Anna Fensel
Ana Ozaki
Dumitru Roman
Ahmet Soylu (Eds.)

LNCS 14244

Rules and Reasoning

7th International Joint Conference, RuleML+RR 2023
Oslo, Norway, September 18–20, 2023
Proceedings

Lecture Notes in Computer Science **14244**

The series Lecture Notes in Computer Science (LNCS), including its subseries Lecture Notes in Artificial Intelligence (LNAI) and Lecture Notes in Bioinformatics (LNBI), has established itself as a medium for the publication of new developments in computer science and information technology research, teaching, and education.

LNCS enjoys close cooperation with the computer science R & D community, the series counts many renowned academics among its volume editors and paper authors, and collaborates with prestigious societies. Its mission is to serve this international community by providing an invaluable service, mainly focused on the publication of conference and workshop proceedings and postproceedings. LNCS commenced publication in 1973.

Anna Fensel · Ana Ozaki · Dumitru Roman ·
Ahmet Soylu
Editors

Rules and Reasoning

7th International Joint Conference, RuleML+RR 2023
Oslo, Norway, September 18–20, 2023
Proceedings

Springer

Editors
Anna Fensel
Wageningen University and Research
Wageningen, The Netherlands

Ana Ozaki
University of Oslo
Oslo, Norway

Dumitru Roman
SINTEF AS/Oslo Metropolitan University
Oslo, Norway

Ahmet Soylu
Oslo Metropolitan University
Oslo, Norway

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-031-45071-6 ISBN 978-3-031-45072-3 (eBook)
https://doi.org/10.1007/978-3-031-45072-3

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Paper in this product is recyclable.

Preface

These are the proceedings of the 7th International Joint Conference on Rules and Reasoning (RuleML+RR). RuleML+RR joined the efforts of two well-established conference series: the International Web Rule Symposia (RuleML) and the Web Reasoning and Rule Systems (RR) conferences.

The RuleML symposia have been held since 2002 and the RR conferences since 2007. The RR conferences have been a forum for discussion and dissemination of new results on Web Reasoning and Rule Systems, with an emphasis on rule-based approaches and languages. The RuleML symposia were devoted to disseminating research, applications, languages, and standards for rule technologies, with attention to both theoretical and practical developments, to challenging new ideas and to industrial applications. Building on the tradition of both, RuleML and RR, the joint conference series RuleML+RR aims at bridging academia and industry in the field of rules, and at fostering cross-fertilization between the different communities focused on the research, development, and applications of rule-based systems. RuleML+RR aims at being the leading conference series for all subjects concerning theoretical advances, novel technologies, and innovative applications of knowledge representation and reasoning with rules.

To leverage these ambitions, RuleML+RR 2023 was organized as part of the event *Declarative AI 2023: Rules, Reasoning, Decisions, and Explanations*, which was held between the 18th and the 20th of September 2023. This event was hosted by OsloMet, Norway. With its general topic *"Declarative Artificial Intelligence"*, a core objective of the event was to present the latest advancements in AI and rules, rule-based machine learning, reasoning, decisions, and explanations and their adoption in IT systems. To this end, *Declarative AI 2023* brought together co-located events with related interests. In addition to RuleML+RR, this included the Reasoning Web Summer School (RW 2023) and DecisionCAMP 2023.

The RuleML+RR conference moreover included two subevents:

1. *Doctoral Consortium* – an initiative to attract and promote student research in rules and reasoning, with the opportunity for students to present and discuss their ideas, and benefit from close contact with leading experts in the field.
2. *International Rule Challenge* – an initiative to provide competition among work in progress and new visionary ideas concerning innovative rule-oriented applications, aimed at both research and industry.

The program of the main track of RuleML+RR 2023 included the presentation of 13 full research papers and 3 short papers. These contributions were carefully selected by the Program Committee from 46 high-quality submissions to the event. Each paper was carefully reviewed and discussed by the reviewers. Almost all of them received three reviews (87%) and all of them received two reviews. The technical program was then enriched with the additional contributions from its subevents as well as from DecisionCAMP 2023, an event aimed at practitioners which was virtual this year.

At RuleML+RR 2023, five keynote speakers were invited:

- Oscar Corcho, Professor at Universidad Politécnica de Madrid (Spain): *On the Governance of all the Artefacts Used in Knowledge Graph Creation and Maintenance Scenarios*
- Evgeny Kharlamov, Senior Expert at Bosch Center for Artificial Intelligence (Germany): *From Declarative to Neuro-Symbolic AI in Smart Manufacturing*
- Nathaniel Palmer, Director at Serco (USA): *Declarative AI at Scale: Powering a Robotic Workforce*
- Heiko Paulheim, Professor at University of Mannheim (Germany): *Knowledge Graph Embeddings meet Symbolic Schemas; or: what do they Actually Learn?*
- Fabian M. Suchanek, Professor at Télécom Paris (France): *Knowledge Bases and Language Models: Complementing Forces*

The chairs sincerely thank the keynote speakers for their contribution to the success of the event. The chairs also thank the Program Committee members and the additional reviewers for their hard work in the careful assessment of the submitted papers. Further thanks go to all authors of contributed papers for their efforts in the preparation of their submissions and the camera-ready versions within the established schedule. Sincere thanks to the chairs of the Doctoral Consortium and the Rule Challenge, and to the chairs of all co-located Declarative AI 2023 events. The chairs finally thank the entire organization team including the Publicity, Proceedings, and Sponsorship Chairs, who actively contributed to the organization and the success of the event.

A special thanks goes to all the sponsors of RuleML+RR 2023 and Declarative AI 2023: Artificial Intelligence Journal, Springer, SECAI, OsloMet, University of Oslo, SINTEF, RuleML Inc, and RR Association. A special thanks also goes to the publisher, Springer, for their cooperation in editing this volume and publication of the proceedings. We are grateful to the sponsors of the RuleML+RR 2023 as they also contributed towards the awards: RuleML+RR Harold Boley Distinguished Paper award, RuleML+RR Best Student Paper award, RuleML+RR Best Rule Challenge Paper award, RuleML+RR Best Doctoral Consortium Paper award.

August 2023

Anna Fensel
Ana Ozaki
Dumitru Roman
Ahmet Soylu

Organization

General Chairs

Ahmet Soylu	Oslo Metropolitan University, Norway
Dumitru Roman	SINTEF AS/Oslo Metropolitan University, Norway
Martin Giese	University of Oslo, Norway

Program Chairs

Anna Fensel	Wageningen University & Research, The Netherlands
Ana Ozaki	University of Oslo & University of Bergen, Norway

Doctoral Consortium

Davide Lanti	Free University of Bozen-Bolzano, Italy
Dörthe Arndt	TU Dresden, Germany
Egor V. Kostylev	University of Oslo, Norway

Rule Challenge Chairs

Jan Vanthienen	KU Leuven, Belgium
Tomáš Kliegr	Prague University of Economics and Business, Czech Republic
Paul Fodor	Stony Brook University, USA

Proceedings Chair

Dumitru Roman	SINTEF AS/Oslo Metropolitan University, Norway

Program Committee

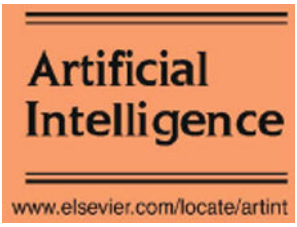
Alisa Kovtunova	TU Dresden, Germany
Aljbin Ahmeti	Semantic Web Company, Austria
Andreas Pieris	University of Edinburgh, UK
Anelia Kurteva	TU Delft, The Netherlands
Angelo Montanari	University of Udine, Italy
Anni-Yasmin Turhan	TU Dresden, Germany
Antonis Bikakis	University College London, UK
Baris Setkaya	Frankfurt University of Applied Sciences, Germany
Bernardo Cuenca Grau	University of Oxford, UK
Claudia d'Amato	Università degli Studi di Bari, Italy
Davide Sottara	Mayo Clinic, USA
Diego Calvanese	Free University of Bozen-Bolzano, Italy
Domenico Lembo	Sapienza University of Rome, Italy
Egor V. Kostylev	University of Oslo, Norway
Emanuel Sallinger	TU Wien, Austria
Erman Acar	University of Amsterdam, The Netherlands
Filip Murlak	University of Warsaw, Poland
Francesca Alessandra Lisi	Università degli Studi di Bari "Aldo Moro", Italy
Francesco Ricca	University of Calabria, Italy
Francesco Santini	Università di Perugia, Italy
Francesco M. Donini	Universita' della Tuscia, Italy
Frank Wolter	University of Liverpool, UK
Giovanni De Gasperis	Università degli Studi dell'Aquila, Italy
Grigoris Antoniou	University of Huddersfield, UK
Guido Governatori	Independent researcher, Australia
Horatiu Cirstea	Loria, France
Jan Rauch	Prague University of Economics and Business, Czechia
Jessica Zangari	University of Calabria, Italy
Jorge García-Gutiérrez	University of Seville, Spain
Jorge Martinez-Gil	Software Competence Center Hagenberg, Austria
Juliana Küster Filipe Bowles	University of St Andrews, UK
Kia Teymourian	University of Texas at Austin, USA
Livia Predoiu	Free University of Bozen-Bolzano, Italy
Livio Robaldo	University of Swansea, UK
Loris Bozzato	Fondazione Bruno Kessler, Italy
Manolis Koubarakis	National and Kapodistrian University of Athens, Greece
Mantas Simkus	TU Vienna, Austria

Marco Manna	University of Calabria, Italy
Marco Maratea	University of Genova, Italy
Maria Vanina Martinez	Universidad de Buenos Aires, Argentina
Markus Krötzsch	TU Dresden, Germany
Michaël Thomazo	Inria, France
Nurulhuda A. Manaf	National Defence University of Malaysia (UPNM), Malaysia
Ognjen Savkovic	Free University of Bozen-Bolzano, Italy
Patrick Koopmann	TU Dresden, Germany
Paul Krause	University of Surrey, UK
Pedro Cabalar	University of A Coruña, Spain
Rafael Peñaloza	University of Milano-Bicocca, Italy
Ricardo Guimarães	University of Bergen, Norway
Rolf Schwitter	Macquarie University, Australia
Roman Kontchakov	Birkbeck, University of London, UK
Sarah Alice Gaggl	TU Dresden, Germany
Sebastian Rudolph	TU Dresden, Germany
Sergio Tessaris	Free University of Bozen-Bolzano, Italy
Shqiponja Ahmetaj	Vienna University of Technology, Austria
Stefan Schlobach	Vrije Universiteit Amsterdam, The Netherlands
Stefania Costantini	University of L'Aquila, Italy
Theresa Swift	Universidade Nova de Lisboa, Portugal
Thom Fruehwirth	University of Ulm, Germany
Thomas Lukasiewicz	University of Oxford, UK
Tomas Kliegr	Prague University of Economics and Business, Czechia
Umberto Straccia	ISTI-CNR, Italy
Umutcan Şimşek	University of Innsbruck, Austria
Yuheng Wang	Stony Brook University, USA
Zaenal Akbar	National Research and Innovation Agency, Indonesia

Additional Reviewers

Aldo Ricioppo	University of Calabria, Italy
Cinzia Marte	University of Calabria, Italy
Dominik Rusovac	TU Dresden, Germany
Ivan Scagnetto	University of Udine, Italy
Nicola Saccomanno	University of Udine, Italy
Sascha Rechenberger	University of Ulm, Germany

RuleML+RR 2023 Sponsors



Invited Talks

On the Governance of all the Artefacts Used in Knowledge Graph Creation and Maintenance Scenarios

Oscar Corcho

Ontology Engineering Group, Universidad Politécnica de Madrid, Boadilla del Monte, Spain
ocorcho@fi.upm.es

Abstract. The creation and maintenance of knowledge graphs is commonly based on the generation and use of several types of artefacts, including ontologies, declarative mappings and different types of scripts and data processing pipelines, sample queries, APIs, etc. All of these artefacts need to be properly maintained so that knowledge graph creation and maintenance processes are sustainable over time, especially in those cases where the original data sources change frequently. It is not uncommon to have situations where ontologies are governed by an organisation or group of organisations, while mappings and data processing pipelines are handled by other organisations or individuals, using different sets of principles. This causes mismatches in the knowledge graphs that are generated, including the need to update all the associated artefacts (declarative mappings, sample queries, APIs, etc.) so as to keep up to date to changes in the ontologies, or in the underlying data sources. In this talk we will discuss several of the challenges associated to the maintenance of all of these artefacts in real-world knowledge graph scenarios, so as to provide some light into how we could set up a complete knowledge graph governance model that may be used across projects and initiatives.

Keywords: Knowledge graph · Governance · Ontologies · Mappings

This work was funded partially by the project Knowledge Spaces: Técnicas y herramientas para la gestión de grafos de conocimientos para dar soporte a espacios de datos (Grant PID2020-118274RB-I00, funded by MCIN/AEI/ 10.13039/501100011033)

Knowledge Graph Embeddings Meet Symbolic Schemas or: What do they Actually Learn?

Heiko Paulheim

Data and Web Science Group, University of Mannheim, Mannheim, Germany
heiko@informatik.unim-annheim.de

Abstract. Knowledge Graph Embeddings are representations of entities and relations as vectors in a continuous space, and, as such, are used in many tasks, like link prediction or entity classification [3]. While most evaluations of knowledge graph embeddings are on quantitative benchmarks [1, 2], it is still not fully understood what they are actually capable of learning. In this talk, I will show how to quantify the representative capabilities of knowledge graph embeddings using the DLCC benchmark [4], and provide insights into what kinds of logical constructs can be represented by which embedding methods. Based on those considerations, I will discuss various ways in which knowledge graph embeddings can benefit from symbolic schema information, and how those combinations open new ways of evaluating knowledge graph embeddings beyond standard metrics.

References

1. Bloem, P., Wilcke, X., van Berkel, L., de Boer, V.: kgbench: a collection of knowledge graph datasets for evaluating relational and multimodal machine learning. In: Verborgh, R., et al. (ed.) The Semantic Web. ESWC 2021. LNCS, vol. 12731, pp. 614–630, Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77385-4_37
2. Pellegrino, M.A., Altabba, A., Garofalo, M., Ristoski, P., Cochez, M.: GEval: a modular and extensible evaluation framework for graph embedding techniques. In: Harth, A., et al. (ed.) The Semantic Web. ESWC 2020. LNCS, vol. 12123, pp. 565–582, Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49461-2_33
3. Portisch, J., Heist, N., Paulheim, H.: Knowledge graph embedding for data mining vs. knowledge graph embedding for link prediction–two sides of the same coin? Semant. Web **13**(3), 399–422 (2022)
4. Portisch, J., Paulheim, H.: The DLCC node classification benchmark for analyzing knowledge graph embeddings. In: Sattler, U., et al. (ed.) The Semantic Web – ISWC 2022. ISWC 2022. LNCS, vol. 13489, pp. 592–609. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19433-7_34

From Declarative to Neuro-Symbolic AI in Smart Manufacturing

Evgeny Kharlamov[1,2]

[1] Bosch Center for Artificial Intelligence, Germany
[2] SIRIUS Research Centre, University of Oslo, Norway

Symbolic or declarative methods have been extensively used in manufacturing, e.g., as digital representations of physical objects or systems, where they have become one of the key building blocks towards digitalization and automation in the whole production value chain. Indeed, rules, ontologies, answer set programs have been used for modelling of industrial assets as Digital Twins, for industrial analytics, integration and querying of production data, process monitoring and equipment diagnostics, moreover, semantic technologies have been adopted or evaluated in a number of large high tech production companies such as Bosch, Equinor, Festo, Siemens, etc.

New trends in manufacturing that often referred to as Industry 4.0 and that are characterised by an extensive use of sensors and IoT technology brought enormous volumes of production data from manufacturing facilities. This requires data driven solutions including Machine Learning that can cope industrial big data. At the same time such solutions should account for the declarative representations in order to take the vital manufacturing knowledge captured in them to the full extend thus ensuring trust, reliability, explainability, and transparency of AI solutions.

In the keynote talk we discuss the declarative aspects of manufacturing, the new smart manufacturing trends and the necessity of Neural-Symbolic solutions. We will give a number of examples from Bosch and other companies, discuss research and industrial challenges and new exciting directions.

Keywords: Knowledge graphs · Ontologies · Machine learning · LLMs

Acknowledgements. The work was partially supported by EU projects Dome 4.0 (GA 953163), OntoCommons (GA 958371), DataCloud (GA 101016835), Graph Massiviser (GA 101093202), EnRichMyData (GA 101093202), and SMARTEDGE (GA 101092908), and the SIRIUS Centre of the Norwegian Research Council (237898).

Declarative AI at Scale: Powering a Robotic Workforce

Nathaniel Palmer

Serco Inc., 12930 Worldgate Drive Suite 6000, Herndon, VA 20170, USA
ngpalmer@protonmail.com

Abstract. This talk presents the results of a multi-year journey applying Declarative AI to a critical government mission. Leveraging commercial-of-the-self components and innovative design patterns, this journey has combined Deep Neural Network (DNN), and Machine Learning (ML) together with Decision Management (DM), and Robotic Process Automation (RPA) to deliver a robotic workforce powered by Declarative AI, performing complex case management alongside human case workers. Results include substantial gains in efficiency, quality, and consistency verifying the eligibility of tens of millions of consumers seeking to requiring government benefit eligibility verification. One of the largest and most complex applications of AI within the federal government arena, the results of this journey makes a compelling illustration of the difference between Statistical and Declarative AI – at scale! This session will feature a transparent presentation of our results, metrics, approach, and lessons learned (including many never before disclosed to a public audience.) You will learn how we leveraged Declarative AI to escape and exceed the traditional boundaries of automation, moving from discrete tasks to perform "robo-adjudication" delivering greater accuracy, efficiency, and quality of work. Also demonstrated will be how our Declarative AI is leveraged to assign work to humans and robots, ensuring every time right work performed by the right worked at precisely the right moment. Nathaniel will show strategies for automation at this massive scale can be executed with full transparency and accountability, eliminating the reliance on subjective interpretation of policies and rules, while delivery more accurate analytics and ensuring program integrity. He will discuss how to deliver intelligent automation at scale, while avoiding the pitfalls which inevitably otherwise doom to fail initiatives of this size and scope, including what challenges were overcome as well as those unforeseeable at the outset.

Keywords: Decision management · Robotic process automation · Machine learning

Contents

Explaining Optimal Trajectories

Celine Rouveirol[2], Malik Kazi Aoual[1,2], Henry Soldano[1,2,3(✉)], and Veronique Ventos[1]

[1] Nukkai, Paris, France
[2] UMR CNRS 7030 Institut Galilée - Université Sorbonne Paris Nord, LIPN, Villetaneuse, France
soldano@lipn.univ-paris13.fr
[3] UMR CNRS 7205 Museum National d'Histoire Naturelle, ISYEB, Paris, France

Abstract. We propose a definition of *common explanation* for the label shared by a group of observations described as first order interpretations, and provide algorithms to enumerate *minimal common explanations*. This was motivated by explaining how performing some action, for instance a card played during a card game play, results in winning a maximum total reward at the end of the trajectory. As there are various ways to reach this reward, each associated to a group of trajectories, we propose to first build groups of trajectories and then build minimal common explanations for each group. The whole method is illustrated on a simplified Bridge game.

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 and must answer at any point in the gameplay queries of the type "How does the chosen action leads to winning a maximum number of tricks?".

Expected explanations for choosing action a in state s then refer to a set of possible optimal trajectories starting from (s, a). For that purpose, we need a way to group trajectories in a limited number of groups, each made of trajectories whose optimality may be explained in the same way. Then, we have to define and search for *common explanations* of optimality for a group of trajectories.

Our definitions are in line with previous works on abductive explanations of the label assigned to a single observation by a logical classifier [2,3,6,10] that we have adapted or extended for our purpose in several directions. First consider that any (state, action) pair (s, a) is associated to the set of all the *possible* trajectories starting from (s, a) and called the *universe* associated to (s, a). We

A. Fensel et al. (Eds.): RuleML+RR 2023, LNCS 14244, pp. 206–221, 2023.
https://doi.org/10.1007/978-3-031-45072-3_15

also consider a classifier D that labels the trajectories as optimal or non-optimal. Explanations are then adapted in the following ways:

- A trajectory is described in first order logics as a set of ground literals and an explanation for a single trajectory is a subset of these literals.
- An explanation depends on the universe, introduced as a formula U whose set of models is the universe.
- A *common explanation* for a set of trajectories is defined as an existentially quantified conjunction of literals [13], we call a *relational motif*.

We also need a way to select groups of trajectories guaranteed to have at least one common explanation. 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). To explain a tree of optimal trajectories, the method consists in learning a rule based classifier, consider the coverage O of each rule, building for each rule the most specific relational motif $lgg(O)$ that holds for all trajectories in O, and extracting from $lgg(O)$ the minimal common explanations for O.

Technically, we solve the problem of enumerating minimal common explanations by addressing two sub-problems; i) Computing a least general generalization (lgg) under θ-subsumption for a group of observations O and ii) Searching for all minimal subsets of $lgg(O)$ that do not θ-subsumes any observation with a label different from the label shared by the observations in O. These problems have been studied in different contexts. Since [17] 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 constraints has been studied in many contexts, in particular in data mining (see for instance [22]) but not often when considering relational data in which various useful algorithmic properties are lost [9].

In Sect. 2 we describe the scenario of our case study in which an artificial player has to explain its actions. After presenting some standard notations in Sect. 3, we introduce and discuss in Sect. 4 minimal common explanations for a group of observations together with their use in explaining a set of optimal trajectories. We propose in Sect. 5 algorithms that enumerate them and illustrate the whole method on our case study in Sect. 6. We discuss then related work in Sect. 7 and conclude in Sect. 8.

2 Scenario

We consider a simplified bridge card game with a single color in which 13 cards (with values from 2 to 14) are distributed among 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 South 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 this scenario *Noo* plays the declarer role, and knows the *WE* hands. We also suppose that in any current state e *Noo* also knows for each possible action a which card the defender would play. *Noo* then solves a MDP: *Noo* computes for any (state s action a) pair accessible from the starting state, the maximum total reward $q(s, a)$, i.e. the maximal number of tricks, the declarer wins by playing a in state s and further playing an optimal action all along the trajectory. Action a is optimal in state s if it maximises $q(s, a)$. This scenario is as follows:

1. *Noo* chooses an optimal action a in current state s
2. *M. X* asks ow playing a may result in winning $q(s, a)$ tricks
3. *Noo* then builds a rule-based model D for optimality of trajectories and uses D to build groups of optimal trajectories sharing 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. We omit the $\forall$ (resp. $\exists$) quantifier when it is clear from the context that we deal with a clause (resp. a relational motif). In the rest of this section, we will adapt definitions initially introduced by [17] 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 [15] is θ-subsumption [17] 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 and D is unique and computed in time $\mathcal{O}(|C| \cdot |D|)$ [16].*

In the following, an *observation* is a Herbrand interpretation of the vocabulary $\mathcal{V}$ (see the learning from interpretations framework [4] or from multiple interpretations [9]). The *coverage* relationship between a relational motif C and an observation o , denoted $covers(C, o)$ holds 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 and o_2, $o_1 = \{p(1,2), p(2,3), r(2), q(3)\}$ and $o_2 = \{p(1,3), p(2,4), r(4), q(3)\}$. The* lgg *of o_1 and o_2 is: $\exists\ p(1, X_1), p(X_2, X_3), p(X_4, 3), p(2, X_5), r(X_3), q(3)$ with matching substitutions $\theta_1 = \{X_1/2, X_2/1, X_3/2, X_4/2, X_5/3\}$ and $\theta_2 = \{X_1/3, X_2/2, X_3/4, X_4/1, X_5/4\}$. This formula is reduced. Note that the lgg of o_1 and o_2 is longer (in number of literals) than both o_1 and o_2.*

4 Explanations

We consider *abductive explanations* investigated in recent works to explain the label assigned to some observation entered as input of a decision tree or a random forest [2,3,6,10]. In these works an observation o is represented by the values taken by a set of attributes. 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 adapt this definition.

In a relational context, an observation o is described as a Herbrand interpretation of some datalog language (see Sect. 3) that we represent as a conjunction of literals. The classifier D is as well a first order formula of this language and an explanation for assigning the class label c to an observation may be defined

as a ground clause [19] whose head is the label or, as we do hereunder, as the body of such a clause, i.e. a ground relational motif. However when turning to an explanation common to a group of observations, we rather consider general relational motifs, following the definition of abdictive explanations in first order logics first proposed by P. Marquis [13] and then further investigated from an operational point of view [8,12].

In what follows, an observation is the subset o of positive literals, excluding its class label in C, that hold for this observation and that we may also represent as a ground relational motif. Furthermore, the set of possible observations in the problem at hand, we refer to as its *universe*, is only part of the whole set of interpretations and is represented below as a formula U that does not mention labels and whose set of models is 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 at 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:

Definition 3. *An explanation of the assignation of label c to an observation o with respect to a classifier D and an 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 formula U the universe 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 $r(1)$ and U we deduce $p(1)$ and from $p(1), r(1)$ and D we deduce $+$. As a consequence though $\{p(1), r(1)\}$ still explains the $+$ label for o_1, it 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 label c to all observations in a group O, with respect to a classifier D and a universe U, is a relational motif $e \subseteq lgg(O)$ 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 add observation $o_2 = \{p(1), p(2), r(2)\}$ with same label $+$ as o_1. Let $O = \{o_1, o_2\}$, we obtain $lgg(O) = \{p(1), p(2), r(X)\}$ with $lgg(O).\{X/1\} \subseteq o_1$ 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 set of minimal common explanations for $\{o_1, o_2\}$ is then $\{r(X); p(2)\}$.

The following property relates common explanations to explanations:

Proposition 1. *Let e be a common explanation for O, then for any $o \in O$ there exists a substitution θ such that $e.\theta$ is an explanation for o.*

Proof. By definition of a lgg there exists some θ such that $lgg(O).\theta \subseteq o$ and as e is a common explanation for O we have $e \subseteq lgg(O)$ and therefore $e.\theta \subseteq o$. As we have $e.\theta \models e$ and $e, U, D \models c$ we also have $e.\theta, U, D \models c$ which means that $e.\theta$ is an explanation for o.

In what follows 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 $U_{\text{notc}} = M(U) \setminus U_c$. We may then build the minimal common explanations for groups of observations with label c from the partition $\{U_c, U_{notc}\}$ without using the classifier:

Proposition 2. *A relational motif $e \subseteq lgg(O)$ is a minimal common explanation for O 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 has 8 models among which only $o_- = \{p(1)\}$ has label $-$. We find back the minimal common explanations for $\{o_1, o_2\}$, namely $\{r(X)\}$ 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 coverage $O \subseteq U_c$ of a clause $c \leftarrow b$ is by definition less general than b and therefore it covers no observation from U_{notc}, which means that the set of common explanations for O is not empty. By construction any observation in U_c belongs to the coverage of some clause from D, and is therefore explained by at least one common explanation.

To summarize, whenever we have neither a classifier nor a formula U but that we do know the set of possible observations and their labels, we may still build common explanations by first building a classifier D, then computing the lgg of the coverage of each clause of D, and finally computing the associated minimal common explanations. Required algorithms are described Sect. 5.

Minimal Common Explanations in Practice. Note that we have defined minimality of a relational motif in $lgg(O)$ according to the subset inclusion ordering, i.e. minimality means maximal *conciseness* in terms of literals. However, we may still choose, among the motifs of equal size, the most specific ones according to θ-subsumption. This turned out to select explanations easier to interpret in our case study of Sect. 6.

Example 5. *Let* $lgg(O)$ *be* $\{p(1), p(2), p(X), q(X), r(X,Y), w(Y,Z)\}$ *and* $U_{notc} = \{q(1), w(2,3)\}$. *The minimal common explanations are* $\{p(1); p(2); p(X); r(X,Y)\}$. *The most specific ones among them are* $\{p(1); p(2); r(X,Y)\}$.

5 Building Explanations

5.1 Approximation of the Least General Generalization for a Subset of Trajectories

Exact *lgg* computation [17] 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 and $n = |O|$. Considering that in our case study (see Sect. 6), observations have an average number of literals of about 250 and some predicate symbols having an average occurrence number by observation above 5, we had to develop an approximation algorithm for *lgg*.

We have implemented a top-down *generate and test* algorithm that computes an approximation of the $lgg(O)$ referred to as *Bottom* in the remainder of the paper. This algorithm is flexible enough to handle constraints meaningful for our problem and is able to bound the size of *Bottom*.

We first define a language bias $\mathcal{V} \subseteq \mathcal{B}$ as a list of predicate symbols associated to their arguments types. The specialisation algorithm randomly selects a seed $\in O$, and applies the ρ specialisation operator to every literal $l_i \in s$. ρ takes as arguments the current generalization *Bottom*, l_i, and the current matching substitution θ ($Bottom.\theta = s$).

In the following, we represent a relational motif as a list of atoms, i.e. $[l_1, ..., l_n]$. If C is a relational motif and l a literal, $[C, l]$ is the relational motif obtained by adding l after last literal of C.

The refinement 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 literal. More importantly it controls which generalisations are allowed for a constant: does a constant generalize to a variable already occurring in *Bottom* and linked to the same constant in θ – therefore creating a "link" within *Bottom* (see Example 6) – or does it only generalize to a fresh variable. 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 generalised literals and finally selects the k-best candidates given the *Score* function. The score function uses the number of variables and fresh variables of 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_{notc}. Once the k-best generalised literals

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

Require: O: observation set, $\mathcal{B}$: language bias, $Score$: score function
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*

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 [17] to keep *in fine* maximally specific literals only.

Example 6. *From Example 1, let us suppose* o_1 *is the seed and that* ρ *satisfies the following constraints :* ρ *only generates generalised literals that contains at most one variable (fresh or already occurring in Bottom). Let us start with predicate symbol p, the most frequent predicate symbol in both* o_1 *and* o_2*. The first instance of p in* o_1 *is* $p(1,2)$*.* $\rho(Bottom, p(1,2), \theta) = \{p(1,2), p(X_0,2), p(1,X_1)\}$*.* $p(1,2)$ *does not cover* o_2*, neither does* $p(X_0,2)$*;* $p(1,X_1)$ *covers both* o_1 *and* o_2 *and is added to Bottom,* θ *becomes* $\{X_1/2\}$*. Now considering the second instance of p in the seed,* $p(2,3)$*,* $\rho(Bottom, p(2,3), \theta) = \{p(2,3), p(X_1,3), p(X_2,3), p(2,X_3)\}$*.* $p(2,3)$ *does not cover* o_2*, neither does* $p(X_1,3)$*;* $p(X_2,3)$ *and* $p(2,X_3)$ *are thus added to Bottom and* $\{X_2/2, X_3/3\}$ *is added to* θ*. In further iterations of the algorithm,* $r(X_4)$ *and* $q(3)$ *are added to Bottom and* $\{X_4/2\}$ *to* θ*. We finally obtain* $Bottom = \{p(1,X_1), p(X_2,3), p(2,X_3), r(X_4), q(3)\}$ *with the matching substitution* $\theta = \{X_1/2, X_2/2, X_3/3, X_4/2\}$ *for the seed* o_1 *that does not need to be reduced. Bottom is an approximation of the exact lgg of* o_1 *and* o_2*, as it does not contain* $p(X_5, X_4)$*.*

The goal when proposing this approximate lgg algorithm is to be able to bound the size of *Bottom* as well as to parameterize the introduction of "links" (shared variables) in *Bottom*. Once computed, 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 $\subseteq$-minimal and correct subsets of *Bottom* (see Definition 4). The first algorithm that extends frequent itemset mining to relational motifs in a learning from interpretations framework is Warmr [7], 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 [9]. We target here a slightly different problem, that of computing the set of all minimal (under $\subseteq$) and correct subsets of *Bottom*. This problem is directly related to that of computing the bound G of a *Version Space* with lower bound *Bottom* [14] in a relational language.

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.

A close task has been investigated in boolean itemset mining [22]. In this work, the 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 of $[mg, l]$ 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 [18] 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 that exploits the structure of *Bottom* in *locales*.

We refer to [5] 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 instantiations 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 1 has 5 locales* $\{p(1, X_1)\}$, $\{p(X_2, X_3), r(X_3)\}$, $\{p(X_4, 3)\}$ $\{p(2, X_5)\}$, $\{q(3)\}$. *The locales associated to Bottom in Example 6 are the same except for the second locale that reduces to* $\{r(X_3)\}$.

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 and 3 upgrade the algorithm [22] 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 Algorithm 2). In other cases, it calls Algorithm 3 for further specialising mg by exploring yet unexplored locales.

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*

```
1:  MGF ← ∅
2:  if mg ∪ LLK is correct then
3:      if mg is correct (NCE = ∅) then return mg
4:      end if
5:      LK ← head(LLK)
6:      nLLK ← tail(LLK)
7:      for all M_i = maxGen(LK, NCE) do                ▷ see alg.3
8:          emg ← [mg, M_i]
9:          rNCE ← coverage(emg, NCE)
10:         MGLK ← mings(emg, nLLK, rNCE)
11:         MGF ← MGF ∪ MGLK
12:     end for
13:     MGWLK ← mings(mg, nLLK, NCE)
14:     MGF ← check_min(MGF ∪ MGWLK)
15: end if
16: return MGF
```

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 [22] because, as M_i and LK do not share any variables (by definition of a locale), $coverage([mg, M_i], NCE) = coverage(mg, NCE) \cap coverage(M_i, NCE)$.

Example 8. *From Example 6, suppose* o_1 *et* o_2 *are labelled with class* c *and assume observation* o_3 *has class* $c' \neq c$ *:* $\{p(2,4), r(2), p(2,3), q(3)\}$. *Three locales of* $lgg(\{o_1, o_2\})$ *cover* o_3*, namely* $\{p(X_4, 3)\}$, $\{p(2, X_5)\}$ *and* $\{q(3)\}$*, are immediately ruled out.* $\{p(1, X)\}$ *and* $\{p(X', Y'), r(Y')\}$ *are both correct and* $\subseteq$*-minimal. Considering the locales of Bottom, all of them except* $\{p(1, X)\}$ *cover* o_3*, yielding a single common explanation.*

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

If needed, and as suggested in Sect. 4, the MGF set may then be pruned to keep among explanations of equal size the most specific ones (i.e., the most instantiated ones) according to θ-subsumption.

Algorithm 3. *maxGen*(LK, NCE)

Require: LK : a locale $\in$ *Bottom*, NCE: critical set of mg
Ensure: MG: set of minimal subsets of LK rejecting at least one element from NCE

```
if LK does not reject any critical observation ∈ NCE then
    return ∅
end if
MG ← ∅
for all subsets mg_i of LK do
    rNCE ← couverage(mg_i, NCE)
    if |rNCE| < |NCE| and mg_i is minimal then
        MG ← MG ∪ mg_i
    end if
end for
return MG
```

6 Case Study

A trajectory is a sequence $p = s_0 a_0 \dots s_t a_t \dots s_n a_n$ 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 Sect. 2. Given a (state, action) pair we build a classifier D made of definite clauses[1] for optimality of the action, then for each clause we search for minimal common explanations for its coverage. 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 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 0 except for the last action of the trajectory where its value is the number of tricks won by the declarer along

[1] Using the ILP system cLear, developed by NukkAI.

the trajectory. 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 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 in Fig. 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. We learn four clauses for optimality from the 40 optimal trajectories and the 104 non-optimal ones:

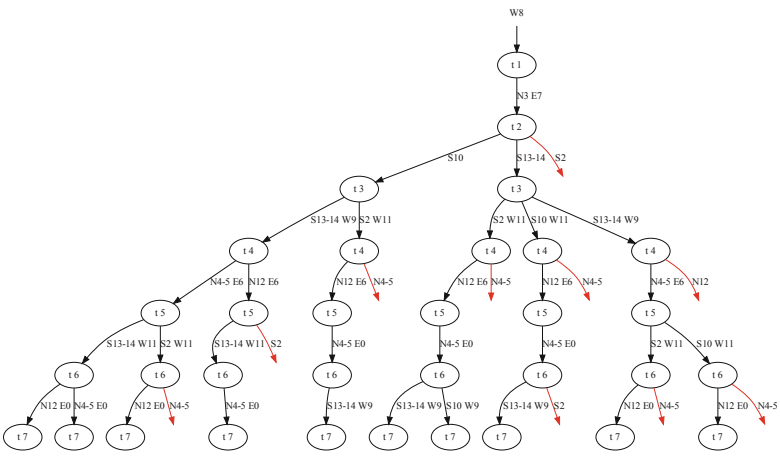


Fig. 1. Optimal abstract trajectories from W8 N3

$$opt \leftarrow playSmallestCard(Card, south, 3), \quad willTakeTrickWith(12, north, T). \tag{1}$$

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

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

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

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[2]. We focus now on 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. $action(10, 2), action(13, T1), action(14, T2)$ says that South plays 10 in trick 1, then 13 and 14 in tricks 2 and 3, therefore winning the 3 tricks. This explanation proposes a very simple plan to reach optimality.
3. $maxCardHand(2, south, [6, 7]), nextDominant(12, north, [Tb, Te]), Te \geq 4$ is 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. and therefore i) South has won the first trick (with any of these cards) and ii) South plays at time t_5 and therefore has won trick 2 with an honor (10 would be beaten by the 11 that West would have played according to the defender model[3]). The second and third atoms state that until a time $Te \geq t_4$ there is one player (South) that has cards higher than the 12 in North hand and so 10 has been played before t_4 and as a consequence South necessarily plays its last honor in t_5 and win trick 3.

The value we give to an explanation depends on its purpose. If the purpose is to know which cards to play, explanation 2 is suitable, while if the purpose is about learning to reason on actions and their consequences, i.e. about progressing as a player, explanations 1 and 3 are better. Of course, a request for

[2] While most predicates are self-explanatory some are not. $willTakeTrickWith(12, north, T)$ says that at time T and by playing 12 which is the highest card on the board or in hands of players that yet have to play in the current trick, North will win the trick. $nbThreats(2, Player, 0, [7, 7])$ says that at time t_7 card 2 of *Player* has no threats by cards from its opponents. It also says that such threatening cards exists in t_6.

[3] If possible the opponent plays a card higher than the last card played in the trick.

explanation may include information about this purpose, resulting in selecting for the explanation process only part of the predicates describing the trajectories.

7 Related Work

Many ILP systems indirectly address the explainability issue, because their output – Logic Programs – are explicit, as opposed to black box learning systems. Still, if supervised learning systems provide explicit models, much work remains to be done to provide useful explanations to domain experts: such supervised models may not be concise enough [1,21] for a human to understand, or may not capture the cause of the classification of an observation, but some of its side effects. As opposed to [1], we do not introduce additional predicates to make the model concise but we fully make use of the vocabulary designed by experts and used for building interpretations as the language for explanations. We use a notion of minimality of explanation, as systems that build abductive explanations do in a propositional logic context [3,11]. One of our originality is to search for shared explanations by observations of a given subgroup sharing the same label. [19] addresses the problem of finding contrastive explanations for an observed instance as minimal changes to an instance so that its class shifts. The authors rely on the notion of near-miss example introduced by Winston (a negative example closest to a positive one) for defining a near miss explanation.

8 Conclusion

We have proposed a definition for explanations of the label shared by a group of observations described in first order logics. Such kind of explanation is in particular mandatory when we have to explain a decision based on consequences that also depends on further decisions. We also propose to build and use a rule-based model in order to define such groups of observations. Unlike [10,20], we heavily rely on the observations at our disposal to construct the explanations. Technically this implies building an approximation of the lower bound of the space of common explanations for a group of observations and enumerating the minimal common explanations included in this bound.

By giving informal proofs of optimality of decisions based on minimal common explanations for a simple card game 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/she may use common explanations to understand why and how some decision is suitable. One obvious perspective in our case study is to build a formal theory allowing to implement deductions from our explanations. Note that though we use in our experiments a very simple model of the opponent, any deterministic model, in particular one simulating a min-max opponent, is suitable for common explanations. A major perspective is to address the more general cases where the artificial player does not know the opponent's hand, and have then to explain decisions resulting from incomplete information. Definition of explanations in this case remains open.

Acknowledgements. We thank Dr Junkang Li who wrote the program that solves the MDP and Dr Dominique Bouthinon for helping us to apply the ILP program cLear for the simplified Bridge game. M. Kazi Aoual is partially supported by ANRT through a CIFRE agreement.

References

1. Ai, L., Muggleton, S.H., Hocquette, C., Gromowski, M., Schmid, U.: Beneficial and harmful explanatory machine learning. Mach. Learn. **110**(4), 695–721 (2021)
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 2022, pp. 643–650 (2022)
3. Audemard, G., Bellart, S., Bounia, L., Koriche, F., Lagniez, J., Marquis, P.: On the explanatory power of boolean decision trees. Data Knowl. Eng. **142**, 102088 (2022)
4. Blockeel, H., Raedt, L.D., Jacobs, N., Demoen, B.: Scaling up inductive logic programming by learning from interpretations. In: DMKD 1999, vol. 3, pp. 59–93 (1999)
5. Cohen, W.W., Jr., Page, C.D.: 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: ECAI 2020. 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: Gramlich, B., Miller, D., Sattler, U. (eds.) IJCAR 2012. LNCS (LNAI), vol. 7364, pp. 194–209. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31365-3_17
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 2021, pp. 356–367 (2021)
11. Ignatiev, A., Narodytska, N., Marques-Silva, J.: Abduction-based explanations for machine learning models. In: AAAI 2019, pp. 1511–1519 (2019)
12. Inoue, K.: Consequence-finding based on ordered linear resolution. In: IJCAI 1991, pp. 158–164. Morgan Kaufmann (1991)
13. Marquis, P.: Extending abduction from propositional to first-order logic. In: Jorrand, P., Kelemen, J. (eds.) FAIR 1991. LNCS, vol. 535, pp. 141–155. Springer, Heidelberg (1991). https://doi.org/10.1007/3-540-54507-7_12
14. Mitchell, T.M.: Generalization as search. Artif. Intell. **18**(2), 203–226 (1982)
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., de Wolf, R.: Foundations of Inductive Logic Programming. Springer, New York (1997). https://doi.org/10.1007/3-540-62927-0
17. Plotkin, G.D.: A note on inductive generalization. Mach. Intell. **5**, 153–163 (1970)
18. Quinlan, J.R., Cameron-Jones, R.M.: FOIL: a midterm report. In: Brazdil, P.B. (ed.) ECML 1993. LNCS, vol. 667, pp. 1–20. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-56602-3_124

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: Proceedings of 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144 (2016)
21. Shakerin, F., Gupta, G.: Induction of non-monotonic logic programs to explain boosted tree models using LIME. In: Proceedings of AAAI 2019, pp. 3052–3059 (2019)
22. Soulet, A., Rioult, F.: Exact and approximate minimal pattern mining. In: Guillet, F., Pinaud, B., Venturini, G. (eds.) Advances in Knowledge Discovery and Management. SCI, vol. 665, pp. 61–81. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-45763-5_4

Author Index

A. Fensel et al. (Eds.): RuleML+RR 2023, LNCS 14244, p. 255, 2023.
https://doi.org/10.1007/978-3-031-45072-3