

Comment gagner : expliquer les bonnes trajectoires

M. Kazi Aoual^{1,2}, C. Rouveirol², H. Soldano^{1,2,3}, V. Ventos¹

¹ Nukkai, France

² UMR CNRS 7030 Institut Galilée – Université Sorbonne Paris Nord, LIPN

³ UMR CNRS 7205 Museum National d’Histoire Naturelle, ISYEB

mkazi[at]nukk.ai / rouveirol[at]lipn.univ-paris13.fr / henry.soldano[at]mnhn.fr / vventos[at]nukk.ai

Résumé

Dans un projet de construction d’un joueur artificiel pour un jeu de bridge très simplifié, nous nous intéressons ici à la question de la manière de jouer pour gagner. Cela nous amène à faire de l’apprentissage relationnel et à proposer une technique de constructions d’explications de trajectoires, ce qui implique en particulier de construire les clauses minimales couvrant un ensemble d’exemples. Nous présentons une étude de cas et discutons des questions encore ouvertes dans ce projet.

Mots-clés

Explications, Programmation Logique Inductive, Abduction

Abstract

To address how to build an artificial player for a simplified bridge game, we are interested in the question of how to play to win. This leads us to do relational learning and to propose a technique for constructing trajectory explanations, which involves in particular constructing minimal clauses covering a set of examples. We present a case study and discuss the open questions in this project.

Keywords

Explanation, Inductive Logic Programming, Abduction

1 Introduction

Nous considérons les explications du classement d’une observation par un classifieur logique lorsque cette observation est un arbre des actions possibles au cours du temps et des états associés. Chaque branche représente alors une trajectoire possible. Notre motivation est un scénario où un joueur artificiel joue à un jeu de bridge simplifié, dans la position du déclarant contre un programme jouant le défenseur, et doit répondre à tout moment du jeu à des requêtes émises par un interlocuteur humain et de la forme "comment l’action choisie conduit-elle à un nombre de plis optimal?". On optimise ici la récompense totale maximale que l’on peut obtenir en suivant une trajectoire à partir d’un état donné, au sens d’un processus de décision de Markov (MDP)[23].

Les explications recherchées pour le choix d’une action a dans un état donné s portent alors sur un certain nombre de

trajectoires optimales possibles. Comme ce nombre peut-être élevé nous allons regrouper ces trajectoires optimales de manière à ce que l’optimalité des trajectoires de chaque groupe puisse être expliquée de la même manière. Cela nous amène à définir d’abord ce qu’est une *explication commune* pour l’optimalité d’un groupe de trajectoires. La première étape du processus explicatif est alors de former des groupes de trajectoires optimales ayant chacun au moins une explication commune de l’optimalité de ses trajectoires. Pour une paire (état, action) donnée nous proposons d’obtenir ces groupes en construisant un classifieur logique à base de règles. Les trajectoires *couvertes* par une règle, c’est-à-dire celles satisfaisant sa prémisse, auront alors au moins une explication commune par définition.

La notion d’explication explorée ici provient de divers travaux sur les explications abductives du label attribué par un classifieur logique à une observation décrite en logique propositionnelle ou par une liste de paires attribut-valeur [10, 2, 3, 6]. Nous avons adapté et étendu ces définitions dans plusieurs directions. Nos observations sont ici des trajectoires : nous associons à une paire (état, action) donnée (s, a) l’ensemble des trajectoires *possibles* que nous appelons *l’univers* associé à (s, a) . Nous supposons également avoir un classifieur logique D qui étiquette les trajectoires comme optimales ou non-optimales. Les explications sont alors adaptées comme suit :

- Une trajectoire est décrite en logique d’ordre un comme un ensemble de faits, c’est-à-dire d’atomes instanciés. L’explication de l’optimalité d’une trajectoire p est alors définie comme un sous-ensemble des faits décrivant la trajectoire.
- L’explication de l’optimalité de p dépend du classifieur D mais aussi de l’univers des trajectoires possibles. Dans la définition U sera considéré comme une formule dont les modèles sont ses trajectoires.
- Une *explication commune* pour un ensemble de trajectoires optimales est une conjonction existentiellement quantifiée sans variable libre, que nous appelons par la suite un *motif relationnel*, c’est-à-dire la forme générale proposée pour une explication abductive en ordre un par P. Marquis [12].

Nous représenterons ce qui est commun à un groupe de trajectoires optimales O par le motif relationnel maximalement spécifique au sens de la θ -

subsomption satisfait par les trajectoires du groupe, et noté $lgg(O)$ ¹. Nous définirons les explications communes comme des sous-ensembles de la lgg du groupe.

Dans la mise en oeuvre l'univers est un ensemble de trajectoires partitionné en U_{opt} (optimales) et U_{notOpt} (non-optimales) à partir desquels nous construirons par Programmation Logique Inductive (PLI) [15] un classifieur D . Une règle r de D est alors une clause de la forme $opt \leftarrow q$ qui couvre une trajectoire quand q est satisfaite. Par construction elle couvre un sous-ensemble U_r de U_{opt} et aucune trajectoire de U_{notOpt} . U_r définit donc un groupe de trajectoires optimales pour lequel on cherchera des explications communes.

Nous réduisons le problème de l'énumération des explications communes minimales de l'optimalité d'un tel groupe de trajectoires à celui de l'énumération des sous-ensembles minimaux m de la lgg du groupe tels que $opt \leftarrow m$ ne couvre aucune trajectoire de U_{notOpt} . Techniquement cela amène à résoudre deux problèmes :

1. La construction de la lgg d'un ensemble d'observations sous θ -subsomption.
2. La recherche des sous-ensembles minimaux de littéraux de la lgg qui ne θ -subsument pas un sous-ensemble d'observations donné.

Ces problèmes ont été étudiés dans différents contextes. La construction de la lgg, unique sous θ -subsomption, a été utilisée en particulier dans les méthodes de PLI ascendantes depuis l'article de Plotkin [17]. Cependant pour limiter la taille de la lgg, ce qui est impératif pour notre propos, nous construirons une approximation de celle-ci en précisant un certain nombre de contraintes déclaratives. La recherche des sous-ensembles minimaux satisfaisant une contrainte de couverture a été étudiée en fouille de données [22], mais peu abordée pour des données relationnelles pour lesquelles différentes propriétés algorithmiques essentielles ne sont pas satisfaites [9]. Les algorithmes proposés pour ces deux problèmes, en particulier le second, font partie des contributions de cet article, et sont décrits section 5.

Nous montrons en section 6 sur un cas d'étude simple, le type d'explications obtenues et le sens ou l'intérêt qu'elles ont relativement aux buts poursuivis.

2 Scenario

2.1 Le jeu

Nous considérons un jeu de cartes de type bridge simplifié avec une seule couleur dont les 13 cartes sont réparties entre les joueurs en une *donne*. Le nombre de cartes de chaque joueur dans la donne n'est pas fixé. On considère les enchères terminées par un contrat demandé par Sud et qui oppose donc le déclarant, c'est-à-dire la paire North-South (NS) au défenseur, la paire West-East (WE). Nous traitons ici du jeu de la carte commençant quand West vient de jouer la première carte du premier pli. La main de North est dévoilée sur la table, et c'est au déclarant North de jouer.

1. pour Least General Generalisation

2.2 Le scenario

Dans le scénario ci-dessous notre joueur artificiel *Noo* répond à des requêtes d'explications de ses décisions émises par son interlocuteur *M. X*. Chaque requête et ses réponses peuvent initier un ensemble d'interactions. Dans le scénario général, les mains *WE* sont inconnues du déclarant *NS*, mais dans cet article nous considérons le scénario simplifié dans lequel les mains *WE* sont connues de *Noo*. Le jeu commence quand West joue, et que les cartes de North sont dévoilées. *Noo* résout le MDP de la donne *NSEW*. On suppose que *Noo* connaît le modèle, ici déterministe, du défenseur : s'il connaît l'état courant du jeu, *Noo* sait la carte que jouera le défenseur *WE*. Une trajectoire partant d'un état s est optimale lorsqu'elle maximise le nombre de plis fait par le déclarant à la fin de la partie. Lorsque le MDP est résolu, dans tout état s du jeu et pour toute action possible a du déclarant en s , la q-valeur $q(s, a)$ est connue. $q(s, a)$ est ici le nombre de plis que le déclarant fera en jouant a en s et en suivant à partir de l'état s' résultant de a une trajectoire optimale. L'action a est optimale en s si elle est de q-valeur maximale. Le scénario simplifié où l'état initial est donc connu est le suivant :

1. Les cartes *WE* sont retournées.
2. *Noo* choisit une action optimale a dans l'état courant s
3. *M. X* demande s'il le souhaite "Comment l'action a conduit-elle à un nombre de plis maximal?"
4. *Noo* construit alors un modèle D permettant de distinguer dans l'ensemble U des trajectoires possibles commençant par (s, a) , les trajectoires optimales de celles qui ne le sont pas. Il utilise D pour constituer des groupes de trajectoires optimales et propose pour chaque groupe une ou plusieurs explications communes aux trajectoires du groupe.
5. S'ensuivent des interactions entre *Noo* et *M. X* conduisant à des explications alternatives.
6. *Noo* joue a conduisant à un nouvel état courant et le scénario reprend en 2.

Dans cet article nous abordons les étapes 3 et 4 de ce scénario simplifié.

3 Notations dans les représentations relationnelles

On traite ici des langages Datalog (i.e. Logique des prédicats sans autre symbole de fonctions que des constantes). Les seuls termes sont des *constantes* et *variables*. Les constantes sont soit des nombres soit des atomes commençant par une minuscule. Par exemple, on peut représenter des cartes par des entiers de l'intervalle [2..14] et les quatre joueurs par les constantes *west*, *north*, *east*, *south*. Les autres termes sont des variables, identifiées par des symboles commençant par une majuscule (X , Y , *Card*, ...). Un *littéral* est un symbole de prédicat appliqué à des termes. Un *fait* est un littéral complètement instancié (sans variables). Par exemple, le littéral *small_card(C)* énonce

que la variable C est une petite carte (i.e. entre 2 and 10), alors que $honor(12)$ est un fait qui déclare que 12 (représentant le valet) est un honneur. Par la suite, nous utiliserons la notation classique p/N où p est un symbole de prédicat et N est un entier indiquant le nombre d'arguments du prédicat (i.e. $small_card/1$ indique un prédicat à un seul argument).

Dans le domaine de la PLI qui est notre cadre, on travaille traditionnellement avec deux types de formules : des *clauses définies* et des *conjonctions quantifiées existentiellement*. Une *clause* est une disjonction de littéraux quantifiés universellement $\forall[h_1 \vee \dots \vee h_m \vee \neg b_1 \vee \dots \vee \neg b_n]$ ou de manière équivalente $\forall[(h_1 \vee \dots \vee h_m) \leftarrow (b_1 \wedge \dots \wedge b_n)]$ où h_1, \dots, h_m représente une disjonction de littéraux positifs (la tête de la clause) et b_1, \dots, b_n est la conjonction de littéraux formant le corps de la clause.

Une *clause définie* est une clause avec exactement un littéral positif. Si la tête est un littéral sans argument, le corps de la clause, appelé dans la suite *motif relationnel* est la conjonction quantifiée existentiellement qui correspond au corps d'une clause. De la même manière qu'on omet classiquement le quantificateur \forall lorsqu'il est clair que l'on manipule des clauses, on omettra dans la suite de de l'article le quantificateur \exists lorsqu'il est clair qu'on parle d'un motif relationnel. Nous allons adapter dans la suite toutes les définitions introduites initialement pour des clauses à des motifs relationnels.

Le vocabulaire \mathcal{V} d'un programme Datalog P est l'ensemble de ses constantes et symboles de prédicats. L'*univers de Herbrand* d'un programme Datalog P est l'ensemble des terms clos construit sur \mathcal{V} , i.e., les constantes de \mathcal{V} , la *base de Herbrand* est l'ensemble des faits construits sur l'univers de Herbrand et les symboles de prédicats de \mathcal{V} . Une *interprétation de Herbrand* de P est un sous-ensemble de la base de Herbrand de P .

La relation de généralité entre deux clauses en PLI [15] est le relation de θ -subsumption [17] que nous étendons ici à des motifs relationnels.

Définition 1. *Un motif relationnel G θ -subsume un motif relationnel S (noté $G \preceq_{\theta} S$) si et seulement si (ssi) il existe une substitution θ telle que $G.\theta \subseteq S$.*

Plotkin a également introduit la notion de *généralisation maximale spécifique* ou *lgg* de deux clauses que nous reformulons ci-dessous pour des motifs relationnels.

Définition 2. *Un motif relationnel S est le généralisé le plus spécifique d'un ensemble de motifs relationnels O (noté par $S = lgg(O)$) si et ssi $S \preceq_{\theta} o_i$ pour tout $o_i \in O$ et pour tout motif G tel que $G \preceq_{\theta} o_i$ pour tout $o_i \in O$, alors $G \preceq_{\theta} S$. La lgg de deux motifs relationnels C et D est unique et calculée en temps $\mathcal{O}(|C||D|)$ [16].*

On appelle dans la suite *observation* une interprétation de Herbrand dans le vocabulaire \mathcal{V} (voir le cadre de l'apprentissage à partir d'interprétations [4] ou d'intéprétations [9]). La relation de couverture entre un motif relationnel C et une observation o est définie par *couvre*(C, o) ssi il existe une substitution θ telle que $C \preceq_{\theta} conj(o)$ où $conj(o)$ est

le motif relationnel complètement instancié correspondant à o (la conjonction de tous les faits de o) et O est un ensemble d'interprétations de Herbrand étant donné \mathcal{V} . Par abus de langage et afin d'alléger les notations, nous identifierons l'observation o et la conjonctions associée à l'observation $conj(o)$. La lgg d'un ensemble d'observations O est définie comme $lgg(\{o_i | o_i \in O\})$ et donc en particulier $lgg(\{o\}) = o$.

Exemple 1. *Soit \mathcal{V} un vocabulaire avec quatre constantes $\{1, 2, 3, 4\}$ et trois symboles de prédicats $\{p/2, r/1, q/1\}$ et deux observations o_1 et o_2 , $o_1 = \{p(1, 2), r(2), p(2, 3), q(3)\}$ et $o_2 = \{p(1, 3), q(3), p(2, 4), r(4)\}$. La lgg des deux observations o_1 et o_2 est : $\exists p(1, X), p(X', Y'), r(Y'), p(X'', 3), q(3), p(2, Y'')$ avec les substitutions $\theta_1 = \{X/2, X'/1, Y'/2, X''/2, Y''/3\}$ et $\theta_2 = \{X/3, X'/2, Y'/4, X''/1, Y''/4\}$. Notons que la lgg de o_1 et o_2 est plus longue (en nombre de littéraux) que o_1 et o_2 .*

4 Explications décisionnelles

4.1 Trajectoires et leur représentation

1. Une trajectoire est une séquence $s_0 a_0 \dots s_t a_t \dots s_n a_n$ où s_t est l'état observé au temps t et a_t l'action effectuée en t et conduisant en l'état s_{t+1} .
2. Nous considérerons par la suite l'arbre représentant le sous-ensemble des trajectoires possibles partant d'un état s_0 suivi d'une action a_0 , s_0 étant un des états de l'arbre complet des trajectoires de la donne NSEW étudiée.
3. Chaque trajectoire est décrite par l'ensemble des faits vrais dans la trajectoire. La plupart des prédicats utilisés sont dits temporels : ils sont vrais ou faux selon l'instant t considéré dans la trajectoire.

Nous avons introduit des arguments de type intervalle dans une partie des prédicats décrivant les états du jeu le long d'une trajectoire. Ces intervalles représentent un segment temporel dans lequel le prédicat est vrai entre ses bornes et faux à l'extérieur. Nous nous plaçons ainsi dans la lignée de travaux classiques de représentation des intervalles [1] et de fouille de données séquentielles [14].

Dans ce cas, pour un atome a donné, instancié sauf pour sa partie temporelle, nous divisons le domaine temporel t_0, \dots, t_n en intervalles durant lesquels a est vrai. L'atome $a([b, e])$ est alors vrai lorsque a est vrai pour tout $t \in [b, e]$ et faux aux temps $b - 1$ (si $b \neq 0$) et $e + 1$ (si $e \neq n$).

Exemple 2. *Soit une trajectoire où a est vrai aux temps 1, 2, 3, 5, 6 et b est vrai aux temps 2, 3, 4, 6, 8, 9. La trajectoire s'écrit $a([1, 3]), a([5, 6]), b([2, 4]), b([6, 6]), b([8, 9])$.*

De tels prédicats sont utilisés dans l'étude de cas en section 6.

4.2 Explications

Les *explications abductives* ont récemment été définies pour rendre compte de l'étiquette attribué à une observation o par un arbre de décision [10, 2, 3, 6]. Dans ces travaux l'observation est représentée par un ensemble ou une conjonction de paires attribut-valeur. Une explication abductive minimale du classement de o par l'arbre de décision D est définie comme une conjonction incluse dans o et suffisante pour classer o avec l'étiquette c . Dans notre contexte nous devons changer et étendre cette définition de plusieurs manières. En particulier chaque observation o est une interprétation de Herbrand d'un langage Datalog, représentée par une conjonction de littéraux instanciés (voir la section 3). Le classifieur D est alors une formule de ce langage. L'explication d'un classement peut alors être représentée comme une clause instanciée [19]. Cependant pour notre définition des explications communes, nous reprenons plutôt ici la définition générale d'une explication abductive en ordre un proposée par P. Marquis [12], sous la forme d'un motif relationnel et étudiée par la suite d'un point de vue opérationnel [11, 8].

Nous considérons de plus que l'ensemble des observations possibles dans le problème en cours ne forme qu'une partie, appelée l'*univers*, des interprétations syntaxiquement correctes. Dans les définitions qui suivent l'univers est représenté par les modèles d'une formule U . Notre connaissance U, D du problème courant est alors divisé en une partie U restreignant les observations o autorisées et une partie D permettant d'inférer le label d'une observation. Ceci nous amène à la définition suivante ajoutant U à la définition usuelle d'une explication abductive de l'inférence d'une étiquette pour une observation :

Définition 3 (Explication minimale de l'étiquette d'une observation). *Une explication de l'étiquette c d'une observation o relativement au classifieur D et à l'univers représenté par U est une conjonction $e \subseteq o$ telle que $e, U, D \models c$.*

Si pour tout $e' \subset e$ nous avons $e', U, D \not\models c$ alors e est une explication minimale de l'étiquette c de o relativement à D et U .

Dans l'exemple ci-dessous nous illustrons les explications minimales et montrons que, comme attendu, on obtient des explications minimales différentes selon que l'on utilise ou non l'univers des observations possibles.

Exemple 3. *Soient les étiquettes $+$ et $-$. D est une ensemble de clauses définies et nous considérons que si $o, D, U \not\models +$ alors D attribue l'étiquette $-$ à o .*

Soient p et r deux prédicats unaires et le domaine de constantes $\{1, 2\}$ formant la base de Herbrand $\{p(1), p(2), r(1), r(2)\}$. Une observation est notée comme un sous-ensemble de cette base. Soient le classifieur D et l'observation o_1 définis comme suit :

- $o_1 = \{p(1), p(2), r(1)\}$
- $D = \{+ \leftarrow p(X), r(X); + \leftarrow p(2)\}$

Considérons d'abord que toute observation est dans l'univers, c'est-à-dire $U = \text{Vrai}$. Les explications minimales de

l'étiquette $+$ attribuée à o_1 sont alors $p(1), r(1)$ et $p(2)$. Ceci résulte de ce que nous avons $p(X), r(X).\{X/1\} = p(1), r(1) \subseteq o_1$, et $p(2) \subseteq o_1$.

Considérons maintenant l'univers représenté par $U = \{p(X) \leftarrow r(X); p(1) \vee p(2)\}$. Comme toute observation doit satisfaire U , de $r(1)$ et U nous déduisons $p(1)$ et de $p(1), r(1), D$ nous déduisons $+$. En conséquence $p(1), r(1)$ n'est plus une explication minimale du classement de o_1 car $r(1)$ explique également l'étiquette.

Nous définissons maintenant une *explication commune* de l'étiquette c commune à un ensemble d'observations O comme un motif relationnel. Pour cela nous cherchons d'abord ce qui est commun à ces observations, sous la forme du généralisé le plus spécifique $\text{lgg}(O)$, puis considérons les explications communes comme des motifs relationnels inclus dans $\text{lgg}(O)$.

Définition 4. *Une explication commune du classement c des observations de O est un motif relationnel e inclus dans $\text{lgg}(O)$ et tel que $e, U, D \models c$.*

Si pour tout $e' \subset e$ on a $e', U, D \not\models c$, e est appelée une explication commune minimale de o .

Notons que par définition, pour tout o et toute explication commune e , e est plus générale que o c'est-à-dire $e \succeq_{\theta} o$.

Exemple 4. *Ajoutons à l'exemple 3 une deuxième observation $o_2 = \{p(1), p(2), r(2)\}$. On obtient $\text{lgg}(\{o_1, o_2\}) = p(1), p(2), r(X)$ avec $\text{lgg}(O).\{X/1\} \subseteq o_1$ et $\text{lgg}(O).\{X/2\} \subseteq o_2$. On observe que de $r(X), U, D$ on peut dériver l'étiquette. En effet, de U et $r(X)$ on déduit $p(X)$ et de $r(X), p(X)$ et D on déduit $+$. Par ailleurs, comme vu précédemment de $p(2)$ et D on déduit $+$. Les explications communes minimales de $\{o_1, o_2\}$ sont $r(X)$ et $p(2)$.*

On a également la propriété suivante :

Proposition 1. *Si e est une explication commune minimale du classement de O , alors pour tout $o \in O$ et pour toute substitution θ telle que $e.\theta \subseteq o$ alors $e.\theta$ est une explication, non nécessairement minimale, du classement de o .*

Dans ce qui suit nous considérons que l'univers est connu à travers l'ensemble de ses modèles $M(U)$ qui est partitionné selon les étiquettes attribuées par D en $\{U_c \mid c \in C, U_c \subseteq M(U)\}$. Dans ce cas, une explication commune e de l'ensemble d'observations $O \subseteq U_c$ est telle que e ne couvre aucune observation de label différent, c'est-à-dire appartenant à $U_{-c} = \bigcup_{d \in C \setminus c} U_d$. On peut alors directement construire les explications communes minimales sans utiliser le classifieur :

Proposition 2. *$e \subseteq \text{lgg}(O)$ est une explication commune minimale du classement des observations de O par D en c si et seulement si $\forall u \in U_{-c}$ e ne couvre pas u et $\forall e' \subset e, \exists u \in U_{-c}$ t.q. e' couvre u .*

Exemple 5. *Nous poursuivons l'exemple 4. L'univers contient 8 observations modèles de U parmi lesquelles*

seule $o_- = \{p(1)\}$ est dans U_- . Nous obtenons alors les explications communes minimales de $\{o_1, o_2\}$, c'est-à-dire $r(X)$ et $p(2)$, comme sous-ensembles minimaux de $p(1), p(2), r(X)$ ne couvrant pas o_- .

Remarquons qu'une explication commune d'un ensemble d'observations quelconque O ayant même label n'existe pas nécessairement. Cependant, lorsque le classifieur D est un ensemble de clauses définies, la lgg de la couverture O d'une clause $c \leftarrow b$, c'est-à-dire l'ensemble des observations $O \subseteq U_c$ couvertes par la clause, est moins générale que b . En conséquence $lgg(O)$ ne couvre aucune observation de U_{-c} et est donc une explication commune de O . Les corps des clauses de D formant un recouvrement de U , on en déduit le résultat suivant :

Proposition 3. *Pour toute observation o de U il existe au moins une explication minimale commune à la couverture d'une clause de D couvrant o .*

Nous exploiterons dans les suivantes ces résultats et définitions en faisant les remarques suivantes :

1. Si on ne dispose pas d'un classifieur, mais que l'on connaît les labels des observations de U on peut en construire un par PLI dont on supposera qu'il ne commet pas d'erreurs sur $M(U)$.
2. Les explications communes minimales d'un sous-ensemble d'observations O dépendent de la clause de D dont la couverture est O et de U_{-c} .
3. Par univers on entend ici l'ensemble des observations possibles, mais cet univers est contextuel : Dans le cas des trajectoires U représente le sous-ensemble des trajectoires possibles à un moment du jeu, donc un sous-arbre de l'arbre de jeu.

5 Construction d'explications

5.1 Approximation de la généralisation maximale spécifique d'un sous-ensemble de trajectoires

La complexité du calcul de la lgg exacte pour un ensemble d'observations O est prohibitive, dans le pire des cas en $\mathcal{O}(C)^n$ où C est la taille (en nombre de littéraux) de la plus grande observation de O et $n = |O|$. Sans détailler l'algorithme de Plotkin, pour un symbole de prédicat donné p et étant donné deux observations o_1 et $o_2 \in O$, l'algorithme naïf de calcul de la lgg calcule pour chaque symbole de prédicat p et pour toutes les paires de littéraux $\in selection(p, o_1) \times selection(p, o_2)$ un littéral maximum spécifique qui θ -subsume chacune de ces paires. Considérant que dans notre étude de cas, les observations ont un nombre moyen de littéraux autour de 260, certains symboles de prédicats ayant un nombre d'instances moyen par observation autour de 10, il n'y a pas moyen d'obtenir une lgg en temps raisonnable.

Nous avons donc adapté pour ce faire un algorithme descendant de type *générer et tester* qui permet de calculer un sous-ensemble de $lgg(O)$ appelé *Bottom* dans la suite

de l'article (voir Algorithme 1). Cet algorithme prend en compte des contraintes pertinentes pour notre problème et de nature à borner la taille de *Bottom*.

Nous définissons dans un premier temps et de façon classique en PLI un biais de langage \mathcal{B} comme une liste de symboles de prédicats associés aux types de leurs arguments. Ce biais de langage permet de construire un motif relationnel *Bottom* (et des explications associées) pour un sous-ensemble des prédicats de \mathcal{V} .

Dans les sections suivantes, et comme dans [9], on représente les motifs relationnels comme des listes d'atomes, i.e. $[l_1, \dots, l_n]$. Si C est un motif relationnel et l un littéral, on représente par $[C, l]$ le motif relationnel obtenu en ajoutant l après le dernier littéral de C . Etant donné une graine $g \in O$ (choisie aléatoirement dans notre cas), et pour chaque symbole de prédicat $p \in \mathcal{B}$, l'algorithme 1 calcule pour tous les littéraux l_i instances de p dans g les littéraux généralisés candidats de façon à générer et tester (noté par $\rho(Bottom, l_i)$),

Algorithm 1 Calcule *Bottom*, approximation de $lgg(O)$

Require: O un ensemble d'observations, \mathcal{B} : un biais de langage, une fonction de score *Score*

Ensure: Un motif relationnel *Bottom* qui θ -subsume toutes les observations de O , et de taille $\leq k * |g|$

```

1:  $g \leftarrow random\_choice(O)$ ;  $Bottom \leftarrow \emptyset$ 
2: for each  $p \in \mathcal{B}$  do
3:   for each  $l_i$  instance de  $p$  dans  $g$  ( $l_i \in g$ ) do
4:      $Cands \leftarrow k$ -meilleurs de  $\rho(Bottom, l_i)$  pour
       Score
5:     for each  $lg_i \in Cands$  do
6:       if  $[Bottom, lg_i]$  couvre toutes les observa-
       tions de  $O$  then
7:          $Bottom \leftarrow [Bottom, lg_i]$ 
8:       end if
9:     end for
10:  end for
11: end for
12:  $Bottom \leftarrow reduce(Bottom)$ 
13: return Bottom

```

Cet opérateur ρ prend en argument la généralisation courante *Bottom* et le littéral courant l_i pour engendrer des littéraux généralisés candidats à l_i . Par exemple, ρ peut contrôler pour les littéraux généralisés de l_i le nombre de variables, le nombre maximum de nouvelles variables (n'apparaissant pas déjà dans *Bottom*), etc. A cette étape, certains littéraux de $\rho(Bottom, l_i)$ peuvent être en relation de généralité (voir exemple). Le cardinal de $\rho(Bottom, l_i)$ pouvant être élevé, l'algorithme sélectionne les k meilleurs candidats pour la fonction *Score*. Un littéral $lg \in \rho(Bottom, l_i)$ intègre *Cands* si $[Bottom, lg]$ couvre toutes les observations de O . La fonction de score définit les critères de tri permettant de sélectionner les k meilleurs candidats. Elle prend en compte différents éléments syntaxiques (par exemple, le nombre d'occurrences des variables dans $[Bottom, lg]$). On peut également si nécessaire dès cette étape sélectionner en priorité les littéraux candi-

ats permettant de rejeter des observations de U_{-c} . Une fois les k -meilleurs littéraux généralisés candidats pour l_i identifiés, ils sont ajoutés de façon gloutonne à *Bottom* si $[Bottom, l_g]$ couvre toutes les observations de O .

Bottom est finalement réduite [17] pour ne garder *in fine* que les littéraux maximale-ment spécifiques.

Exemple 6. Reprenons l'exemple 1 et calculons le motif relationnel *Bottom* associé aux deux observations o_1 et o_2 . On suppose que ρ n'engendre que les littéraux généralisés qui contiennent au plus une variable (nouvelle ou apparaissant déjà dans *Bottom*). Commençons par le symbole de prédicat p le plus fréquent dans les deux observations et supposons que o_1 soit la graine. Soit la première instance de p dans o_1 , $p(1, 2)$. $\rho(Bottom, p(1, 2)) = \{p(1, 2), p(X, 2), p(1, Y)\}$. $p(1, 2)$ ne couvre pas o_2 , $p(X, 2)$ et $p(1, Y)$ couvrent tous les deux o_1 et o_2 et sont ajoutés à *Bottom*. La substitution θ pour o_1 devient $\{X/1, Y/2\}$. $\rho(Bottom, p(2, 3)) = \{p(2, 3), p(Y, 3), p(Y', 3), p(2, Z)\}$. $p(2, 3)$ ne couvre pas o_2 , $p(1, Y), p(Y, 3)$ ne couvrent pas o_2 , on ajoute donc à *Bottom* $p(Y', 3), p(2, Z)$ et $\{Y'/2, Z/3\}$ à θ . De la même manière, on ajoutera à *Bottom* $r(Y)$ et $q(3)$. On obtiendra finalement $Bottom = \{p(X, 2), p(1, Y), p(Y', 3), p(2, Z), r(Y), q(3)\}$, qui n'a pas besoin d'être réduite et qui est la lgg exacte de o_1 et o_2 , avec la substitution $\theta = \{X/1, Y/2, Y'/2, Z/3\}$ pour la graine o_1 .

Le motif relationnel *Bottom* est la borne inférieure de l'espace de recherche pour les explications communes minimales de O , que nous décrivons dans la section suivante.

5.2 Construction d'explications dynamiques

L'algorithme 1 construit un motif relationnel *Bottom* qui est une approximation de la lgg(O) (i.e., $Bottom \preceq_{\theta} lgg(O)$). Nous construisons maintenant des explications communes à O comme des sous-ensembles (au sens de \subseteq) corrects de *Bottom* (voir def. 4). Le premier système de fouille de motifs relationnels est Warmr [7], qui construisait des motifs relationnels fréquents appartenant à un biais de langage. Depuis, d'autres travaux se sont également intéressés à la fouille de motifs relationnels, notamment clos [9]. On souhaite ici résoudre un problème un peu différent, celui d'extraire de *Bottom* – motif relationnel – l'ensemble de ses sous-ensembles minimaux (sous \subseteq) et corrects. Par correct, nous entendons que si toutes les observations de O ont la classe c (dérivée grâce au classifieur D), alors les motifs relationnels recherchés ne devront couvrir aucune observation de l'ensemble U_{-c} des observations associées à une classe autre que c . Si un motif relationnel m couvre de telles observations, nous nommerons cet ensemble d'observations l'ensemble critique de m . Dans la suite, nous appellerons ces observations *critiques*. En d'autres termes, nous cherchons à identifier la borne G d'un espace des versions dont la borne inférieure est *Bottom* [13], le tout dans un langage relationnel.

Cette tâche a été étudiée dans le cadre de la fouille de données booléennes [21]. Dans cet article, les auteurs pro-

posent un algorithme descendant dans lequel un motif mg est spécialisé en ajoutant un attribut booléen l si $[mg, l]$ rejette au moins une observation critique de mg . L'ajout de l à mg est effectué si aucun des sous-ensembles de $[mg, l]$ qui contient l ne rejette les mêmes observations critiques.

Nous adaptons dans l'article cet algorithme à la recherche de motifs relationnels sous-ensembles (pour \subseteq) de *Bottom* et corrects. Cette adaptation n'est pas triviale pour les raisons suivantes. Lorsque l'on spécialise un motif relationnel mg afin de rejeter de nouvelles observations critiques, il peut être nécessaire d'ajouter pour ce faire des littéraux qui ne permettent pas dans un premier temps d'écarter de nouvelles observations critiques mais qui introduisent de nouvelles variables (objets) portant une information discriminante (voir [18] pour une des premières discussions à ce sujet). De multiples stratégies de *lookahead* ont été proposées dans ce cadre toutes fondées sur des biais de recherche *ad-hoc*. Nous proposons ici une stratégie originale qui s'appuie sur la structuration de *Bottom* en locales.

Définition 1. (d'après [5]). Soit $VarBottom$ l'ensemble des variables de *Bottom*, soient X_1 et X_2 deux variables de $VarBottom$. X_1 touche X_2 si les deux variables apparaissent dans le même littéral, et X_1 influence X_2 si elle touche X_2 ou si elle touche au moins une variable X_i qui influence X_2 . La locale d'une variable X est l'ensemble des littéraux qui contiennent X ou au moins une variable influencée par X .

Intuitivement, une locale est un ensemble maximal de littéraux qui partagent des variables. Les instanci-ations possibles d'une variable sont donc contraintes uniquement par des variables de sa locale. Un fait est une locale de taille un.

Exemple 7. La lgg de l'exemple 1 contient 5 locales $\{p(1, X)\}, \{p(X', Y'), r(Y')\}, \{p(X'', 3)\}, \{q(3)\}, \{p(2, Y'')\}$

On note dans la suite par $couverture(m, O)$ où m est un motif relationnel et O un ensemble d'observations l'ensemble des observations $o_i \in O$ telles que $couvre(m, o_i)$. Les algorithmes 2, 3 et 4 étendent l'algorithme décrit dans [21] au calcul de motifs relationnels corrects. L'algorithme 4 recherche dans une locale un sous-ensemble minimal de littéraux qui permet de rejeter au moins une observation de l'ensemble critique de mg . Un tel sous-ensemble de littéraux peut-être géré comme un attribut booléen dans l'algorithme de [21]. L'algorithme 3 permet d'identifier quand la recherche doit s'arrêter : soit mg est correct (l'ensemble critique de mg , NCE est vide), soit il n'existe plus de solution si la formule maximale atteignable n'est pas correcte (ligne 2 de l'algorithme 3).

Par définition de *Bottom*, mg couvre toutes les observations de O , et donc de tout sous-ensemble de O . Comme mg et LK ne partagent aucune variable (par définition d'une locale), si LK_i et LK_j sont deux locales de *Bottom*, $i \neq j$, $couverture([LK_i, LK_j], NCE) = couverture(LK_i, NCE) \cap couverture(LK_j, NCE)$.

Exemple 8. Toujours dans le cadre de l'exemple 1, supposons que o_1 et o_2 soient de classe c et qu'il n'existe qu'un

Algorithm 2 Common_Explanations($Bottom, U_{\neg c}$)

Ensure: MG : ensemble de sous-ensembles minimaux et corrects de $\subseteq Bottom$ (explications communes pour tout $o_i \in O$)

- 1: $LLK \leftarrow compute_locales(Bottom)$
 - 2: **return** $mings(\emptyset, LLK, U_{\neg c})$
-

Algorithm 3 $mings(mg, LLK, NCE)$

Require: mg : motif minimal courant, NCE : ensemble critique de mg , LLK : locales à explorer

Ensure: MGF : ensembles des motifs minimaux et corrects de $Bottom$

- 1: $MGF \leftarrow \emptyset$
 - 2: **if** $mg \cup LLK$ est correct **then**
 - 3: **if** mg est correct ($NCE = \emptyset$) **then return** mg
 - 4: **end if**
 - 5: $LK \leftarrow head(LLK)$
 - 6: $nLLK \leftarrow tail(LLK)$
 - 7: **for** tous les $M = maxGen(LK, NCE)$ **do**
 - 8: \triangleright voir alg.4
 - 9: $emg \leftarrow [mg, M]$
 - 10: $rNCE \leftarrow couverture(emg, NCE)$
 - 11: $MGLK \leftarrow mings(emg, nLLK, rNCE)$
 - 12: $MGF \leftarrow MGF \cup MGLK$
 - 13: **end for**
 - 14: $MGWLK \leftarrow mings(mg, nLLK, NCE)$
 - 15: $MGF \leftarrow check_min(MGF \cup MGWLK)$
 - 16: **end if**
 - 17: **return** MGF
-

Algorithm 4 $maxGen(LK, NCE)$

Require: LK : une locale de $Bottom$, NCE : observations critiques de mg

Ensure: MG ensembles des minimaux de LK qui rejettent au moins une obs. critique de NCE

- 1: **if** LK ne rejette aucun exemple de NCE **then**
 - 2: **return** \emptyset
 - 3: **end if**
 - 4: $MG \leftarrow \emptyset$
 - 5: **for** pour tous les sous-ensembles mg_i de LK **do**
 - 6: $rNCE \leftarrow couverture(mg_i, NCE)$
 - 7: **if** $|rNCE| < |NCE|$ et mg_i est minimal **then**
 - 8: $MG \leftarrow MG \cup mg_i$
 - 9: **end if**
 - 10: **end for**
 - 11: **return** MG
-

exemple o_3 de classe $\neg c$: $\{p(2, 4), r(2), p(2, 3), q(3)\}$. Si on ordonne les locales par rejet des observations de classe $\neg c$, $\{q(3)\}$, $\{p(2, Y'')\}$ et $\{p(X'', 3)\}$ sont immédiatement rejetées. $\{p(1, X)\}$ et $\{p(X', Y'), r(Y')\}$ sont en revanche chacune des minimaux corrects ($p(X', Y')$ et $r(Y')$ couvrent o_3 alors que $p(X', Y'), r(Y')$ rejette o_3 et est donc un minimal correct).

Proposition 4. L'algorithme 2 est correct et complet.

La preuve est fournie en annexe ².

6 Exemple développé

Dans cette section nous considérons une donne du jeu simplifié et deux ensembles de trajectoires. Dans chaque cas nous calculons

- un ensemble de clauses D couvrant les trajectoires optimales et aucune trajectoire non-optimale³.
- pour chaque clause l'ensemble des explications communes minimales des trajectoires couvertes par la clause. Pour certaines explications nous donnons une preuve informelle de ce qu'elle implique l'optimalité des trajectoires.

La donne est la suivante

W 8 9 11 N 3 4 5 12 E 6 7 S 2 10 13 14

Le problème de décision du déclarant posé au joueur artificiel Noo est considéré comme un MDP déterministe. Les actions sont les cartes jouées par le déclarant en North ou South aux différents instants du jeu. La transition d'un état du jeu au suivant est déterministe et connue de Noo ce qui signifie en particulier que Noo a un modèle de la défense : il sait à tout moment du jeu quelle carte jouera le défenseur East ou West. On décrira une trajectoire par les cartes jouées par les déclarants, c'est-à-dire les actions, et celles jouées par les défenseurs permettant les transitions d'un état au suivant. La récompense obtenue $r(s, a)$ quand le déclarant joue une carte a à l'instant t dans l'état s est 1 si le pli courant est gagné entre t et $t + 1$. Chaque instant t marque le moment où le déclarant (North ou South) doit jouer.

Exemple 9. Nous considérons le début d'une trajectoire commençant en l'état initial s_1 après que West ait joué le 8 au début du premier pli :

W8 (t1) N12 E6 (t2) S2 (t3) S13 W9

Quand North joue le 12 en t1 la récompense $r(s1, 12)$ est nulle mais on a $r(s2, 2) = 1$ car après que South ait joué le 2 le déclarant a fait le premier pli.

On peut associer à chaque trajectoire la récompense totale qui lui est associée, c'est à dire le nombre de plis fait par le déclarant à la fin du jeu.

2. L'annexe est consultable à l'adresse suivante : https://github.com/Malickick/PFIA23-trajectoires/blob/main/annexe_PFIA_23_trajectoires.pdf

3. en utilisant cLear, un programme d'apprentissage relationnel développé par NukkAI

Notons que si dans l'état courant s le déclarant dont c 'est le tour de jouer a plusieurs cartes *consécutives* dans sa main, jouer l'une ou l'autre ne change pas le déroulement de la trajectoire. Formellement deux cartes a et a' dans la main d'un joueur à l'état s sont s -consécutives si il n'existe pas en s de carte a'' dans la main d'un autre joueur telle que $a < a'' < a'$ ou $a' < a'' < a$. Par exemple que North joue 5 ou 3 en s_1 n'affecte pas la suite du jeu. Par la suite dans l'illustration d'un sous-arbre de trajectoires de racine s on considère équivalentes les cartes s -consécutives pour North ou South. L'arbre représenté est alors un arbre de trajectoires dites *abstraites* dans lequel on a un seul arc pour plusieurs actions équivalentes. Nous ne représentons que la partie *utile* de la trajectoire se terminant en t_7 , temps où le défenseur n'ayant plus de cartes, l'optimalité ou non de la trajectoire st acquise.

6.1 North joue son honneur en t_1

Ci-dessous nous considérons le sous-arbre des trajectoires commençant en l'état s_2 après que North ait joué 12 (action N12) et East ait répondu 6. L'action N12 conduit a une q -valeur optimale de 3 plis et nous cherchons ici à répondre à la question "comment atteindre cette q -valeur optimale à partir de l'état s_2 ?". L'ensemble des trajectoires optimales se représente par deux trajectoires abstraites qui ne diffèrent que par la carte 2 ou 10 jouée par South au temps t_2 et que nous représentons de manière compacte ci-dessous :

W8	N12	E6	S(2 ou 10)
N3-4-5	E7	S13-14	W9
S13-14	W11	N3-4-5	E0

On remarque que dans ces deux trajectoires abstraites optimales South joue une petite carte (≤ 10) au premier pli remporté par North et ses deux honneurs 13 (Roi) et 14 (As) aux plis 2 et 3. Ici l'univers est l'ensemble des chemins du sous-arbre de racine s_2 , composé des 24 trajectoires optimales de U_{opt} , résumées en les deux trajectoires abstraites ci-dessus et des 120 non optimales de U_{notOpt} . On construit à l'aide de *cLear* un modèle pour la cible *optimale* à partir de U et on obtient une clause unique :

$$opt \leftarrow nbSmallCards(1, B, [3, 7]).$$

Tous les prédicats utilisés sont définis dans l'annexe. Cette clause permet de dériver *opt* pour toute trajectoire commençant en t_2 après W8 N12 et dit qu'il existe un joueur B dont le nombre de cartes de valeur inférieure ou égale à 10 atteint 1 en t_3 puis reste constant jusqu'à la fin de la trajectoire utile au temps t_7 .

On construit alors $lgg(U_{opt})$ (Algorithm 1) et on cherche les explications communes minimales de U_{opt} (Algorithm 2). Parmi celles-ci on trouve les deux explications communes minimales suivantes dont nous montrons l'optimalité.

1. $x_1 = nbSmallCards(1, south, [3, 7])$

Le corps x_1 d'unique clause du modèle est complètement instancié et est une explication commune minimale des 24 trajectoires couvertes par la clause. Pour montrer qu'elle conduit à l'optimalité on raisonne sur la partie en cours. x_1 implique que South a

joué une petite carte en t_2 , puisque en t_3 son nombre de petites cartes a diminué, et c'est donc North qui fait le premier pli avec le 12. Puis, ce nombre ne diminuant plus jusqu'au 4ème pli, South joue ses deux honneurs 13 et 14 aux plis 2 et 3, qu'il remporte (East et West n'ont pas de cartes plus fortes). Le déclarant remporte donc les 3 plis et la trajectoire est optimale.

2. $x_2 = \exists B action(B, 5), honor(B)$ qui dit que le déclarant joue un honneur en t_5 début du 3ème pli.

L'explication commune alternative x_2 conduit l'interlocuteur à un raisonnement plus élaboré pour conclure à l'optimalité. North n'ayant plus d'honneur en t_5 c'est forcément South qui joue l'honneur B (donc 13 ou 14) en t_5 et fait le pli 3. South jouant en t_5 cela implique que South ou East a gagné le second pli : en effet si North ou West gagnait le pli 2, South jouerait au pli 3 en t_6 et non en t_5 . Comme au début du pli 2 l'unique carte de East (le 7) est plus petite que celles de West (9 ou 11) East ne peut pas faire ce pli et c'est donc South qui a gagné le pli 2. Finalement, North ayant fait le premier pli (avec le 12) et South les deux suivants, la trajectoire est optimale.

6.2 North joue une petite carte en t_1

Si au temps t_2 North joue une petite carte, ici le 3 (mais 3 4 et 5 sont équivalentes car s_1 -consécutives) il y a plusieurs trajectoires abstraites optimales représentées Figure 1 dans un arbre dont les feuilles, numérotées de 1 à 10 à partir de la gauche, représentent les trajectoires abstraites optimales commençant par W8 N3. Les actions non optimales, possibles mais conduisant à des trajectoires non optimales, sont marquées par un arc conduisant à une feuille vide.

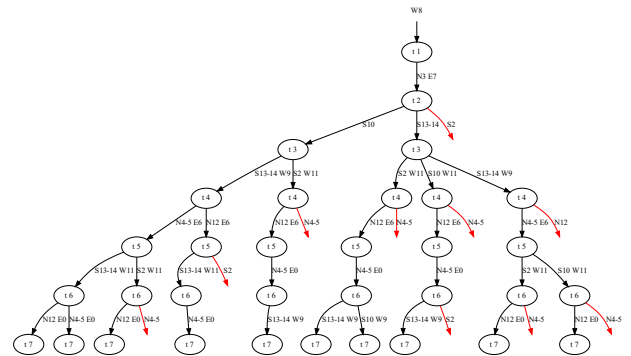


FIGURE 1 – Arbre de trajectoires abstraites optimales

On obtient cette fois quatre clauses pour le label *opt* par apprentissage sur les 40 trajectoires optimales et 104 non-

optimales :

$$\begin{aligned} \text{opt} &\leftarrow \text{playSmallestCard}(C, \text{south}, 3), \\ &\quad \text{willTakeTrickWithDominant}(12, \text{north}, F). \end{aligned} \quad (1)$$

$$\text{opt} \leftarrow \text{action}(A, 12, 6), \text{nbSmallCards}(A, 1, B, [1, 3]). \quad (2)$$

$$\text{opt} \leftarrow \text{nbHonors}(A, 1, B, [4, 5]). \quad (3)$$

$$\text{opt} \leftarrow \text{nbThreats}_h(A, 2, B, 0, [7, 7]). \quad (4)$$

Les trajectoires abstraites couvertes par les clauses sont dans l'arbre respectivement aux feuilles 5,6,7 pour la clause 1, aux feuilles 1,3,9,10 pour la clause 2, aux feuilles 1,2,4 pour la clause 3, et à la feuille 8 pour la clause 4. Avant de détailler quelques explications associées à la clause 3, notons qu'on trouve comme explication associée à la clause (1) une instanciation du corps de la clause avec $C = 2$ et $F = 4$: les clauses apprises sont maximale-ment générales, mais ce n'est pas le cas des explications qui sont des sous-ensemble de la lgg. Même chose pour la clause (2) où le joueur ayant un honneur entre t_4 et t_5 est $B = \text{West}$. Parmi les explications communes minimales de la clause 3 on trouve :

1. $\text{nbHonors}(1, \text{south}, [4, 5])$. Il s'agit du corps de la clause (3). Il dit qu'il y a 1 honneur à South exactement entre t_4 et t_5 . On sait donc a) l'un des honneurs (13-14) est donc joué en t_3 début du 2ème pli et b) l'autre en t_5 début du 3ème pli. De a) on déduit que South a fait le premier pli (East n'a pu gagner ce pli) et le second (13 et 14 sont dominants dès le début). De b) on déduit qu'il a fait le 3ème.
2. $\text{action}(10, 2), \text{action}(13, B), \text{action}(14, C)$. dit que South joue le 10 au premier pli, puis le 13 et le 14 aux deux autres plis de la trajectoire, la trajectoire est donc optimale. Ce qui donne un plan de jeu très directif au déclarant et est d'une interprétation bien plus simple que la clause (3) elle-même.
3. $\text{maxCardHand}(A, 2, \text{south}, [6, 7]), \text{nextDominant}_h(A, 12, \text{north}, [B, C]), \text{geq}(C, 4)$. Le premier atome dit qu'à partir de t_6 la plus grande carte de South est le 2 et donc que South a déjà joué le 10, le 13 et le 14 et a donc fait en particulier le pli 1. L'un de ces cartes est jouée en t_5 donc South a fait le pli précédent, le pli 2. Le deuxième et troisième atomes disent que 12 est la seconde carte dominante de North dans un intervalle se finissant en un $t \geq t_4$. On en déduit qu'en t_4 South n'a pas encore joué son deuxième honneur (13 ou 14), il jouera donc cet honneur en t_5 et fera le pli 3. Le déclarant ayant remporté les 3 plis la trajectoire est optimale.

La valeur de ces explications dépend de leur propos. S'il s'agit de voir la suite d'actions résumant ces trajectoires favorables, l'explication 2, la plus directe convient. Si le but est pédagogique, on peut voir les explications 1 et 3 comme des exercices poussant un apprenti-joueur à raisonner sur les actions de jeu et leurs conséquences.

7 Conclusion

Nous avons proposé dans cet article une formalisation et une proposition fondée sur des outils de Programmation Logique Inductive de la notion d'explication de l'étiquette partagée par un groupe d'exemples. Nous utilisons, de même que [20], la notion de voisinage de l'observation à expliquer (défini comme l'ensemble des observations couvertes par une même règle du classifieur D) en l'adaptant au cadre relationnel. Contrairement à [10], nous nous appuyons fortement sur les exemples dont nous disposons pour construire les explications, et pour ce faire, nous avons recours largement à des opérations d'apprentissage. Nous avons proposé une formalisation logique de la notion d'explications en ordre un d'un groupe d'observations et nous avons conçu et implémenté des algorithmes de calcul d'une borne inférieure de l'espace des explications qui s'approche d'un moindre généralisé du groupe à expliquer. Nous avons également proposé un algorithme de calcul de motifs minimaux relationnels à partir de cette borne inférieure. Enfin, nous avons pu mener de premiers tests de la méthode sur des problèmes d'analyse de trajectoires d'un jeu de Bridge simplifié.

Ces premières expérimentations nous ont montré que les explications calculées ici définissent a minima la notion intuitive d'explication : on suppose l'interlocuteur omniscient, et connaissant donc la sémantique du langage utilisé, et l'univers U des observations possibles et leurs labels (via la classifieur D). Dans la mesure où on lui fournit un ensemble de faits extraits d'une observation o suffisants, sachant U , pour en déduire le label, on suppose qu'il est en mesure de faire un travail d'interprétation/validation de cette explication, qui lui fournit des informations minimale nécessaire pour dériver le label de cette trajectoire.

Une première perspective de ce travail est la construction d'une théorie reliant les prédicats du langage, incluant une version intensionnelle de l'univers U . On pourra alors faire des déductions à partir des explications. Nous avons en particulier montré informellement dans notre étude de cas comment on conclut à partir d'une explication à l'optimalité d'un groupe de trajectoires. Par ailleurs les explications fournies peuvent être nombreuses et partiellement redondantes selon le langage et les contraintes de score utilisées pour le calcul de *Bottom*. Nous pensons utiliser une partie de cette théorie dans le calcul d'une distance entre explications et garantir ainsi une certaine diversité des explications. Une perspective majeure est le passage au cas où le joueur artificiel ne connaît pas les mains de l'adversaire. Du point de vue de l'optimalité on passe de la résolution d'un MDP à celle d'un MDP partiellement observé, donc de trajectoires de récompense maximale à des trajectoires dont l'espérance de la récompense totale est maximale. La question de ce qu'est une explication dans ce cadre est ouverte.

Remerciements

Nous tenons à exprimer notre gratitude envers NukkAI et l'ANRT pour le soutien et la confiance qu'ils ont placés en notre recherche. Egalement, nous tenons à remercier Domi-

nique Bouthinon pour son élaboration du programme cLear, qui a permis de réaliser les expérimentations présentées dans cet article, et Junkang Li qui a élaboré le système que nous avons utilisé afin de résoudre le MDP. Nous adressons enfin nos remerciements aux relecteurs de CNIA pour leurs précieuses suggestions et commentaires qui ont contribué à améliorer cet article.

Références

- [1] Allen, J.F. : Maintaining knowledge about temporal intervals. *Commun. ACM* **26**(11), 832–843 (1983)
- [2] Audemard, G., Bellart, S., Bounia, L., Koriche, F., Lagniez, J., Marquis, P. : On preferred abductive explanations for decision trees and random forests. In : Raedt, L.D. (ed.) *Proceedings of IJCAI'22*. pp. 643–650 (2022)
- [3] Audemard, G., Bellart, S., Bounia, L., Koriche, F., Lagniez, J., Marquis, P. : Sur le pouvoir explicatif des arbres de décision. In : *EGC*. vol. E-38, pp. 147–158. Editions RNTI (2022)
- [4] Blockeel, H., Raedt, L.D., Jacobs, N., Demoen, B. : Scaling up inductive logic programming by learning from interpretations. *DMKD'99* **3**, 59–93 (1999)
- [5] Cohen, W.W., Jr., C.D.P. : Polynomial learnability and inductive logic programming : Methods and results. *New Gener. Comput.* **13**(3&4), 369–409 (1995)
- [6] Darwiche, A., Hirth, A. : On the reasons behind decisions. In : L (ed.) *ECAI'20. Frontiers in Artificial Intelligence and Applications*, vol. 325, pp. 712–720 (2020)
- [7] Dehaspe, L. : Frequent pattern discovery in first-order logic. *AI Commun.* **12**(1-2), 115–117 (1999)
- [8] Echenim, M., Peltier, N. : A calculus for generating ground explanations. In : *IJCAR'12*. vol. 7364, pp. 194–209. Springer (2012)
- [9] Garriga, G.C., Khardon, R., Raedt, L.D. : Mining closed patterns in relational, graph and network data. *Ann. Math. Artif. Intell.* **69**(4), 315–342 (2013)
- [10] Huang, X., Izza, Y., Ignatiev, A., Marques-Silva, J. : On efficiently explaining graph-based classifiers. In : *Proceedings of KR'21*. pp. 356–367 (2021)
- [11] Inoue, K. : Consequence-finding based on ordered linear resolution. In : *IJCAI'91*. pp. 158–164. Morgan Kaufmann (1991)
- [12] Marquis, P. : Extending abduction from propositional to first-order logic. In : *FAIR*. vol. 535, pp. 141–155. Springer (1991)
- [13] Mitchell, T.M. : Generalization as search. *Artif. Intell.* **18**(2), 203–226 (1982)
- [14] Mörchen, F., Fradkin, D. : Robust mining of time intervals with semi-interval partial order patterns. In : *SDM*. pp. 315–326. SIAM (2010)
- [15] Muggleton, S., Raedt, L.D. : Inductive logic programming : Theory and methods. *J. Log. Program.* **19/20**, 629–679 (1994)
- [16] Nienhuys-Cheng, S.H., Wolf, R.D., de Wolf, R. : *Foundations of Inductive Logic Programming*. Springer-Verlag New York, Inc. (1997)
- [17] Plotkin, G.D. : A note on inductive generalization. *Machine Intelligence* **5**, 153–163 (1970)
- [18] Quinlan, J.R., Cameron-Jones, R.M. : FOIL : A mid-term report. In : *ECML'93. Lecture Notes in Computer Science*, vol. 667, pp. 3–20. Springer (1993)
- [19] Rabold, J., Siebers, M., Schmid, U. : Generating contrastive explanations for inductive logic programming based on a near miss approach. *Mach. Learn.* **111**(5), 1799–1820 (2022)
- [20] Ribeiro, M.T., Singh, S., Guestrin, C. : "Why Should I Trust You ?" : Explaining the predictions of any classifier. In : *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 1135–1144 (2016)
- [21] Soulet, A., Rioult, F. : Exact and approximate minimal pattern mining. In : *Advances in Knowledge Discovery and Management - Volume 6 [Best of EGC 2014-2015]*. *Studies in Computational Intelligence*, vol. 665, pp. 61–81 (2015)
- [22] Soulet, A., Rioult, F. : *Exact and Approximate Minimal Pattern Mining*, pp. 61–81. Springer International Publishing (2017)
- [23] Sutton, R.S., Barto, A.G. : *Reinforcement Learning : An Introduction*. MIT Press (1998)