Explaining the optimal trajectories

C. Rouveirol[2], M. Kazi Aoual[1,2], H. Soldano[1,2,3], and V. Ventos[1]

[1] Nukkai,France
[2] UMR CNRS 7030 Institut Galilée – Université Sorbonne Paris Nord, LIPN
[3] UMR CNRS 7205 Museum National d'Histoire Naturelle, ISYEB

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: Abductive Explanation, Inductive Logic Programming, Markov Decision Process

1 Introduction

We are interested here in explanations for the classification of an observation by a logical classifier when the observation is a tree of possible future actions and resulting states whose branches are called trajectories. The work is motivated by a scenario in which a machine plays a simplified bridge game in the declarer position against a program playing the defender, and must answer at any point in the gameplay queries of the type "How does the chosen action lead to win a maximum number of tricks ? “. Technically, when the artificial player knows the cards its adversaries have in hand, it optimizes the total reward obtained at the end of a trajectory by solving a Markov Decision Process (MDP) [21].

Expected explanations for choosing action a in state s refer then to a set of possible optimal trajectories starting from (s, a). For that purpose, we need a way to regroup trajectories in a limited number of groups, each made of trajectories that may be explained in the same way. Therefore, we have to define and search for *common explanations* of optimality for a group of trajectories. A first step is to select groups of trajectories guaranteed to have at least one common explanation. Given a(state, action) pair (s, a), we propose to build such groups by learning a rule-based classifier concluding on the optimality, or not, of any trajectory starting from (s, a). Our definitions are in line with previous works on abductive explanations of the label assigned to an observation by a logical classifier [9, 1, 2, 5] that we have adapted or extended for our purpose in several directions. First consider that to any optimal (state, action) pair (s, a) is associated the set U of all the *possible* trajectories starting from (s, a). U is called the *universe* associated to (s, a) and we assume that we have a classifier D to label the trajectories in U as optimal or non-optimal. Explanations are adapted in the following ways:

- A trajectory is described in first order logics as the set of ground literals that hold for the trajectory and an explanation for a single trajectory is a subset of these literals.
- An explanation depends on the classifier D and the trajectory description, as expected, but also depends on the universe, introduced as a formula whose models are the trajectories.
- A *common explanation* for a set of trajectories is then defined as an existentially quantified conjunction [11], we further call a *relational motif*. We will associate to a group O of trajectories the most specific relational motif $\text{lgg}(O)$, that hold for all the trajectories in the group and then define common explanations as parts of $\text{lgg}(O)$.

Technically, we solve the problem of enumerating minimal common explanations by addressing two sub-problems; i) Computing a least general generalization (lgg) for a group of observations with a given label under θ-subsumption and ii) Searching for all minimal subsets of a lgg that does not θ-subsumes any observation with a different label. These problems have been studied in different contexts. Since [15] the construction of lggs has been widely used in particular in ascending ILP methods. To limit the lgg size, which is mandatory for our practical purpose, we rather consider building an approximated lgg satisfying a set of declarative constraints. The search for minimal subsets satisfying covering contraints has been studied in many contexts, in particular in data mining (see for instance [20]) but not often when considering relational data in which different useful algorithmic properties are lost [8]. We propose in Section 5 algorithms to solve these problems. In Section 6 we discuss the case study and the explanations we obtain.

2 Scenario

We consider a simplified bridge card game with a single color which 13 cards (with values from 2 to 14) are distributed between the players in a *deal*. The number of cards in each player's hand is not fixed. We consider the bidding has ended by a contract requested by North and which opposes the North-South pair (the declarer NS) to the West-East pair (the defender WE).

In the scenario below, we call our artificial player *Noo* and his interlocutor *M. X*. *M. X* asks for explanations of the decisions taken by *Noo* during the game. In the general scenario, the hands *WE* are unknown to the declarer *NS*, but in this article we consider the simplified scenario in which *Noo* knows the *WE* hands. We suppose that whenever *Noo* knows the current state e he/she also knows for each possible action a which card the opponent will play. *Noo* then solves the general MDP only considering the declarer actions: to any possible *WE* cards repartition is associated a starting state, and for any (state t action a) pair accessible from a starting state, the maximal total reward $q(s, a)$, i.e. the maximal number of tricks the declarer win by playing a in state s and further playing optimally all along the trajectory. Action a is optimal in state s if it maximises $q(s, a)$. The scenario begins after West has played the first card and the hand of North has been unveiled:

1. *Noo* chooses an optimal action a in current state s
2. *M. X* asks for explanations on how playing a can further result in winning $q(s, a)$ tricks
3. *Noo* then builds a model D to distinguish among the trajectories starting from s, a the *optimal* trajectories ending in winning $q(s, a)$ tricks from the others. *Noo* uses D to define groups of optimal trajectories within which optimality of a has common explanations.
4. *Noo* proposes minimal common explanations to answer the request.
5. *Noo* plays a which leads to a new state and the scenario goes on at step 1.

3 Notations in relational representations

We handle in this paper Datalog languages (i.e. First Order Logics with no function symbols other than constants). The only terms are *constants* and *variables*. Constants are either numbers or atoms starting with a lowercase letter. For instance, cards can be represented by integers in the range [2..14] and players by the four constants *west*, *north*, *east*, *south*. Other terms are variables, identified by symbols starting with an uppercase letter (X, Y, $Card$, ...). The vocabulary $\mathcal{V}$ of a Datalog program P is the set of its constants and predicate symbols. A *literal* is a predicate applied to terms A *fact* is a ground literal (without variables). For instance, the literal $small\_card(C)$ states that variable C is a small card (i.e. between 2 and 10), whereas $honor(12)$ is a fact stating that 11 (representing a jack) is an honor. In the following, we use the usual notation p/N where p is a predicate symbol and N is its arity (i.e. $small\_card/1$ is the predicate symbol $small\_card/1$ with a single argument).

In the context of Inductive Logic Programming, two types of formulas are handled : *definite clauses* and *existentially quantified conjunctions*. A *clause* is a disjunction of literals universally quantified $\forall[h_1 \vee \ldots \vee h_m \vee \neg b_1 \vee \ldots \vee \neg b_n]$ or equivalently $\forall[(h_1 \vee \ldots \vee h_m) \leftarrow (b_1 \wedge \ldots \wedge b_n)]$ where $h_1, \ldots, h_m$ is a disjunction of positive literals (refered to as the *head* of the clause) and $b_1, \ldots, b_n$ is the conjunction of literals forming the *body* of the clause.

A *definite* clause is a clause with exactly one positive literal. If the head of the clause is a literal without argument, the body of this clause, refered to as a *relational motif* in the following, is an existentially quantified conjunction of literals. As one classicially omit the $\forall$ quantifier when it is clear from the context that we handle clauses, we omit the $\exists$ quantifier when we handle relational motifs. In the rest of this section, we will adapt definitions initially introduced by [15] for clauses to relational motifs that we mainly study in this paper.

Given a vocabulary $\mathcal{V}$, the *Herbrand universe* of $\mathcal{V}$ is the set of ground terms built on $\mathcal{V}$, the *Herbrand base* is the set of ground facts built on the Herbrand universe and predicate symbols of $\mathcal{V}$. A *Herbrand interpretation* of a set of FOL formulas P built on $\mathcal{V}$ (here a Datalog program or a set of relational motifs) is a subset of the Herbrand base of $\mathcal{V}$.

The generality relationship between two clauses classically used in ILP [13] is the θ-subsumption relationship [15] that we adapt here to relational motifs.

Definition 1 *A relational motif G θ-subsumes a relational motif S (denoted $G \preceq_\theta S$) if and only if (iff) there exists a substitution θ such that $G.\theta \subseteq S$.*

Plotkin also introduced the notion of *maximally specific* or *least general* generalisation denoted *lgg* of two clauses, that we reformulate hereafter for relational motifs.

Definition 2 *A relational motif S is the* most specific generalisation *of a set of relational motifs O (denoted by $S = lgg(O)$) iff $S \preceq_\theta o_i$ for all $o_i \in O$ and for all G_j such that $G_j \preceq_\theta o_i$ for all $o_i \in O$, $G_j \preceq_\theta S$. The* lgg *of two relational motifs C et D is unique and computed in time $\mathcal{O}(|C||D|)$ [14].*

In the following, an *observation* is a Herbrand interpretation of the vocabulary $\mathcal{V}$ (see (the learning from interpretations framework [3] or from multiple interpretations [8]). The *coverage* relationship between a relational motif C and an observation o , denoted $covers(C, o)$ iff there exists a substitution θ such that $C \preceq_\theta conj(o)$ where $conj(o)$ is the ground relational motif corresponding to o (the conjunction of all facts in o). For the sake of simplicity, we identify in the following the observation o and the corresponding conjunction $conj(o)$. The *lgg* of a set of observations O is defined by $lgg(\{o_i | o_i \in O\})$ (and in particular, $lgg(\{o\}) = o$).

Example 1 *Let $\mathcal{V}$ be a vocabulary with four constants $\{1, 2, 3, 4\}$ and three predicate symbols $\{p/2, r/1, q/1\}$. Let us assume that we have two observations o_1 et o_2, $o_1 = \{p(1, 2), r(2), p(2, 3), q(3)\}$ and $o_2 = \{p(1, 3), q(3), p(2, 4), r(4)\}$. The* lgg *of o_1 and o_2 is: $\exists\ p(1, X), p(X', Y'), r(Y'), p(X'', 3), q(3), p(2, Y'')$ with matching substitutions $\theta_1 = \{X/2, X'/1, Y'/2, X''/2, Y''/3\}$ and $\theta_2 = \{X/3, X'/2, Y'/4, X''/1, Y''/4\}$. Note that the lgg of o_1 and o_2 is longer (in the number of literals) than both o_1 and o_2.*

4 Explanations

Explanations considered here are *abductive explanations* investigated in recent works to explain in particular the label assigned to some observation o entered as input of a decision tree[9, 1, 2, 5]. In these works an observation is represented by the values taken by a set of attributes, i.e. by a set o of (attribute, value) pairs. An abductive explanation for classifying o into label c with classifier D is then a minimal subset of o which is sufficient to classify o into c. For our purpose we need to extend and change this definition in various ways.

For our purpose an observation o is an Herbrand interpretation of some datalog language (see Section 3) that we represent as a conjunction of litterals, The classifier D is as well a first order formula of this language. An explanation for assigning the class label c to an observation is then defined as a ground clause[17] whose head is the label. Accordingly we define an explanation as the body of such a clause, i.e. a ground relational motif. However when turning to common explanations, we rather consider general relational motifs, following the general definition of abdictive explanations in first order logics first proposed by P. Marquis [11] and then further investigated from an operational point of view [10, 7].

In what follows an observation is a subset o of positive literals, excluding class labels in C, that hold for this observation and that we may also represent as a conjunction, i.e. a ground relational motif..We write $t \subseteq o$ to state that the ground relational motif t is more general than the ground relational motif o. Furthermore, we consider that the set of possible observations in the problem at hand, we call a *universe*, is only part of the whole set of interpretations. In the definitions below this subset is represented as a formula U, that does not mention labels, whose models are the elements of the universe. A classifier D is then a formula that assigns to o one label c from C and we write $o, D \models c$. Our knowledge U, D on the problem in hand is then divided into a part U, the universe, restricting which observations o are allowed and a part D allowing to infer the label of any observation. This leads to the following definition, adding U to the usual definition of an abductive explanation of the assignation of a label to an observation:

Definition 3 *An explanation of the assignation of label c to observation o with respect to classifier D and universe U is a subset t of o such that: $t, U, D \models c$*

If for any $t' \subset t$ we have $t', U, D \not\models c$ then t is a minimal explanation of assignation of c to o w.r.t D and U.

In the example below we illustrate minimal explanations and see that by adding as a constraint the universe U of possible observations, we obtain different explanations from those obtained when omitting U.

Example 2 *We consider labels $+$ and $-$.D is a set of definite clauses concluding on $+$ and we consider that whenever $o, D, U \not\models +$ then o is classified as $-$.*

Let p and r be two unary predicates and $\{p(1), p(2), r(1), r(2)\}$ be the Herbrand basis. Consider observation o_1 and classifier D defined as follows:

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

Let us first suppose that any description is allowed, i.e. $U = true$. The minimal explanations for classifying o_1 as $+$ are $p(1), r(1)$ and $p(2)$). This is because we have $p(X), r(X)(X/1) = p(1), r(1) \subseteq o_1$, and $p(2) \subseteq o_1$.

Now consider $U = \{p(X) \leftarrow r(X); p(1) \vee p(2)\}$. As observations have to satisfy U, from $r(1), U$ we deduce $p(1)$ and from $p(1), r(1), D$ we deduce $+$. As a consequence $p(1), r(1)$ though it still explains the $+$ label for o_1 is not minimal anymore as $r(1)$ also explains the label. The set of minimal explantations is therefore $\{r(1); p(2)\}$.

We now define a *common explanation* of the assignation of a same label c to a set of observations O as a relational motif. For that purpose we first search for what is common to these observations, then considering candidate explanations as part of what is common to these observations.

Definition 4 *A* common explanation *of the assignation of same label c to observations in a group O with respect to classifier D and universe U is a relational motif $e \subseteq \mathrm{lgg}(O)$ such that $e, U, D \models c$. If for any $e' \subset e$ we have $e', U, D \not\models c$ then e is a minimal common explanation for O.*

Example 3 *Building on example 2 we consider the observation $o_2 = \{p(1), p(2), r(2)\}$. with same label $+$ as o_1. Let $O = (\{o_1, o_2\}$, when only considering positive literals we obtain $lgg(O) = p(1), p(2), r(X)$ as $lgg(O)(X/1) \subseteq o_1$ and $lgg(O)(X/2) \subseteq o_2$.*

From U and $r(X)$ we infer $p(X)$ and from $r(X), p(X)$ and D we infer $+$. We also have that from $p(2)$ and D we infer $+$. The minimal common explanations for $\{o_1, o_2\}$ are then $r(X)$ and $p(2)$.

We also have the following property relating explanations and common explanations:

Proposition 1. *Let e be a minimal common explanation for E, then for any $o \in O$, an substitution θ such that $e.\theta \subseteq o$ then $e.\theta$ is an explanation, not necessarily minimal for observation o.*

In what follows we consider that U is known through its set of models $M(U)$ which is partitioned according to the labels assigned by D in $\{U_c \subseteq M(U) \mid c \in C\}$. A common explanation e of assignation of label c to $O \subseteq U_c$ is then such that e does not cover any observation belonging to $U_{notc} = M(U) \setminus U_c$.

We may then build the minimal common explanations for groups of observations with label c from the partition $\{U_c, U_{notc}\}$ without explicitly using the classifier:

Proposition 2. *A relational motif $e \in \mathrm{lgg}(O)$ is a minimal common explanation of the assignation of label c to observations in a group O with respect to classifier D and universe U if and only if $\forall u \in U_{notc}$ e does not cover u and $\forall e' \subset e, \exists u \in U_{notc}$ s.t. e' covers u*

Example 4 *Continuing with Example 3, U contains 8 models among which only $o_- = \{p(1)\}$ is in U_-. We find back the minimal common explanations for $\{o_1, o_2\}$, $r(X)$ and $p(2)$ as the minimal subsets of $\{p(1), p(2), r(X)\}$ that does not cover $\{p(1)\}$.*

Note that whenever the classifier D is a set of definite clauses, the lgg of the cover set $O \subseteq U_c$ of some clause $c \leftarrow b$ is by definition less general than b, considered as a relational motif, and therefore it covers no observations from U_{notc}. As a consequence the set of common explanations for O is not empty. As the bodies of clauses in D represent a covering of U_c, the assignation of label c

to any observation from U_c is explained through a minimal common explanation for the cover set of some clause from D.

Note that we have defined minimality according to the inclusion ordering in $\mathrm{lgg}(O)$ rather than the more natural choice of θ-subsomption. Let e and e' be two common explanations, we have that $e \subseteq e'$ implies $e\theta \subseteq e'$ which means that the set of minimal common explanations according to θ-subsomption is included into the set of minimal common explanations according to set theoretic inclusion. This means that after enumerating minimal common explanations as defined in Definition 4 we may still choose to select either the most general ones or the most specific ones according to θ. The latter choice, closer to the intuition of abductive explanations, turned out to select explanations easier to interpret in our case study of Section 6.

Example 5 *Let* $\mathrm{lgg}(O)$ *be* $\{p(1), p(2), p(X), q(X)\}$ *and* $U_{notc} = \{q(1)\}$. *The minimal common explanations according to inclusion are* $\{p(1); p(2); p(X)\}$. *The only one according to* θ*-subsumption is* $p(X)$. *The most specific ones according to* θ*-subsumption are* $\{p(1); p(2)\}$. *Chosing* $p(1)$ *or* $p(2)$ *as explanations avoid to search for substitutions for* X *when relating the explanation to the observations to explain.*

To summarize, whenever we have neither a classifier nor a formula U but that we do know the set of possible observations, i.e. $M(U)$ and their labels, we may still build common explanations. We have then to first divide the observations into labelled subsets U_c and U_{notc} and build a classifier D by PLI. Then we have to compute the lgg of the cover set of each each clause of D together with the minimal common explanations for each cover set. Algorithms for that purpose are described in the next Section. In Section 6 we take advantage of these definitions and results to explain the set of optimal trajectories associated to some (state, action) pair.

5 Building explanations

5.1 Approximation of the least general generalization for a subset of trajectories

Exact *lgg* computation for a set of observations O has a prohibitive complexity, in $\mathcal{O}(C)^n$ where C is the size (in literals) of the largest observation of O et $n = |O|$. We refer to [15] for the fully detailed algorithm, let us just mention here that the basic step in the naive *lgg* algorithm computes two observations o_1 and $o_2 \in O$ and for all predicate symbols p_i shared by o_1 and o_2 a maximally specific generalised literal for all pairs of literals $\in selection(p, o_1) \times selection(p, o_2)$ [4]. Considering that in our case study (see section 6), observations have an average number of literals of about 250 and some predicate symbols having an average number of occurrences by observation above 5, we had to develop an approximation algorithm for *lgg*.

[4] $selection(p, o)$ is the set of literals of o having predicate symbol p

We have designed a top-down *generate and test* algorithm that allows computing an approximation of the $lgg(O)$ refered to as *Bottom* in the remainder of the paper (see Algorithme 1). This algorithm handle constraints meaningful for our problem and able. to bound the size of *Bottom*.

We first define (as classically in ILP) a langage bias $\mathcal{B}$ as a list of predicate symbols associated to their arguments types. Ce biais de langage permet de construire un motif relationnel *Bottom* (et des explications associées) pour un sous-ensemble des prédicats de $\mathcal{V}$.

In the following sections, and as welle as [8], we represent the relational patterns into lists of atoms, i.e. $[l_1, ..., l_n]$. If C is a relational pattern and l a literal, we represent by $[C, l]$ the relational pattern obtained by adding l after the last literal of C. Given a seed example $g \in O$ (randomly chosen in our case), and for each predicate symbol $p \in \mathcal{B}$, algorithm 1 computes for all literals l_i instances of p in g the candidate generalised literals with the generate and test technique (denoted by $\rho(Bottom, l_i)$).

Algorithm 1 Computes *Bottom*, an approximation of *lgg(O)*

Require: O observation set, $\mathcal{B}$: language bias, fonction de score *Score*
Ensure: A relational motif *Bottom* that θ-subsumes all $o_k \in O$, and of size $\leq k * |s|$

$s \leftarrow random\_choice(O)$; $Bottom \leftarrow \emptyset$; $\theta \leftarrow \emptyset$
for each $p \in \mathcal{B}$ **do**
  for each l_i instance of p occurring in s ($l_i \in s$) **do**
    Cands $\leftarrow$ k-best literals $\in \rho(Bottom, l_i, \theta)$ given *Score*
    for each $gl_{i_j} \in Cands$ **do**
      if $[Bottom, gl_{i_j}]$ covers all $o_j \in O$ **then**
        $Bottom \leftarrow [Bottom, gl_{i_j}]$
      end if
    end for
  end for
end for
$Bottom \leftarrow reduce(Bottom)$
return *Bottom*

The ρ operator takes as arguments the current generalization *Bottom* and the current literal l_i to generalise, and the current matching substitution θ $Bottom.\theta = s$. The operator ρ controls the number of generalised literals for each l_i (at most k), the number and types of variables occurring in each generalised literals, the types of generalisation for a constant (allowing to generalise this constant by a variable already occurring in *Bottom* and linked to the same constant in θ and therefore introducing may not be maximally specific allowing for a larger variety of links within *Bottom* (see example 6). A literal gl_{i_j} is added to *Cands* if $[Bottom, gl_{i_j}]$ covers all observations of O. Still, the cardinal of $\rho(Bottom, l_i, \theta)$ can be high, the algorithm therefore ranks the candidate generalized literals and finally selects the k-best candidates given the *Score* function. The score function uses different syntatic measures (for instance, the number of fresh variablesof gl_{i_j}, the maximum degree of variables in gl_{i_j} in $[Bottom, gl_{i_j}]$, ...). It also integrates the coverage of $[Bottom, gl_{i_j}]$ on observations of $U_{\neg c}$. Once the k-best generalised literals for l_i have been identified, they are greedily added to *Bottom* if $[Bottom, gl_{i_j}]$ covers all observations of O (note that literals in *Cands* can share variables).

Bottom is finally reduced [15] to keep *in fine* maximally specific literals only.

Example 6 *After example 1 let us compute the relational motif Bottom associated to observations o_1 and o_2. Let us suppose that ρ only generates generalised literals that contains at most one variable (fresh or already occurring in Bottom). Let us first consider predicate symbol p, the most frequent one pin both observations and suppose o_1 is the seed. The first instance of p in o_1 is $p(1,2)$. $\rho(Bottom, p(1,2), \theta) = \{p(1,2), p(X,2), p(1,Y)\}$. $p(1,2)$ does not cover o_2, $p(X,2)$ and $p(1,Y)$ both cover o_1 and o_2 and are added to Bottom, θ becomes $\{X/1, Y/2\}$. Now considering the second instance of p in the seed, $p(2,3)$, $\rho(Bottom, p(2,3), \theta) = \{p(2,3), p(Y,3), p(Y',3), p(2,Z)\}$. $p(2,3)$ does not cover o_2, neither does $p(1,Y), p(Y,3)$, $p(Y',3)$ and. $p(2,Z)$ are thus added to Bottom and $\{Y'/2, Z/3\}$ is added to θ. In further iterations of the algorithm, $r(Y)$ and $q(3)$ will be added to Bottom. We finally obtain Bottom $= \{p(X,2), p(1,Y), p(Y',3), p(2,Z), r(Y), q(3)\}$, that does not need to be reduced and which is incidently the exact lgg of o_1 and o_2, with the matching substitution $\theta = \{X/1, Y/2, Y'/2, Z/3\}$ for the seed o_1.*

The relational motif *Bottom* is the lower bound of the search space for minimal common explanations for O.

5.2 Building common explanations

Algorithm 1 builds a relational motif *Bottom* which approximates $lgg(O)$ (i.e., $Bottom \preceq_\theta lgg(O)$). The next algorithm takes *Bottom* as input and builds a set of common explanations of O seen as minimal (according to $\subseteq$) and correct subsets of *Bottom* (see def. 4). The first algorithm that extends frequent itemset mining to relational motifs in a learning from interpretations framework is Warmr [6], which was relying in particular on a flexible language bias definition. Since then, a number of works in ILP have targeted relational motifs mining, in particular closed relational motifs [8]. We target here a slightly different problem, that of computing the set of all minimal (under $\subseteq$) and correct subsets of *Bottom*. We assume here all observations of O are labelled with class c (thanks to classifier D), a correct motif should not cover any observation of U_{notc}, observations of U labelled by a class other than c. Given a relational motif m, the set of observations covered by m and belonging to U_{notc} is refered to as the *critical set* of m, whereas observations belonging to U_{notc} are refered to as *critical* as far as the goal is to explain O. In other terms, this problem is directly related to that of computing the bound G of a *Version Space* with lower bound *Bottom* [12], in a relational language.

This task has been investigated in a boolean itemset mining [19]. In this work, authors propose a top down algorithm in which a current motif mg is iteratively specialised by adding an item l if $[mg, l]$ rejects at least one critical observation of mg. Adding l to mg is validated if none of the subsets or $[mg, l]$ containing l rejects the same critical observations as $[mg, l]$. We upgrade in the following this algorithm for extracting minimal and correct relational motifs from *Bottom*. This adaptation is not trivial for the following reasons. One such reason is that specialising a relational motif mg may require adding several literals at once for rejecting critical observations : it may be necessary to add so-called "bridge" literals that do not allow to reject a critical observation but that introduce new variables necessary to do so (see [16] for one of the first discussions on that point). Mutliple *lookahead* strategies have been proposed in ILP, all of them relying on *ad-hoc* search or language bias. We propose here an original strategy qui that relies on exploiting the structuration of *Bottom* in *locales*.

We refer to [4] for the formal definition of a *locale*, but intuitively, a locale of *Bottom* is a maximal set of *Bottom* literals that "share" variables. The possible instanciations of a variable within a motif are therefore only constrained by variables occurring in the same locale. A ground fact is a locale of size one.

Example 7 *The* lgg *of example 6 contains 5 locales* $\{p(1, X)\}$, $\{p(X', Y'), r(Y')\}$, $\{p(X'', 3)\}$, $\{q(3)\}$, $\{p(2, Y'')\}$

In the following, we denote by $coverage(m, O)$ where m is a relational motif and $O \subseteq U$ a set of observations the set $\{o_i \in O \mid covers(m, o_i)\}$. Algorithms 2 et 3 upgrade the algorithm [19] for building minimal and correct relational motifs. Algorithm 2 explores all possible subsets of locales of *Bottom*. This algorithm succeeds if mg is correct (the critical set of mg is empty), it fails and backtracks if the largest motif that can be built given the unexplored locales is not correct (line 2 of alg. 2). In other cases, it calls algorithm 3 for further specialising mg by exploring next unexplored locales.

Algorithm 3 builds for a given locale LK all minimal subsets of LK rejecting at least one critical observation of mg. Such a minimal subset M_i of LK can be handled as a boolean item as in [19] because, as mg_i and LK do not share any variables (by definition of a locale), $coverage([mg, M_i], NCE) = coverage(mg, NCE) \cap coverage(M_, NCE) = coverage(M_i, NCE)$.

Example 8 *After example 6, suppose o_1 et o_2 are labelled with class c an assume observtion o_3 has class $c' \neq c$: $\{p(2, 4), r(2), p(2, 3), q(3)\}$. If the locales of $lgg(\{o_1, o_2\})$ by rejection of o_3, $\{q(3)\}$, $\{p(2, Y'')\}$ and $\{p(X'', 3)\}$ are immediately rules out. $\{p(1, X)\}$ and $\{p(X', Y'), r(Y')\}$ are both correct and mininal ($p(X', Y')$ and $r(Y')$ both cover o_3 whereas $p(X', Y'), r(Y')$ rejects o_3).*

Proposition 3. *Algorithme 2 applied to the set of locales of Bottom and to a critical set of observations NCE is correct and complete : it builds all subsets of Bottom that reject all observations of NCE.*

A sketch of proof is provided in appendix[5]. Several postprocessing steps can be applied to this set of common explanations for O. For instance, considering

Algorithm 2 $mings(mg, LLK, NCE)$

Require: mg : current minimal motif, NCE: critical set of mg, $LLK = \{LK_i\}$ locales of $Bottom$ still to be explored
Ensure: MGF : set of all minimal and correct subsets of $Bottom$

$MGF \leftarrow \emptyset$
if $mg \cup LLK$ is correct **then**
  if mg is correct $(NCE = \emptyset)$ **then return** mg
  end if
  $LK \leftarrow head(LLK)$
  $nLLK \leftarrow tail(LLK)$
  for all $M_i = maxGen(LK, NCE)$ **do** ▷ see alg.3
    $emg \leftarrow [mg, M_i]$
    $rNCE \leftarrow coverage(emg, NCE)$
    $MGLK \leftarrow mings(emg, nLLK, rNCE)$
    $MGF \leftarrow MGF \cup MGLK$
  end for
  $MGWLK \leftarrow mings(mg, nLLK, NCE)$
  $MGF \leftarrow check\_min(MGF \cup MGWLK)$
end if
return MGF

Algorithm 3 $maxGen(LK, NCE)$

Require: LK : a locale $\in$ $Bottom$, NCE: critical set of mg
Ensure: MG set of minimal subsets of LK that reject at least one critical obs.of NCE

if LK does not reject any critical obs. $\in NCE$ **then**
  return $\emptyset$
end if
$MG \leftarrow \emptyset$
for all subsets mg_i of LK **do**
  $rNCE \leftarrow couverage(mg_i, NCE)$
  if $|rNCE| < |NCE|$ and mg_i is minimal **then**
    $MG \leftarrow MG \cup mg_i$
  end if
end for
return MG

that minimal subsets are extracted according to $\subseteq$, some minimal common explanations can be ordered by θ-subsumption. the MGF set can then be pruned to keep most specific ones explanations according to θ-subsumption (i.e., the most instantiated ones), which may be easier for an expert to interpet.

6 Case study

A trajectory is a sequence $p = s_0 a_0 \ldots s_t a_t \ldots s_n a_n$ where s_t is the state observed at time t and a_t the action performed in t to progress to state s_{t+1}. Most of the predicates describing the trajectories have a temporal argument: the atoms hold or not depending on the instant t along the trajectory. For part of these predicates, we use a compact representation using time intervals as arguments. When considering the truth value of some atom a, ground except from its time stamp, we divide the timeline into intervals during which a is true. The positive literal $a([b, e])$ is then true whenever a is true for all $t \in [b, e]$ and is false at time $b - 1$ and at time $e + 1$.

Example 9 *Consider a trajectory where a is true at times* $1, 2, 3, 5, 6$ *and b is true at times* $2, 3, 4, 6, 8, 9$. *The truth of a and b along the trajectory is written* $a([1, 3]), a([5, 6]), b([2, 4]), b([6, 6]), b([8, 9])$.

We consider now a deal of our game and follow the scenario of Section 2. Given a (state, action) pair we build a classifier D made of definite clauses [6]for optimality of the action, then for each clause we search for minimal common explanations associated its the cover set. The deal is as follows:

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

The decision problem faced by the artificial player Noo is seen as a deterministic Markov Decision Process. The actions are the cards played by Noo at different times of the game as the declarer, either as North or South player. The transition from a state s_t to the next state s_{t+1} is fully determined by the action a meaning that Noo knows which card the defender, either East or West, will play as a reaction to action a in state s_t. The reward $r(s_t, a)$ obtained by Noo is 1 wether the declarer wins the trick between t and $t+1$ or is 0 otherwise. Time t represents the moment in which North or South has to play and s_t is the current state of the game at time t. This means that each trick is associated to two timestamps.

Example 10 *The game starts after West has played 8 and the North hand has been unveiled. Consider a trajectory starting as follows: W8 (t1) N12 E6 (t2) S2 (t3) S13 W9. North plays 12 (the queen) in* t_1 *with null reward* $r(s_1, 12)$. *Then South plays* 2 *and we have* $r(s_2, 2) = 1$ *as North wins the first trick.*

Whatever North plays in t_1 by playing afterwards optimally the declarer will win the three first tricks ending in t_7 in which the defender has void hands. In

[6] using the ILP system cLear, developed by NukkAI

what follows we discuss the case in which in t_1 North plays a small card, say 3. We say that two cards $a < a'$ in a player hand in state s are consécutives whenever in s there is no card a'' in the hand of another player or played by another player during the current trick such that $a < a'' < a'$. Consecutive cards may be exchanged during a trajectory starting in s or after without any effect on the total reward at the end of the trajectory. To discuss optimality of the $(s_1, 3)$ pair we will display a tree of *abstract trajectories*, i.e. trajectories in which consecutive cards in a South or North hand are represented as a single action. There are various abstract optimal trajectories starting from W8 N3, displayed Figure 1 and ending on leaves numbered from 1 to 10 from the left. Non optimal actions, leading to non optimal trajectories are denoted by an ending arc towards an empty, unnumbered, leaf.

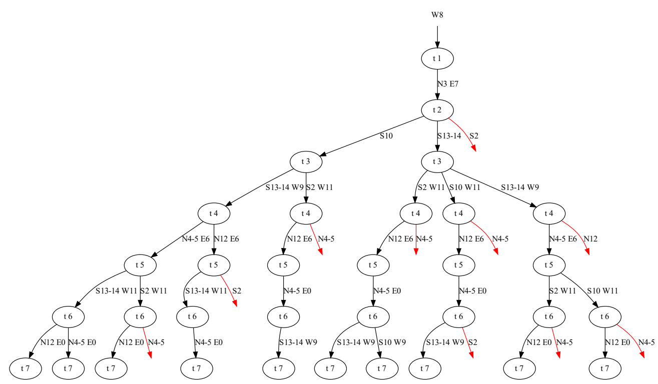

Fig. 1. Optimal abstract trajectories from W8 N3

We learn four clauses for optimality from the 40 optimal trajectories and the 104 non-optimales ones:

$$opt \leftarrow playSmallestCard(C, south, 3),$$
$$willTakeTrickWithDominant(12, north, F). \tag{1}$$

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

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

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

The abstract trajectories covered by the clauses respectively are leaves 5,6,7 for clause 1, leaves 1,3,9,10 for clause 2, leaves 1,2,4 for clause 3, and leaf 8 regarding clause 4. We focus now some minimal common explanations for the trajectories covered by clause 3 and give informal proofs of optimality:

1. $nbHonors(1, south, [4, 5])$ grounds the clause 3 body. It says that there is exactly one honor in the South hand between times t_4 and t_5 (and a different

number outside $[t_4, t_5]$). This means that i) South plays an honor (either 13 or 14) in t_3, at the beginning of trick 2 and ii) South plays its second honor in t_5 at the beginning of trick 3. From i) we infer that South has won the first trick (with 6 and 7 East cannot win this trick) and wins the second trick (13 et 14 are the highest cards). From ii) we infer that South also wins the third trick, hence the optimality of the trajectories satisfying the explanation.

2. $action(10, 2), action(13, B), action(14, C)$ says that South plays 10 in trick 1, then 13 and 14 in tricks 13 and 14, therefore winning the 3 tricks. This explanation proposes a vey simple plan to reach optimality.
3. $maxCardHand(A, 2, south, [6, 7]), nextDominant\_h(A, 12, north, [B, C]), geq(C, 4)$ is much trickier. The first atom says that from time t_6 the highest South card is 2 which means that South has previously played the cards 10, 13 and 14. This means that i) South has wins the first trick (with any of these cards), ii) South played at time t_5 meaning that South has won the trick 2. The second and third atoms states that the 12 in the North hand is the second dominant card in the game until a time $t \geq t_4$ and therefore in t_5 South yet has not played its honors (13 or 14). At a time $\geq t_5$ this becomes false because South plays an honor in t_5 and wins trick 3 leading to optimality.

The value we give to an explanation depends on its purpose. Clearly whether the purpose is to know which cards to play, explanation 2 is suitable. However if the purpose is more about learning to reason on actions and their consequences, i.e. about progressing as a player, explanations 1 and 3 are more suitable.

7 Conclusion

Motivated by explaining optimal decisions in a simplified bridge game, we have proposed a formalization to explanations of the label shared by a group of observations described in first order logics. Unlike [18, 9], we heavily rely on the observations at our disposal to construct the explanations, extensively using machine learning techniques. For that purpose we have designed and implemented algorithms to compute a lower bound of the space of explanations that approximates a least general generalization of the group of observations to be explained. We have also proposed an algorithm to enumerate minimal common explanations based on this lower bound. Finally, we were able to experiment the method on on a simplified Bridge game. By giving informal proofs of optimality built from our minimal common explanations we have emphasized that, as far as the interlocutor understands the semantics of the language and have a sufficient knowledge on the problem in hand, he may use common explanations to understand why and how his decision is optimal. One obvious perspective is to build a theory allowing to implement formal deductions from our explanations. In an other hand, to interact with a human we have to select from the possibly large number of explanations a small and diverse subset, which suppose in particular to define a similarity function on explanations. Finally a major perspective is to address the more general case where the artificial player does not know the opponent's hand, i.e. explaining decisions resulting from solving a partially observed MDP. The question of what is an explanation in this case remains open.

References

1. 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)
2. 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)
3. Blockeel, H., Raedt, L.D., Jacobs, N., Demoen, B.: Scaling up inductive logic programming by learning from interpretations. DMKD'99 **3**, 59–93 (1999)
4. 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)
5. 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)
6. Dehaspe, L.: Frequent pattern discovery in first-order logic. AI Commun. **12**(1-2), 115–117 (1999)
7. Echenim, M., Peltier, N.: A calculus for generating ground explanations. In: IJCAR'12. vol. 7364, pp. 194–209. Springer (2012)
8. 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)
9. Huang, X., Izza, Y., Ignatiev, A., Marques-Silva, J.: On efficiently explaining graph-based classifiers. In: Proceedings of KR'21. pp. 356–367 (2021)
10. Inoue, K.: Consequence-finding based on ordered linear resolution. In: IJCAI'91. pp. 158–164. Morgan Kaufmann (1991)
11. Marquis, P.: Extending abduction from propositional to first-order logic. In: FAIR. vol. 535, pp. 141–155. Springer (1991)
12. Mitchell, T.M.: Generalization as search. Artif. Intell. **18**(2), 203–226 (1982)
13. Muggleton, S., Raedt, L.D.: Inductive logic programming: Theory and methods. J. Log. Program. **19/20**, 629–679 (1994)
14. Nienhuys-Cheng, S.H., Wolf, R.D., de Wolf, R.: Foundations of Inductive Logic Programming. Springer-Verlag New York, Inc. (1997)
15. Plotkin, G.D.: A note on inductive generalization. Machine Intelligence **5**, 153–163 (1970)
16. Quinlan, J.R., Cameron-Jones, R.M.: FOIL: A midterm report. In: ECML'93. Lecture Notes in Computer Science, vol. 667, pp. 3–20. Springer (1993)
17. 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)
18. 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)
19. 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)
20. Soulet, A., Rioult, F.: Exact and Approximate Minimal Pattern Mining, pp. 61–81. Springer International Publishing (2017)
21. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press (1998)