nous avons présenté dans ce chapitre trois types différents de diversité.

Une suite intéressante à ces travaux est la combinaison de ces trois types de diversité. La section 7.3 a montré qu'un type de diversité n'implique pas toujours un autre type de diversité et la majorité des méthodes ne considèrent qu'un seul type de diversité. Seuls Dandl et al., 2020 proposent de combiner la diversité dans l'espace des critères et des données. Il serait intéressant de proposer une méthode de génération d'explications qui fournisse un ensemble d'exemples contre-factuels qui soient divers simultanément dans l'espace des données, des critères et des actions.

Il existe plusieurs pistes intéressantes pour aborder cette perspective. Une première piste est la combinaison de ces diversités par un opérateur d'agrégation. Comme discuté dans le chapitre 6 et la section 2.5.4, choisir un opérateur d'agrégation n'est pas simple, une étude sur les propriétés à vérifier ou la sémantique à avoir est nécessaire. Une seconde piste est la priorisation d'une diversité par rapport à une autre, par exemple sélectionner les explications diverses dans l'espace des critères parmi celles qui sont diverses dans l'espace des données. Cette piste soulève la question de l'ordre de priorisation. Une question supplémentaire à se poser est de savoir si la combinaison des critères implique une perte d'informations sur les besoins ou permet bien de couvrir tous les besoins.

8.2.2 Collecte des informations utilisateur

Dans notre thèse, nous avons étudié l'intégration d'informations supplémentaires comme les connaissances dans les chapitres 4, 5 et les besoins dans le chapitre 6. Nous avons considéré que ces informations sont fournies directement par l'utilisateur dans la forme souhaitée. Une piste intéressante porte sur la collecte de ces informations, que ce soit directement ou par apprentissage à partir de données.

Collecte des connaissances De nombreuses études (Chklovski and Gil, 2005; Blythe et al., 2001) étudient la collecte de connaissances à travers des interfaces utilisateur. Cette

tâche n'est pas simple, elle soulève de nombreuses questions comme la cohérence des réponses utilisateurs, la compréhensibilité des informations extraites, la clarté des actions à effectuer ou encore la vérification de la complétude de la connaissance. Deux types d'outils de collecte d'informations peuvent être distingués : ceux qui proposent une interface complète qui demande de façon explicite l'information à l'utilisateur et ceux qui extraient la connaissance en utilisant des outils d'IA. Dans les deux cas, la procédure utilisée dépend de la forme de connaissances souhaitée.

Nous présentons ici des pistes pour la collecte des deux types de connaissances étudiés dans cette thèse, un ensemble d'attributs interprétés comme les attributs connus par l'utilisateur et un système de règles. Une manière simple de collecter un ensemble d'attributs est de présenter à l'utilisateur tous les attributs considérés par le modèle et de lui demander de sélectionner ceux qui font sens pour lui. Cette approche, simple, peut être fastidieuse pour l'utilisateur : une limitation sur le nombre d'attributs présentés est nécessaire, de même que le développement d'une interface adaptée. L'utilisateur peut se baser sur différents critères pour effectuer son choix : les attributs retenus peuvent être ceux qu'il comprend, ceux qu'il est capable de modifier ou ceux dont il pense qu'ils ont un impact sur la prédiction. Il est alors nécessaire d'exprimer de façon claire la sémantique souhaitée pour s'assurer que les informations exprimées par les utilisateurs répondent à la même question.

Dans le chapitre 5, nous avons considéré des connaissances sous forme de systèmes de règles. Une manière de collecter ces règles est de présenter successivement plusieurs instances à l'utilisateur et de lui demander quelle classe il associerait à chacune d'entre elle, précisant les raisons à l'aide de conditions. Une sélection dynamique des instances présentées peut être envisagée afin de minimiser la tâche de l'utilisateur en garantissant la couverture des espaces des données. Il faut noter qu'il est important de vérifier la cohérence des résultats, c'est-à-dire par exemple que l'utilisateur n'associe pas deux classes différentes à la même instance.

Élicitation des préférences utilisateur Au delà de la collecte des connaissances utilisateur, il y a la question des besoins utilisateur sur les critères considérés. Ces besoins définissent des seuils de tolérance pour les critères qu'on souhaite optimiser. Nos travaux de thèse ont porté sur la prise en compte de ces besoins dans la génération des explications. Une perspective de recherche à ses travaux porte sur leur acquisition et leur collecte comme pour les connaissances utilisateur. L'expression de ces besoins par l'utilisateur est une tâche potentiellement complexe, notamment s'il ne connaît pas le domaine ou les critères. Nous proposons plusieurs pistes pour aborder cette tâche.

Tout d'abord, les besoins peuvent être définis de manière qualitative et non quantitative en utilisant des valeurs linguistiques et non des valeurs numériques, par exemple la pénalité doit être faible, moyenne ou élevée. Cela peut aider l'utilisateur à définir plus facilement les besoins même s'il n'a pas de connaissances techniques.

Une deuxième piste est d'automatiser l'extraction des besoins par l'apprentissage automatique. Pour cela, une solution peut être de demander à l'utilisateur d'évaluer des

exemples contre-factuels deux à deux distincts dans le but de faire une approximation des seuils associés aux contraintes (6.1) et (6.2), par exemple représenté sous la forme d'un intervalle de valeurs acceptées (initialisé à $[0, 1]$). L'évaluation par l'utilisateur à chaque étape de deux explications a pour but d'avoir une information supplémentaire sur les seuils. Cette information peut aider à réduire l'intervalle associé aux seuils jusqu'à ce qu'une unique valeur soit obtenue. Une question importante de cette étude sera d'identifier les explications soumises à l'utilisateur à chaque étape pour obtenir le plus rapidement possible la valeur du seuil.

Diversité selon les informations utilisateurs Nous avons étudié deux cas distincts : celui où les informations utilisateur sont fournies directement comme dans les chapitres 4, 5 et 6 et celui où aucune information n'est donnée comme dans le chapitre 7. Dans ce dernier cas, des explications diverses sont proposées dans le but de couvrir tous les types d'utilisateurs. Par exemple, la diversité des actions permet de proposer des explications associées à des connaissances différentes. La diversité des critères quant à elle propose des explications associées à des besoins utilisateur différents. Ainsi, ces diversités distinguent des profils d'utilisateurs caractérisés par leurs connaissances ou leurs besoins sur les critères et proposent une explication associée à chacune de ces informations.

Il serait intéressant de définir d'autres profils d'utilisateurs basés sur des caractéristiques autres que la connaissance ou les besoins. Une première question est de savoir quelles autres informations sur les utilisateurs existent. Une seconde question est de savoir comment les extraire. Il existe de nombreuses études (Sipos et al., 2023; Liao et al., 2020) qui proposent des banques de questions permettant de collecter les informations utilisateur. Cependant, ces études se focalisent sur les questions quoi ? et pourquoi ? présentées dans la section 2.1.1, c'est-à-dire qu'est ce qu'on souhaite expliquer et les motivations de l'utilisateur, et très peu sur la question qui ?. Ainsi les questions ne cherchent pas à extraire des informations sur l'utilisateur. Il serait intéressant d'effectuer une étude similaire sur les caractéristiques de l'utilisateur.

Une piste pour cette perspective est de constituer une banque de questions qui permette d'exprimer toutes les informations utiles pour proposer une explication adaptée. A partir des réponses à ces questions il serait intéressant regrouper les utilisateurs ayant des résultats similaires pour établir des profils d'utilisateurs. Une possibilité est alors de définir une nouvelle diversité qui propose une explication associée à chaque groupe d'utilisateurs créé.

8.2.3 Évaluation des explications : expérimentations avec des utilisateurs réels

Dans cette thèse, nous avons étudié différentes méthodes générant des explications pour un utilisateur et nous avons fait une évaluation numérique selon les critères qui définissent la fonction de coût optimisée. Une autre famille de l'évaluation qui constitue une perspective cruciale est l'évaluation subjective faisant intervenir l'humain.

Mohseni et al., 2021 ou encore Doshi-Velez and Kim, 2017 proposent des taxonomies pour effectuer des expérimentations avec des utilisateurs. Cependant, comme l'évoquent Lopes et al., 2022, il y a un manque d'évaluation par les méthodes de la littérature : ainsi dans l'étude comparative présentée par Adadi and Berrada, 2018 sur 381 approches seules 5% d'entre elles proposent une évaluation utilisateur. Cela est notamment dû au fait que mettre en place une expérimentation utilisateur peut représenter un coût, mais surtout il y a des contraintes comme le fait que les utilisateurs n'ont pas des connaissances techniques et que l'expérimentation proposée doit être compréhensible par tous. Une piste intéressante est d'effectuer différentes expérimentations avec des utilisateurs pour valider les méthodes proposées dans nos travaux.

Dans cette thèse, nous avons comparé des explications qui minimisent différents critères comme la pénalité, l'incompatibilité ou la diversité. Une piste d'expérimentation utilisateur intéressante est d'observer pour un critère choisi la définition préférée par les utilisateurs. Ainsi, une piste est de proposer à l'utilisateur des explications associées à un critère mais définies différemment, ce qui permet de choisir la définition la plus adaptée. Cette expérimentation permettrait de définir un ensemble de préférences utilisateurs et d'apprendre la définition la plus adaptée.

Bibliographie

Amina Adadi and Mohammed Berrada. Peeking Inside the Black-Box : A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6 :52138–52160, 2018.

Shaaron Ainsworth. *The Educational Value of Multiple-representations when Learning Complex Scientific Concepts*, pages 191–208. Springer, 2008.

Daniel W. Apley and Jingyu Zhu. Visualizing the Effects of Predictor Variables in Black Box Supervised Learning Models. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 82(4) :1059–1086, 2016.

André Artelt and Barbara Hammer. *Convex Density Constraints for Computing Plausible Counterfactual Explanations*, pages 353–365. Springer, 2020.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11 :1803–1831, 2010.

Solon Barocas, Andrew D. Selbst, and Manish Raghavan. The hidden assumptions behind counterfactual explanations and principal reasons. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (FAT* 20)*, page 80–89, 2020.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (XAI) : Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58 :82–115, 2020.

Maximilian Becker. Personalized explanations. In *Proceedings of the 2022 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*, pages 1–10, 2023.

Maximilian Becker, Nadia Burkart, Pascal Birnstill, and Jürgen Beyerer. A step towards global counterfactual explanations : Approximating the feature space through hierarchical division and graph search. *Advances in Artificial Intelligence and Machine Learning*, 1(2) :90–110, 2021.

Umang Bhatt, Isabel Chien, Muhammad Bilal Zafar, and Adrian Weller. Divine : Diverse influential training points for data visualization and model refinement. *preprint arXiv :2107.05978*, 2021.

Christopher Blier-Wong, Hélène Cossette, Luc Lamontagne, and Etienne Marceau. Machine learning in P&C insurance : A review for pricing and reserving. *Risks*, 9(1), 2021.

Jim Blythe, Jihie Kim, Surya Ramachandran, and Yolanda Gil. An integrated environment for knowledge acquisition. In *Proceedings of the 6th International Conference on Intelligent User Interfaces*, IUI '01, page 13–20, 2001.

Francesco Bodria, Fosca Giannotti, Riccardo Guidotti, Francesca Naretto, Dino Pedreschi, and Rinzivillo Salvatore. Benchmarking and Survey of Explanation Methods for Black Box Models. *preprint arxiv :2102.13076*, 2021.

Grégory Bourguin, Arnaud Lewandowski, Mourad Bouneffa, and Adeel Ahmad. Towards ontologically explainable classifiers. In *Artificial Neural Networks and Machine Learning, ICANN 2021*, pages 472–484, 2021.

Clara Bove, Jonathan Aigrain, Marie-Jeanne Lesot, Charles Tijus, and Marcin Detyniecki. Contextualization and exploration of local feature importance explanations to improve understanding and satisfaction of non-expert users. In *27th International Conference on Intelligent User Interfaces, IUI'22*, page 807–819, 2022.

Clara Bove, Marie-Jeanne Lesot, Charles Tijus, and Marcin Detyniecki. Investigating the intelligibility of plural counterfactual examples for non-expert users : an explanation user interface proposition and user study. In *28th International Conference on Intelligent User Interfaces, IUI'23*, page 188–203, 2023.

Ngoc Bui, Duy Nguyen, and Viet Nguyen. Counterfactual plans under distributional ambiguity. *International Conference on Learning Representations*, 2022.

Nadia Burkart and Marco F. Huber. A Survey on the Explainability of Supervised Machine Learning. *Journal of Artificial Intelligence Research*, 70 :245–317, 2021.

Tomasa Calvo, Gaspar Mayor, and Radko Mesiar. *Aggregation Operators : New Trends and Applications*, volume 97. Springer, 2002.

Rachele Carli, Amro Najjar, and Davide Calvaresi. Risk and exposure of XAI in persuasion and argumentation : The case of manipulation. In *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, pages 204–220. Springer, 2022.

Miguel Á Carreira-Perpiñán and Suryabhan Singh Hada. Counterfactual explanations for oblique decision trees : Exact, efficient algorithms. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8) :6903–6911, 2021.

Vinay Chamola, Vikas Hassija, A Razia Sulthana, Debshishu Ghosh, Divyansh Dhingra, and Biplab Sikdar. A review of trustworthy and explainable artificial intelligence (XAI). *IEEE Access*, 11 :78994–79015, 2023.

Furui Cheng, Yao Ming, and Huamin Qu. Dece : Decision explorer with counterfactual explanations for machine learning models. *IEEE Transactions on Visualization and Computer Graphics*, 27(2) :1438–1447, 2020.

Timothy Chklovski and Yolanda Gil. Improving the design of intelligent acquisition interfaces for collecting world knowledge from web contributors. In *Proceedings of the 3rd International Conference on Knowledge Capture*, page 35–42. ACM, 2005.

Michael Chromik and Andreas Butz. Human-XAI interaction : A review and design principles for explanation user interfaces. In *Human-Computer Interaction*, pages 619–640, 2021.

Dennis Collaris, Leo M Vink, and Jarke J van Wijk. Instance-level explanations for fraud detection : A case study. *preprint arXiv :1806.07129*, 2018.

Dennis Collaris, Pratik Gajane, Joost Jorritsma, Jarke J. van Wijk, and Mykola Pechenizkiy. Lemon : Alternative sampling for more faithful explanation through local surrogate models. In *Advances in Intelligent Data Analysis XXI : 21st International Symposium on Intelligent Data Analysis, IDA 2023*, page 77–90, 2023.

Commission européenne. Proposal for a regulation laying down harmonised rules on artificial intelligence and amending certain union legislative acts, 2021. URL https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A52021PC0206.

Cristina Conati, Oswald Barral, Vanessa Putnam, and Lea Rieger. Toward personalized XAI : A case study in intelligent tutoring systems. *Artificial Intelligence*, 298 :23, 2021.

Mark W. Craven and Jude W. Shavlik. Extracting tree-structured representations of trained networks. In *Proceedings of the 8th International Conference on NeurIPS, NeurIPS'95*, page 24–30, 1995.

Kristijonas Čyras, Antonio Rago, Emanuele Albini, Pietro Baroni, and Francesca Toni. Argumentative XAI : a survey. *arXiv*, 2021.

Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. Multi-objective counterfactual explanations. In *Parallel Problem Solving from Nature – PPSN XVI*, pages 448–469, 2020.

Arun Das and Paul Rad. Opportunities and Challenges in Explainable Artificial Intelligence (XAI) : A Survey. *preprint arXiv :2006.11371*, 2020.

Jeffrey Dastin. Amazon scraps secret AI recruiting tool that showed bias against women. In *Ethics of data and analytics*, pages 296–299. Auerbach Publications, 2022.

Mark Day. Counterfactual reasoning and method in historical geography. *Journal of Historical Geography*, 36(3) :253–260, 2010.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *preprint arXiv :1702.08608*, 2017.

Michael Downs, Jonathan L. Chu, Yaniv Yacoby, Finale Doshi-Velez, and Pan WeiWei. CRUDS : Counterfactual Recourse Using Disentangled Subspaces. *ICML Workshop on Human Interpretability in Machine Learning*, pages 1–23, 2020.

Didier Dubois and Henri Prade. A theorem on implication functions defined from triangular norms. *Stochastica*, 8(3) :267–279, 1984.

Didier Dubois, Henri Prade, Agnès Rico, and Bruno Teheux. Generalized qualitative Sugeno integrals. *Information Sciences*, 415 :429–445, 2017.

Antonín Dvořák and Michal Holčapek. Fuzzy measures and integrals defined on algebras of fuzzy subsets over complete residuated lattices. *Information Sciences*, 185(1) : 205–229, 2012.

Andrea Ferrario and Michele Loi. How explainability contributes to trust in AI. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, ACM FAccT '22*, page 1457–1466, 2022.

Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful : Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20 :177 :1–177 :81, 2019.

Jerome H. Friedman. Greedy function approximation : A gradient boosting machine. *The Annals of Statistics*, 29(5) :1189 – 1232, 2001.

Christopher Frye, Colin Rowat, and Ilya Feige. Asymmetric Shapley values : Incorporating causal knowledge into model-agnostic explainability. In *Proceedings of the 34th International Conference on NeurIPS, NeurIPS'20*, 2020.

H Kevin Fulk, Heidi L Dent, William A Kapakos, and Barbara Jo White. Doing more with less : Using AI-based big interview to combine exam preparation and interview practice. *Issues in Information Systems*, 23(4), 2022.

Heta Gandhi and Andrew White. Explaining structure-activity relationships using locally faithful surrogate models. *ChemRxiv*, 2022.

Konstantinos Gavriilidis, Andrea Munafo, Wei Pang, and Helen Hastie. A surrogate model framework for explainable autonomous behaviour. *preprint arXiv :2305.19724*, 2023.

Julie Gerlings, Arisa Shollo, and Ioanna Constantiou. Reviewing the need for explainable artificial intelligence (XAI). *preprint arXiv :2012.01007*, 2021.

Bryce Goodman and Seth Flaxman. European Union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38(3) :50–57, 2017.

Michel Grabisch and Christophe Labreuche. A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid. *Annals of Operations Research*, 175 (1) :247–290, 2010.

Michel Grabisch, Jean-Luc Marichal, Radko Mesiar, and Endre Pap. *Aggregation Functions*. Cambridge University Press, 2009.

Nina Grgic-Hlaca, Elissa M. Redmiles, Krishna P. Gummadi, and Adrian Weller. Human perceptions of fairness in algorithmic decision making : A case study of criminal risk prediction. In *Proceedings of the 2018 World Wide Web Conference, WWW'18*, page 903–912, 2018.

Riccardo Guidotti. Counterfactual explanations and how to find them : literature review and benchmarking. *Data Mining and Knowledge Discovery*, pages 1–55, 2022.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A Survey of Methods for Explaining Black Box Models. *ACM Computing Surveys*, 51(5) :1–42, 2018.

Riccardo Guidotti, Anna Monreale, Fosca Giannotti, Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. Factual and counterfactual explanations for black box decision making. *IEEE Intelligent Systems*, 34(6) :14–23, 2019.

Wenbo Guo, Dongliang Mu, Jun Xu, Purui Su, Gang Wang, and Xinyu Xing. LEMNA : Explaining Deep Learning based Security Applications. *Proc. of the 2018 ACM SIGSAC*, page 364–379, 2018.

Suryabhan Singh Hada and Miguel Á Carreira-Perpiñán. Exploring counterfactual explanations for classification and regression trees. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 489–504, 2021.

Satoshi Hara and Kohei Hayashi. Making Tree Ensembles Interpretable : A Bayesian Model Selection Approach. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, pages 77–85, 2016.

Denis J. Hilton. Mental models and causal explanation : Judgements of probable cause and explanatory relevance. *Thinking & Reasoning*, 2 :273–308, 1996.

Michael Hind. Explaining explainable AI. *XRDS Crossroads ACM*, 25 :16–19, 2019.

Adulam Jeyasothy, Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. Integrating prior knowledge in post-hoc explanations. In *Proceedings of the Internation Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 707–719, 2022a.

Adulam Jeyasothy, Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. Intégration de connaissances dans les méthodes d'explications post-hoc. In *Rencontres francophones sur la logique floue et ses applications (LFA)*, 2022b.

Adulam Jeyasothy, Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. A general framework for personalising post hoc explanations through user knowledge integration. *International Journal of Approximate Reasoning*, 160, 2023a.

Adulam Jeyasothy, Agnès Rico, Marie-Jeanne Lesot, Christophe Marsala, and Thibault Laugel. Knowledge Integration in XAI with Gödel Integrals. In *IEEE International Conference on Fuzzy Systems*, 2023b.

Adulam Jeyasothy, Agnès Rico, Marie-Jeanne Lesot, Christophe Marsala, and Thibault Laugel. Intégration de connaissances en XAI avec les intégrales de Gödel. In *Rencontres Francophones sur la Logique Floue et ses Applications (LFA)*, 2023c.

Yunzhe Jia, James Bailey, Kotagiri Ramamohanarao, Christopher Leckie, and Michael E. Houle. Improving the quality of explanations with local embedding perturbations. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD'19*, page 875–884, 2019.

Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, Yuichi Ike, Kento Uemura, and Hiroki Arimura. Ordered counterfactual explanation by mixed-integer linear optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13) :11564–11574, 2021.

Amir-Hossein Karimi, Gilles Barthe, Borja Balle, and Isabel Valera. Model-agnostic counterfactual explanations for consequential decisions. *International Conference on Artificial Intelligence and Statistics*, pages 895–905, 2020.

Laurent Karsenty. Une définition psychologique de l'explication. *Intellectica - La revue de l'Association pour la Recherche sur les sciences de la Cognition (ARCo)*, 23(2) :327–345, 1996.

Leonard Kaufmann and Peter Rousseeuw. Clustering by means of medoids. *Data Analysis based on the L1-Norm and Related Methods*, pages 405–416, 1987.

Davinder Kaur, Suleyman Uslu, Kaley J. Rittichier, and Arjan Durresi. Trustworthy Artificial Intelligence : A Review. *ACM Computer Survey*, 55(2), 2022.

Jalil Kazemitabar, Arash Amini, Adam Bloniarz, and Ameet S Talwalkar. Variable importance using decision trees. In *Advances in NeurIPS*, volume 30, page 425–434, 2017.

Hassan Khosravi, Simon Buckingham Shum, Guanliang Chen, Cristina Conati, Yi-Shan Tsai, Judy Kay, Simon Knight, Roberto Martinez-Maldonado, Shazia Sadiq, and Dragan Gašević. Explainable Artificial Intelligence in education. *Computers and Education : Artificial Intelligence*, 3 :100074, 2022.

Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize ! criticism for interpretability. *Advances in NeurIPS*, 29 :2288–2296, 2016.

Erich Peter Klement, Radko Mesiar, and Endre Pap. *Triangular Norms*. Springer, 2000.

Hima Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. Interpretable & Explorable Approximations of Black Box Models. *KDD Workshop on Fairness, Acountability and Transparency in Machine Learning (FAT ML)*, 2017.

Michael T. Lash, Qihang Lin, Nick Street, Jennifer G. Robinson, and Jeffrey Ohlmann. Generalized Inverse Classification. *Proceedings of the SIAM International Conference on Data Mining*, page 162–170, 2017.

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. Comparison-based Inverse Classification for Interpretability in Machine Learning. In *Proceedings of International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 100–111, 2018a.

Thibault Laugel, Xavier Renard, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. Defining locality for surrogates in post-hoc interpretablity. 2018b.

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. Issues with post-hoc counterfactual explanations : a discussion. *preprint arXiv :1906.04774*, 2019.

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. Unjustified classification regions and counterfactual explanations in machine learning. *ECML PKDD*, pages 37–54, 2020.

Thibault Laugel, Adulam Jeyasothy, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. Achieving diversity in counterfactual explanations : A review and discussion. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, FAccT '23*, page 1859–1869, 2023.

Dan Ley, Umang Bhatt, and Adrian Weller. Diverse, global and amortised counterfactual explanations for uncertainty estimates. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7) :7390–7398, 2022.

Q. Vera Liao, Daniel Gruen, and Sarah Miller. Questioning the AI : Informing design practices for explainable ai user experiences. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20. ACM, 2020.

Pedro Lopes, Eduardo Silva, Cristiana Braga, Tiago Oliveira, and Luís Rosado. XAI systems evaluation : A review of human and computer-centred methods. *Applied Sciences*, 12(19), 2022.

Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on NeurIPS*, pages 4768–4777, 2017.

Scott Lundberg, Gabriel Erion, and Su-In Lee. Consistent individualized feature attribution for tree ensembles. *preprint arXiv :1802.03888*, 2018.

Divyat Mahajan, Chenhao Tan, and Amit Sharma. Preserving causal constraints in counterfactual explanations for machine learning classifiers. *preprint arXiv :1912.03277*, 2019.

Jean-Luc Marichal. On Sugeno integral as an aggregation function. *Fuzzy Sets and Systems*, 114(3) :347–365, 2000.

Naser Masri, Yousef Abu Sultan, Alaa N Akkila, Abdelbaset Almasri, Adel Ahmed, Ahmed Y Mahmoud, Ihab Zaqout, and Samy S Abu-Naser. Survey of rule-based systems. *International Journal of Academic Information Systems Research (IJAISR)*, 3(7) : 1–23, 2019.

Raphael Mazzine and David Martens. A framework and benchmarking study for counterfactual generating methods on tabular data. *Applied Science*, 2021.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys*, 54 :1–35, 2021.

Tim Miller. Explanation in artificial intelligence : Insights from the social sciences. *Artificial Intelligence*, 267 :1–38, 2019.

Kiarash Mohammadi, Amir-Hossein Karimi, Gilles Barthe, and Isabel Valera. Scaling guarantees for nearest counterfactual explanations. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 177–187, 2021.

Sina Mohseni, Niloofar Zarei, and Eric D. Ragan. A multidisciplinary survey and framework for design and evaluation of explainable ai systems. *ACM Trans. Interact. Intell. Syst.*, 11(3–4), 2021.

Christoph Molnar. *Interpretable machine learning, A Guide for Making Black Box Models Explainable*. 2022.

Jonathan Moore, Nils Hammerla, and Chris Watkins. Explaining deep learning models with constrained adversarial examples. *Pacific Rim international conference on artificial intelligence*, pages 43–56, 2019.

Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT*'20*, page 607–617, 2020.

Mervin E. Muller. A Note on a Method for Generating Points Uniformly on N-Dimensional Spheres. *Commun. ACM*, 2(4) :19–20, 1959.

Philip Naumann and Eirini Ntoutsi. Consequence-aware sequential counterfactual generation. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 682–698, 2021.

Guillermo Navas-Palencia. Optimal counterfactual explanations for scorecard modelling. *preprint arXiv :2104.08619*, 2021.

Duy Nguyen, Ngoc Bui, and Viet Anh Nguyen. Feasible recourse plan via diverse interpolation. *International Conference on Artificial Intelligence and Statistics*, pages 4679–4698, 2023.

Angela Nyhout and Patricia A. Ganea. Mature counterfactual reasoning in 4- and 5-year-olds. *Cognition*, 183 :57–66, 2019.

Cecilia Panigutti, Ronan Hamon, Isabelle Hupont, David Fernandez Llorca, Delia Fano Yela, Henrik Junklewitz, Salvatore Scalzo, Gabriele Mazzini, Ignacio Sanchez, Josep Soler Garrido, and Emilia Gomez. The role of explainable AI in the context of the AI Act. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, FaccT'23*, page 1139–1150, 2023.

Axel Parmentier and Thibaut Vidal. Optimal counterfactual explanations in tree ensembles. *International Conference on Machine Learning*, pages 8422–8431, 2021.

Yves Poullet. *Le RGPD face aux défis de l'intelligence artificielle*. Éditions Larcier, 2021.

Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. FACE : Feasible and Actionable Counterfactual Explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, AIES'20*, page 344–350, 2020.

Rafael Poyiadzi, Xavier Renard, Thibault Laugel, Raul Santos-Rodriguez, and Marcin Detyniecki. On the overlooked issue of defining explanation objectives for local-surrogate explainers. *preprint arXiv :2106.05810*, 2021.

Peyman Rasouli and Ingrid Chieh Yu. CARE : Coherent Actionable Recourse based on sound counterfactual Explanations. *International Journal of Data Science and Analytics*, pages 1–26, 2022.

Kaivalya Rawal and Himabindu Lakkaraju. Beyond individualized recourse : Interpretable and interactive summaries of actionable recourses. *Advances in NeurIPS*, 33 : 12187–12198, 2020.

Annabelle Redelmeier, Martin Jullum, Kjersti Aas, and Anders Løland. MCCE : Monte Carlo sampling of realistic Counterfactual Explanations. *arXiv preprint arXiv :2111.09790*, 2021.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should I trust you ?" Explaining the predictions of any classifier. *Proc. of the 22nd ACM SIGKDD Int. Conf. on knowledge discovery and data mining*, pages 1135–1144, 2016.

Pau Rodríguez, Massimo Caccia, Alexandre Lacoste, Lee Zamparo, Issam Laradji, Laurent Charlin, and David Vazquez. Beyond trivial counterfactual explanations with diverse valuable explanations. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1056–1065, October 2021.

Cynthia Rudin. Stop Explaining Black Box Machine Learning Models for High Stakes decisions and Use Interpretable Models Instead. *Nature Machine Intelligence*, 1 :206–215, 2019.

Chris Russell. Efficient search for diverse coherent explanations. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 20–28, 2019.

Hicham Sadok, Fadi Sakka, and Mohammed El Hadi El Maknouzi. Artificial intelligence and bank credit analysis : A review. *Cogent Economics & Finance*, 10(1) :2023262, 2022.

Robert-Florian Samoilescu, Arnaud Van Looveren, and Janis Klaise. Model-agnostic and scalable counterfactual explanations via reinforcement learning. *arXiv preprint arXiv :2106.02597*, 2021.

Maximilian Schleich, Zixuan Geng, Yihong Zhang, and Dan Suciu. GeCo : Quality counterfactual explanations in real time. *Proc. VLDB Endow.*, 14(9) :1681–1693, 2021. ISSN 2150-8097.

Shubham Sharma, Jette Henderson, and Joydeep Ghosh. CERTIFAI : A common framework to provide explanations and analyse the fairness and robustness of black-box models. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, page 166–172, 2020.

Niel Shilkret. Maxitive measure and integration. *Indagationes Mathematicae (Proceedings)*, 74 :109–116, 1971.

Herbert A. Simon. What is an "explanation" of behavior ? *Psychological Science*, 3 :150–161, 1992.

Lars Sipos, Ulrike Schäfer, Katrin Glinka, and Claudia Müller-Birn. Identifying explanation needs of end-users : Applying and extending the XAI question bank. In *Mensch und Computer 2023*, MuC '23. ACM, 2023.

Dylan Slack, Anna Hilgard, Himabindu Lakkaraju, and Sameer Singh. Counterfactual explanations can be manipulated. *Advances in neural information processing systems*, 34 :62–75, 2021.

Barry Smyth and Mark T Keane. A few good counterfactuals : generating interpretable, plausible and diverse counterfactual explanations. *International Conference on Case-Based Reasoning*, pages 18–32, 2022.

Rand J Spiro, Paul J Feltovich, Richard L Coulson, and Daniel K Anderson. Multiple analogies for complex concepts : antidotes for analogy-induced misconception in advanced knowledge acquisition. pages 498–531. 1989.

Ramya Srinivasan and Ajay Chander. Explanation perspectives from the cognitive sciences–a survey. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4812–4818, 7 2020.

Ilia Stepin, Jose M. Alonso, Alejandro Catala, and Martín Pereira-Fariña. A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. *IEEE Access*, 9 :11974–12001, 2021.

Michio Sugeno. *Theory of fuzzy integrals and its applications*. PhD thesis, Tokyo Institute of Technology, 1974.

Emily Sullivan and Philippe Verreault-Julien. From explanation to recommendation : Ethical standards for algorithmic recourse. *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, pages 712–722, 2022.

Hendra Suryanto and Paul Compton. Learning classification taxonomies from a classification knowledge based system. In *ECAI Workshop on Ontology Learning*, 2000.

Stratis Tsirtsis and Manuel Gomez Rodriguez. Decisions, counterfactual explanations and strategic behavior. *Advances in NeurIPS*, 33 :16749–16760, 2020.

Berk Ustun, Alexander Spangher, and Yang Liu. Actionable Recourse in Linear Classification. *Proc. of the Conf. on Fairness, Accountability, and Transparency*, page 10–19, 2019.

Arnaud Van Looveren and Janis Klaise. Interpretable Counterfactual Explanations Guided by Prototypes. *Proc. of European Conf. on Machine Learning*, 2021.

Sahil Verma, John Dickerson, and Keegan Hines. Counterfactual explanations for machine learning : A review. *arXiv preprint arXiv :2010.10596*, 2020.

Sahil Verma, Keegan Hines, and John P Dickerson. Amortized generation of sequential counterfactual explanations for black-box models. *arXiv preprint arXiv :2106.03962*, 2021.

Giulia Vilone and Luca Longo. Notions of explainability and evaluation approaches for explainable artificial intelligence. *Information Fusion*, 76 :89–106, 2021. ISSN 1566-2535.

James Vincent. Google 'fixed' its racist algorithm by removing gorillas from its image-labeling tech. *The Verge*, 12 :2018, 2018.

Klaus Virtanen. Using XAI tools to detect harmful bias in ML models, 2022.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual Explanations without Opening the Black Box : Automated Decisions and the GDPR. *Harvard journal of law & technology*, 31 :841–887, 2018.

Danding Wang, Qian Yang, Ashraf Abdul, and Brian Y Lim. Designing theory-driven user-centric explainable AI. *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–15, 2019.

Jia Wang, Yuchao Su, Qiuzhen Lin, Lijia Ma, Dunwei Gong, Jianqiang Li, and Zhong Ming. A survey of decomposition approaches in multiobjective evolutionary algorithms. *Neurocomputing*, 408 :308–330, 2020.

Prateek Yadav, Peter Hase, and Mohit Bansal. Low-cost algorithmic recourse for users with uncertain cost functions. *arXiv preprint arXiv :2111.01235*, 2021.

Fan Yang, Sahan Suresh Alva, Jiahao Chen, and Xia Hu. Model-based counterfactual synthesizer for interpretation. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1964–1974, 2021.

Wenzhuo Yang, Jia Li, Caiming Xiong, and Steven CH Hoi. MACE : An efficient Model-Agnostic framework for Counterfactual Explanation. *arXiv preprint arXiv :2205.15540*, 2022.

Alessio Zanga, Elif Ozkirimli, and Fabio Stella. A survey on causal discovery : Theory and practice. *International Journal of Approximate Reasoning*, 151 :101–129, 2022.