IJCAI-17 Workshop on Explainable AI (XAI)

Proceedings

20 August 2017 | Melbourne, Australia
http://home.earthlink.net/~dwaha/research/meetings/ijcai17-xai

Organizers

David W. Aha (Naval Research Laboratory, USA) (david.aha@nrl.navy.mil)
Trevor Darrell (University of California, Berkeley, USA)
Michael Pazzani (University of California, Riverside, USA)
Darryn Reid (Defence Science & Technology Group, Australia)
Claude Sammut (University of New South Wales, Australia)
Peter Stone (University of Texas, Austin, USA)

Table of Contents

Workshop Description

Explainable Artificial Intelligence (XAI) concerns, in part, the challenge of shedding light on opaque machine learning (ML) models in contexts for which transparency is important, where these models could be used to solve analysis (e.g., classification) or synthesis tasks (e.g., planning, design). Indeed, most ML research usually focuses on prediction tasks but rarely on providing explanations/justifications for them. Yet users of many applications (e.g., related to autonomous control, medical, financial, investment) require understanding before committing to decisions with inherent risk. For example, a delivery drone should explain (to its remote operator) why it is operating normally or why it suspends its behavior (e.g., to avoid placing its fragile package on an unsafe location), and an intelligent decision aid should explain its recommendation of an aggressive medical intervention (e.g., in reaction to a patient's recent health patterns). Addressing this challenge has increased in urgency with the increasing reliance of learned models in deployed applications.

The need for interpretable models exists independently of how models were acquired (i.e., perhaps they were hand-crafted, or interactively elicited without using ML techniques). This raises several questions, such as: how should explainable models be designed? How should user interfaces communicate decision making? What types of user interactions should be supported? How should explanation quality be measured? And what can be learned from research on XAI that has not involved ML?

This workshop will provide a forum for sharing and learning about recent research on interactive XAI methods, highlighting and documenting promising approaches, and encouraging further work, thereby fostering connections among researchers interested in ML (and AI more generally), human-computer interaction, cognitive modeling, and cognitive theories of explanation and transparency. While sharing an interest in technical methods with other workshops, the XAI Workshop will have a distinct problem focus on agent explanation problems, which are also seen as necessary requirements for human-machine teaming. This topic is of particular importance to (1) deep learning techniques (given their many recent real-world successes and black-box models) and (2) other types of ML and knowledge acquisition models, but also (3) application of symbolic logical methods to facilitate their use in applications where supporting explanations is critical.

We gratefully thank our invited speakers and paper contributors to this workshop! Thanks also to Dan Magazzeni, who served as IJCAI-17 Workshops Chair and ensured that this workshop ran smoothly.

Agenda

Session 1: Perspectives

0830-0835: Welcome: David W. Aha (Naval Research Laboratory, USA)
0835-0900: *Why Explain?* **Raymond Sheh** (Curtin U., Australia)
0905-0930: *Uncertainty, Resource Allocation, & Trust* **Darryn Reid** (DSTG, Australia)
0935-1000: *Explainable Recommendations* **Barry Smyth** (UC Dublin, Ireland)

1000-1030: **Coffee Break**

Session 2: Machine Learning

1030-1055: *Deep Learning for Perception, Action, and Explanation* **Trevor Darrell** (UCB, USA)
1100-1125: *Beyond Machine Learning: Delivering Technology for People through Explainable AI*
Freddy LeCue (Accenture Technology Labs, Dublin (Ireland) & INRIA (France))
1130-1155: *Knowledge Representation and Reasoning Challenges in Explainable Agency*
Mohan Sridharan (U. Auckland, New Zealand)
1200-1230: Poster Advertisements (9 papers × 3 minutes; preloaded on one laptop)

1230-1400: **Lunch**

Session 3: Planning, Inmates, & Posters Session

1400-1425: *Explainable Planning* **Daniele Magazzeni** (King's College London, UK)
1430-1445: *Explainable AI: Beware of Inmates Running the Asylum.*
Or: How I Learnt to Stop Worrying and Love the Social and Behavioural Sciences
Tim Miller, **Piers Howe**, and **Liz Sonenberg** (U. Melbourne)
1450-1600: Poster Session

- *Explanation and Justification in Machine Learning: A Survey* **Or Biran** and **Courtenay Cotton** (n-Join, USA)
- *A Framework for Explanation of Machine Learning Decisions* **Christopher Brinton** (Mosaic ATM, USA)
- *An Architecture for Explainable Text Classification by Jointly Learning Lexicon and Modifier Terms*
Jérémie Clos, Nirmalie Wiratunga, and **Stewart Massie** (Robert Gordon U., Scotland)
- *Explainable Planning* **Maria Fox, Derek Long,** and **Daniele Magazzeni** (King's College London, UK)
- *Towards Compact Interpretable Models: Learning and Shrinking Probabilistic Sentential Decision Diagrams*
Yitao Liang and **Guy Van den Broeck** (UCLA, USA)
- *Using Explanations to Improve Ensembling of Visual Question Answering Systems*
Nazneen Rajani and **Raymond Mooney** (U. Texas @ Austin, USA)
- *A Characterization of Monotone Influence Measures for Data Classification*
Jakub Sliwinski, Martin Strobel, and **Yair Zick** (National U. and Nanyang U., Singapore)
- *Regulatory Mechanisms and Algorithms towards Trust in AI/ML*
Eva Thelisson, Kirtan Padh, and **L. Elisa Celis** (U. Fribourg & EPFL, Switzerland)
- *Unsupervised Neural-Symbolic Integration* **Son Tran** (CSIRO, Australia)
- *Towards Explainable Tool Creation by a Robot*
Handy Wicaksono, Claude Sammut, and **Raymond Sheh** (UNSW, Petra Christian U., & Curtin U., Australia)

1600-1630: **Coffee Break**

Session 4: User Interaction & Panel

1630-1655: *Explicability and Explanations in Human-Aware AI Agents* **Rao Kambhampati** (ASU, USA)
1700-1800: Panel: *Reflections on Explainable AI*

- **Claude Sammut** (UNSW), **Michael Pazzani** (UC Riverside, USA), & **Jason Scholz** (DSTG, Australia)

1800: Wrap-Up: David Aha (NRL, USA)

Invited Speakers

Trevor Darrell (University of California, Berkeley, USA)

- *Deep Learning for Perception, Action, and Explanation*
- **Abstract**: Learning of layered or *deep* representations has provided significant advances in computer vision in recent years, but has traditionally been limited to fully supervised settings with very large amounts of training data, where the model lacked interpretability. New results in adversarial adaptive representation learning show how such methods can also excel when learning across modalities and domains, and further can be trained or constrained to provide natural language explanations or multimodal visualizations to their users. I'll present recent long-term recurrent network models that learn cross-modal description and explanation, using implicit and explicit approaches, which can be applied to domains including fine-grained recognition and visuomotor policies.
- **Bio**: Prof. Darrell is on the faculty of the CS and EE Divisions of the EECS Department at UC Berkeley. He leads Berkeley's DeepDrive (BDD) Industrial Consortia, is co-Director of the Berkeley Artificial Intelligence Research (BAIR) lab, and is Faculty Director of PATH at UC Berkeley. Darrell's group develops algorithms for large-scale perceptual learning, including object and activity recognition and detection, for a variety of applications including autonomous vehicles, media search, and multimodal interaction with robots and mobile devices. His areas of interest include computer vision, machine learning, natural language processing, and perception-based human computer interfaces. Prof. Darrell previously led the vision group at the International Computer Science Institute in Berkeley, and was on the faculty of the MIT EECS department from 1999-2008, where he directed the Vision Interface Group. He was a member of the research staff at Interval Research Corporation from 1996-1999, and received the S.M., and PhD. degrees from MIT in 1992 and 1996, respectively. He obtained the B.S.E. degree from the University of Pennsylvania in 1988.

  Prof. Darrell also serves as consulting Chief Scientist for the start-up Nexar, and is a technical consultant on deep learning and computer vision for Pinterest. Darrell is on the scientific advisory board of several other ventures, including DeepScale, WaveOne, SafelyYou, and Graymatics. Previously, Darrell advised Tyzx (acquired by Intel), IQ Engines (acquired by Yahoo), Koozoo, BotSquare/Flutter (acquired by Google), and MetaMind (acquired by Salesforce). As time permits, Darrell has served and is available as an expert witness for patent litigation relating to computer vision.

Rao Kambhampati (Arizona State University, USA)

- *Explicability and Explanations in Human-Aware AI Agents*
- **Abstract**: Human-aware AI agents need to exhibit behavior that is "explicable" to the humans, and be ready to provide "explanations" where needed. I will argue that both explicability and explanations can be understood from the point of view of the differences between the AI agent's model M, and the human partner's mental model of the agent M_h. The agent plans its behavior based on M, but that behavior is viewed by the human partner through the lens of M_h. Explicability in this setup involves the agent making its behavior close to what is expected in terms of M_h. Explanations can be formalized as a dialog between the agent and human intended to thet M_h closer to M. I will discuss the realization of this set-up in our ongoing work on human-robot teaming.
- **Bio**: Subbarao Kambhampati (Rao) is a professor of Computer Science at Arizona State University, and is the current president of the Association for the Advancement of AI (AAAI), and a trustee of the Partnership for AI. His research focuses on automated planning and decision making, especially in the context of human-aware AI systems. He is an award-winning teacher and spends significant time pondering the public perceptions and societal impacts of AI. He was an NSF young investigator, and is a fellow of AAAI. He served the AI community in multiple roles, including as the program chair for IJCAI 2016 and program co-chair for AAAI 2005. Rao received his bachelor's degree from Indian Institute of Technology, Madras, and his PhD from University of Maryland, College Park. More information can be found at rakaposhi.eas.asu.edu.

Freddy Lecue (Accenture Technology Labs, Dublin (Ireland) & INRIA (France))

- *Beyond Machine Learning: Delivering Technology for People through Explainable AI*
- **Abstract**: Machine learning and its models have been largely studied to derive efficient solutions for problems ranging from regression, classification to clustering. However, explaining such models and their underlying predictions remains an open problem, mainly due to the model complexity and interpretability. This work presents our journey towards Explainable AI (i.e., how systematically decoding and enriching machine learning systems with knowledge graphs; and applying such learning and reasoning systems to explain abnormal items in the contexts of (1) finance from 80,000,000+ travel expenses lines of 191,346 employees, and (2) contract risks from 600,000+ clients deals in Accenture). Our semantics-aware travel expenses reasoning system has demonstrated scalability and accuracy for the tasks of explaining abnormalities at large scale.
- **Bio**: Dr Freddy Lecue (PhD 2008, Habilitation 2015) is a principal scientist and research manager in large scale reasoning Systems in Accenture Technology Labs, Dublin - Ireland. He is also a research associate at INRIA, in WIMMICS, Sophia Antipolis - France. His research area is at the frontier of learning and reasoning systems with a strong interest in semantics-driven explainable AI. Before joining Accenture as a principal scientist and research manager in large scale reasoning system in January 2016, he was a research scientist and lead investigator in large scale reasoning systems at IBM Research - Ireland. His research has received IBM internal recognition: IBM research division award in 2015 and IBM Technical Accomplishment award in 2014. His research received external recognition: best paper awards from ISWC (International Semantic Web Conference) in 2014, and ESWC (Extended Semantic Web Conference) in 2014, as well as semantic Web challenge awards from ISWC in 2013 and 2012. Prior to joining IBM Research he was Research Fellow at The University of Manchester from 2008 to 2011 and Research Engineer at Orange Labs (formerly France Telecom R&D) from 2005 to 2008. He received his Research Habilitation (HdR - Accreditation to supervise research) from the University of Nice (France) in 2015, and a PhD from École des Mines de Saint-Etienne (France) in 2008. His PhD thesis was sponsored by Orange Labs and was awarded by the French Association in Artificial Intelligence.

Daniele Magazzeni (King's College London, UK)

- *Explainable Planning*
- **Abstract**: As AI is increasingly being adopted into application solutions, the challenge of supporting interaction with humans is becoming more apparent. Partly this is to support integrated working styles, in which humans and intelligent systems cooperate in problem-solving, but also it is a necessary step in the process of building trust as humans migrate greater responsibility to such systems. The challenge is to find effective ways to communicate the foundations of AI-driven behaviour, when the algorithms that drive it are far from transparent to humans. In this talk we consider the opportunities that arise in AI planning, exploiting the model-based representations that form a familiar and common basis for communication with users, while acknowledging the gap between planning algorithms and human problem-solving.
- **Bio**: Dr. Daniele Magazzeni is Lecturer in Artificial Intelligence at King's College London. His research explores the links between Artificial Intelligence and Verification, and the use of AI in innovative applications. Magazzeni is an elected member of ICAPS Executive Council. He is Editor-in-Chief of AI Communications. He was Conference Chair of ICAPS 2016, is Workshop Chair of IJCAI 2017, and will be chair of the Robotics track at ICAPS 2018. He is co-investigator in UK and EU projects. Daniele is scientific advisor and has collaborations and consultancy projects with a number of companies and organisations.

Darryn Reid (DSTG Edinburgh, Australia)

- *Uncertainty, Resource Allocation, and Trust*
- **Abstract**: Decision making involves resource allocation under uncertainty, whereby the consequences of choices cannot be predicted or controlled. This is of particular concern to military operations, where (due to this uncertainty) there is the possibility of unforeseeable failure and grave consequences. Uncertainty requires awareness (the construction of rational beliefs) and autonomous decision-making can be framed as a set of economic problems. Economic notions of uncertainty, which contrast with assumptions underpinning much of AI, can be connected through ergodic theory and nonlinear dynamics to incompleteness phenomena. This foundation identifies types of uncertainty and provides a formal basis for methods that reason with them (by shifting problem choices rather than by refining solution methods). In this talk, I will argue that, for an intelligent agent to operate successfully in complex environments, it will need to reason at multiple abstraction levels and timescales, will benefit from a theory of self to drive its resource allocation decisions, and thus needs an ability to explain its actions to itself.
- **Bio**: Dr. Reid is Principal Scientist in the Defence Science and Technology Group in the areas of autonomy, behaviour and control, and has been with DSTG since 1995. He has worked in distributed systems, ML and AI, interoperability, formal reasoning and logics, operations research, simulation, optimisation and optimal control, electronic warfare, intelligence analysis, missile targeting and control, command support systems, complexity, nonlinear dynamics and ergodic theory, web-based technologies, software development, functional languages, formal languages and model theory, theory of computation and algorithmic information theory, crowd modelling, economic theory and military theory. He holds a Doctor of Philosophy in Theoretical Computer Science from the University of Queensland. He has strong research interests in pure and applied mathematics, theoretical and applied computer science, philosophy, military theory and economics.

Raymond Sheh (Curtin University, Australia)

- *Why Explain?*
- **Abstract**: Explainable AI (XAI) is enjoying renewed focus from both the research community as well as broader society. Matching up the growing body of techniques for XAI with its applications benefits from an understanding of the contrasting reasons for why explanations are desired in different situations. We present a way of categorising different approaches to XAI from the perspective of why the explanations are required. Such a classification informs not only provides the research community with a framework for discussing XAI but also helps the broader community to better specify their goals in demanding explanations, and to understand the explanations that they are provided with.
- **Bio**: Dr Raymond Sheh is a Senior Lecturer at the Department of Computing, Curtin University. He specialises in the areas of Artificial Intelligence, Robotics and Cyber Security. He has been involved in robotics research since 2003. Dr Sheh established the Intelligent Robots Group at the Department of Computing, with the aim of developing ways of allowing robots and other intelligent systems to learn about their environments and tasks in a way that not only allows them to perform those tasks better, but to also explain their actions and justify their decisions. This ability has significant implications for issues of safety and trust between humans and intelligent systems.

  Dr Sheh's current activities include robotics for hazardous environments, surgical robotics and the application of robotic sensing technologies to industrial automation. The former includes a significant education and research outreach component to other universities and high schools through Dr Sheh's position on the Executive Committee of the International RoboCup Rescue Robot League competition.

  Prior to joining Curtin in 2013, Dr Sheh was with the US National Institute of Standards and Technology, developing standardised test methods for response robots, used in hazardous environments. He holds a PhD in Artificial Intelligence from The University of New South Wales, an Honours degree in Electronic and Communications Engineering from Curtin University and a degree in Computer Science from Curtin University.

Barry Smyth (University College Dublin, Ireland)

- *Explainable Recommendations*
- **Abstract**: Recommender systems are now a near ubiquitous part of the online landscape, influencing the news and books that we read, the music we listen to, the movies we watch, and even the people we date. In this talk we will consider the role of explanation in modern recommender systems, starting with the conventional approach of providing explanations as a way to justify suggestions to a user. But we will take this a step further, by arguing for a more intimate connection between explanations and recommendations, one that sees explanations playing a more central, formative role in the selection and ranking of recommendations.
- **Bio**: Barry is a professor of computer science at University College Dublin in Ireland. His research interests include recommender systems, artificial intelligence, case-based reasoning, information retrieval and the sensor web. He is a Founding Director of the Insight Centre for Data Analytics. You can find out more about his research through his publication list and the people he works with.

  Barry has a keen interest in the commercialisation of research. To this end he has co-founded a number of startups - ChangingWorlds Ltd (now a division of Amdocs) and HeyStaks Technologies Ltd - and he also advises and serves on the boards of a number of organisations.

Mohan Sridharan (University of Auckland, New Zealand)

- *Knowledge Representation and Reasoning Challenges in Explainable Agency*
- **Abstract**: Hardware and software systems are increasingly being used to automate tasks such as sensing, actuation and decision making in a variety of application domains. Before they can be used more widely, these systems must inspire trust in humans. One way to do that is to enable these systems to explain their decisions to humans and justify the choices made under different circumstances. We refer to this ability as *Explainable Agency*. This talk will describe explainable agency, its functional attributes, and the key elements in an architecture for such explainable agents. We will primarily focus on the underlying knowledge representation, reasoning and interactive learning problems, and describe results of some initial work.
- **Bio**: Mohan Sridharan is a Senior Lecturer of Electrical and Computer Engineering at The University of Auckland (NZ). Prior to his current appointment, he was a faculty member at Texas Tech University (USA), where he is currently an Adjunct Associate Professor of Mathematics and Statistics. Dr. Sridharan received his Ph.D. in Electrical and Computer Engineering from The University of Texas at Austin (USA), and was a Research Fellow in the School of Computer Science at the University of Birmingham (UK). His current research interests include knowledge representation and reasoning, machine learning, computational vision, and cognitive systems, as applied to robots and software agents collaborating with humans.

Explanation and Justification in Machine Learning: A Survey

Or Biran
n-Join
or@n-join.com

Courtenay Cotton
n-Join
courtenay@n-join.com

Abstract

We present a survey of the research concerning explanation and justification in the Machine Learning literature and several adjacent fields. Within Machine Learning, we differentiate between two main branches of current research: interpretable models, and prediction interpretation and justification.

1 Introduction

A key component of an artificially intelligent system is the ability to *explain* the decisions, recommendations, predictions or actions made by it and the process through which they are made. Explanation is closely related to the concept of *interpretability*: systems are interpretable if their operations can be understood by a human, either through introspection or through a produced explanation. In the case of machine learning models, explanation is often a difficult task since most models are not readily interpretable. A related concept is *justification*: intuitively, a justification explains why a decision is a good one, but it may or may not do so by explaining exactly how it was made. Unlike introspective explanations, justifications can be produced for non-interpretable systems.

Explanation has been shown to be important for user acceptance and satisfaction in a number of studies. In one early study, physicians rated the ability to explain decisions as the most highly desirable feature of a decision-assisting system [Teach and Shortliffe, 1981]. [Ye and Johnson, 1995] experimented with three types of explanations for an expert system - trace, justification and strategy - and found that explanations in general and justifications in particular make the generated advice more acceptable to users, and that justification (defined as showing the rationale behind each step in the decision) was the most effective type of explanation in changing users' attitudes towards the system. Later studies that empirically tested the importance of explanation to users, in various fields, consistently showed that explanations significantly increase users' confidence and trust [Herlocker *et al.*, 2000; Sinha and Swearingen, 2002; Bilgic and Mooney, 2005; Symeonidis *et al.*, 2009] as well as their ability to correctly assess whether a prediction is accurate [Kim *et al.*, 2016; Gkatzia *et al.*, 2016; Biran and McKeown, 2017].

2 History and Adjacent Fields

Work on producing explanations comes from multiple fields. In this section, we focus on historical background and current work in fields adjacent to machine learning.

2.1 Expert Systems and Bayesian Networks

Historically, explanations first appeared in the context of rule-based expert systems, and were mostly treated as a systems design task (i.e., the task of designing a system capable of producing drill-down into its decisions). The need for explaining the decisions of expert systems was discussed as early as the 1970's [Shortliffe and Buchanan, 1975]. [Swartout, 1983] described a framework for creating expert systems with explanation capabilities, and was one of the first to stress the importance of explanations that are not merely traces, but also contain justifications. [Swartout *et al.*, 1991] is a later example of such a framework. Both were exclusively for rule-based systems and relied on a domain-specific taxonomic knowledge base and a separate strategic knowledge base. [Barzilay *et al.*, 1998] further separated the knowledge into three layers, adding the *communication* layer to the previously described *domain* and *strategic* layers. Separating the communication layer from the rest of the system was intended to allow a communication expert to create solutions that were independent of the specific system and domain.

In some domains probabilistic decision-making systems, often based on Bayesian Networks (BN), are still referred to as expert systems and regarded as successors of earlier rule-based systems. The (scarce) work on explanation for these BN systems self-describes as expert systems explanation. [Lacave and Díez, 2002] present a survey of methods of explanation for Bayesian networks and an excellent analysis of the methods in terms of several properties of explanation. Of particular interest is their classification of the focus of explanation into an explanation of the *reasoning*, the *model*, and the *evidence* for the decision. Most work on explanation in Bayesian networks has been within the narrow context of a particular system, and relies on producing canned text showing the actual posterior probabilities of each node and providing no explanation for what the nodes themselves symbolize, assuming that their names are enough (individual nodes are often symptoms, in the medical domain, or physical evidence, e.g. "valve open", in other domains) [Druzdzel, 1996; Haddawy *et al.*, 1997; Yap *et al.*, 2008].

2.2 Recommender Systems

Recommender systems are online services that serve a large number of users and provide individualized recommendations for media or products. It is usually desirable to produce a short and intuitive justification to help the users decide whether to follow the recommendation or not.

[Herlocker *et al.*, 2000] conducted an experiment measuring user satisfaction with a variety of justification types for a collaborative filtering (neighbor-based) movie recommendation system. They found that the most satisfying were simple and conclusive methods, such as stating the neighbors' ratings or showing one strong feature like a favorite actor. Justifications using ML concepts such as model confidence and complex justifications such as a full neighbor graph scored significantly lower. Regardless of type, 86% of users wanted the justifications they were shown added to the system. Other studies from the early 2000's have also shown that users are overwhelmingly more satisfied with systems that contain some form of justification [Sinha and Swearingen, 2002].

[Symeonidis *et al.*, 2009] presented a style of justification that focused on the most important feature along with the user's past history with regards to that feature. A user study showed that this justification style was significantly more satisfying to users than previous methods. [Papadimitriou *et al.*, 2012] defined a classification of recommender system explanations into three types: those based on previous items chosen by the user, those based on choices of similar users, and those based on features. They also defined a hybrid type which combines two or more of the above, and following a user study concluded that feature-based explanations were the best of the three core types, and that hybrid explanations were best overall. [Bilgic and Mooney, 2005] noted that previous studies have often evaluated the persuasiveness of the justification and not its justifiability. Their experiments showed that for justifiability, feature-based justifications were superior to neighbor-based and user-history-based ones.

2.3 Other adjacent Fields

Generating explanations for users has also been explored in *constraint programming*, specifically for problems where the user may have an interactive role in solving the problem and therefore needs to understand why certain choices were made. In [Freuder *et al.*, 2001] and [Wallace and Freuder, 2001] the authors explored a method of presenting explanations for any assignment decision made by their program. An assignment can be explained by identifying a set of previous assignments that form a sufficient basis to justify the current one. Applying this at each subsequent assignment step forms what the authors call an explanation tree.

Context-aware systems are those that are capable of sensing environmental changes and responding to them. Some work has been done on providing users with explanations for the behavior of these systems. [Tullio *et al.*, 2007] studied mental models that users developed of a system for predicting their managers' interruptibility. However, they concluded that the low level feature contributions that they presented to users were only moderately helpful in improving users understanding of the system, and recommended using higher level concepts instead. [Lim and Dey, 2010] developed a toolkit for use in context-aware applications that provides eight types of explanation for four of the most common model types (rules, decision trees, naïve Bayes and HMMs).

There has been some work on explanation of *Markov Decision Processes* (MDPs). In the context of a particular state in a MDP, it is sometimes desirable to explain to a user what is the best current course of action and why. [Elizalde *et al.*, 2007] describe an explanation system that assists plant operators in executing necessary operations; [Khan *et al.*, 2009] explore the minimal explanation sufficient for tasks such as picking the next course in a college curriculum; [Dodson *et al.*, 2011] propose a dialog system, instead of a single fixed explanation, which allows the user to argue and ask questions.

The *case-based reasoning* community has also explored explanation of probabilistic systems. One example is [Nugent *et al.*, 2009] who proposed a case-based method of explanation for decision support systems, where alternative samples are selected and the explanation focuses on how they differ (if the decision is different) or their similarities (otherwise).

Causal discovery is concerned with determining the direction of causality between variables in a model, which can help explain the behavior of the model. [Hoyer *et al.*, 2009] exploited both non-linearity and non-Gaussianity of real data to identify causality between variables, even in the presence of additive noise. Demonstrating causal relationships is useful for justifying predictions based on these models to users.

In *forensic science*, [Vlek *et al.*, 2016] explained legal cases by combining Bayesian networks with a narrative idiom they call a scenario, taking statistical evidence into account while also maintaining a narrative framework which helps a judge or jury understand the assumptions being made and the relationships among them. This allows insight into the structure of the statistical model, which is crucial for humans to make an informed decision in a legal case. [Timmer *et al.*, 2017] used a somewhat similar approach to generate explanations for Bayesian networks in legal cases. Their work relies on defining a support graph directed toward the variable of interest, then using it to construct an argument.

Interpretability has also been studied in the context of *communicating agents*. [Lazaridou *et al.*, 2017] experimented with neural agents which learn to communicate with each other about images. They then leverage a human-supervised task to ground the learned communication in a way that would be understandable to humans. [Andreas *et al.*, 2017] also studied messages passed between agents in systems with learned deep communicating policies. They developed a strategy for translating these messages into natural language based on the underlying beliefs implied by messages. This is similar to understanding the beliefs implied by any model.

A field that is particularly closely related to explanation is Natural Language Generation (NLG). Much of the work discussed in this survey uses NLG (of varying sophistication) to produce explanations. In addition to explaining ML and other AI systems, however, there has been work on explanations of other kinds. For example, [Pace and Rosner, 2014] produce explanations of user interactions with a software system, intended for administrators, while [Gkatzia *et al.*, 2016] explain how a weather forecast was produced and show that it helps readers decide whether or not to believe the forecast.

Other related work includes [McGuinness and Borgida, 1995], who proposed generating explanations as a debugging tool for the developers and users of a Description Logic-based system. They first break down inference rules into atomic descriptions; corresponding atomic explanations are then created using subsumption rules, and chained to form proofs supporting the system's conclusions.

2.4 Theoretical Work

There has also been some theoretical work on explanation. [Chajewska and Halpern, 1997] proposed a formal definition of explanation in general probabilistic systems, after examining two contemporary ideas and finding them incomplete. In expert systems, [Johnson and Johnson, 1993] presented a short survey of accounts of explanation in philosophy, psychology and cognitive science and found that they fall into three categories: associations between antecedent and consequent; contrasts and differences; and causal mechanisms. In recommender systems, [Yetim, 2008] proposed a framework of justifications which uses existing models of argument to enumerate the components of a justification and provide a taxonomy of justification types. [Corfield, 2010] aims to formalize justifications for the accuracy of ML models by classifying them into four types of reasonings, two based on absolute performance and two rooted in Bayesian ideas.

More recently, [Doshi-Velez and Kim, 2017] considered how to evaluate human *interpretability* of machine learning models. They proposed a taxonomy of three approaches: application-grounded, which judges explanations based on how much they assist humans in performing a real task; human-grounded, which judges explanations based on human preference or ability to reason about a model from the explanation; and functionally-grounded, which judges explanations without human input, based on some formal proxy for interpretability. For this third approach, they hypothesized that matrix factorization of result data (quantized by domain and method) may be useful for identifying common latent factors that influence interpretability.

3 Machine Learning

In the machine learning literature, early work on explanation often focused on producing visualizations of the prediction in order to assist machine learning experts in evaluating the correctness of the model. One very common visualization technique is *nomograms*. It was first applied to logistic regression models by [Lubsen *et al.*, 1978], and later to Naive Bayes [Možina *et al.*, 2004], SVM [Jakulin *et al.*, 2005] and other models. [Szafron *et al.*, 2003] proposed a visualization-based explanation framework for Naive Bayes classifiers.

More recently, visualization techniques have focused on visualizing the hidden states of neural models [Tzeng and Ma, 2005], most notably of Convolutional Neural Nets (CNNs) in image classification [Simonyan *et al.*, 2013; Zeiler and Fergus, 2013] and of Recurrent Neural Nets (RNNs) in Natural Language Processing (NLP) applications [Karpathy *et al.*, 2015; Li *et al.*, 2016; Strobelt *et al.*, 2016].

Beyond visualization, research has focused on two broad approaches to explanation. The first is *prediction interpretation and justification*, where a (usually non-interpretable) model and prediction are given, and a justification for the prediction must be produced. The second is *interpretable models*, which aims to devise models that are intrinsically interpretable and can be explained through reasoning.

3.1 Prediction Interpretation and Justification

This approach has focused on interpreting the predictions of complex models, often by proposing to isolate the contributions of individual features to the prediction. Such proposals were made for Bayesian networks [Suermondt, 1992], multilayer Perceptrons [Feraud and Clerot, 2002], RBF networks [Robnik-Šikonja *et al.*, 2011] and general hierarchical networks [Landecker *et al.*, 2013]. [Martens *et al.*, 2008] proposed to interpret the predictions of an SVM classifier by extracting conjunctive rules using a small subset of features.

In addition to model-specific methods, there have been a few suggestions for model-agnostic frameworks. [Robnik-Šikonja and Kononenko, 2008] proposed measuring the effect of an individual feature on an unknown classifier's prediction by checking what the prediction would have been if that feature value was absent and comparing the two using various distance measures. The effects are then displayed visually to explain the main contributors towards a prediction or to compare the effect of the feature in various models. This method was extended to include regression models in [Kononenko *et al.*, 2013]. [Baehrens *et al.*, 2010] described an alternative approach using *explanation vectors* (class probability gradients) which highlight the effect of the most important features.

Other work, especially in the NLP literature, has focused on using a small portion of input as evidence to justify the prediction result, and often explored alternative definitions of evidence and styles of explanation. [Martens and Provost, 2014] describe a framework of linguistic explanations for document classification with bag-of-words features. Their method shows removal-based explanations of the type "the classification would change to [alternative class] if the words [list of words] were removed from the document", which can help a domain expert intuitively assess how solid the prediction is. [Kim *et al.*, 2016] select two subset of training samples: *prototypes* - samples of different types that the model represents well; and *criticisms* - samples that are most misrepresented by the model. They show that this model-level explanation makes users more likely to correctly predict the model's success with new samples. [Lei *et al.*, 2016] select small snippets of the input text of text classification tasks as justification for the decision. The justification model is separate from the prediction model, but trained on the same data, with the constraint that the prediction of the main model for the (much shorter) justification should be very similar to that for the full text. [Biran and McKeown, 2017] define evidence as the intersection of a feature's actual contribution and expected contribution, and categorize features that are important to the prediction based on that definition. Their work therefore shows not only actual evidence but also *missing evidence*, an important part of human reasoning, and differentiates between expected and unexpected evidence.

Work on *model approximation* focuses on deriving a simple, interpretable model (such as a shallow decision tree, rule list, or sparse linear model) that approximates a more com-

plex, uninterpretable one (e.g., a neural net). Early work described approximations of the entire model [Thrun, 1995; Craven and Shavlik, 1999]; the disadvantage of these approaches is that for even moderately complex models, a good global approximation cannot generally be found. [Ribeiro *et al.*, 2016] introduce an approach that focuses on local approximations, which behave similarly to the global model only in the vicinity of a particular prediction. Their algorithm is agnostic to the details of the original model.

In image classification, there has been work on secondary neural models, inspired by neural caption generation, that learn to generate textual justifications for classifications of the primary neural model. [Hendricks *et al.*, 2016] use an LSTM caption generation model with a loss function that encourages class discriminative information to generate justifications for the image classification of a CNN. [Park *et al.*, 2016] produce both a textual justification and a visual attention map, making their approach a combination of an interpretable model (See Section 3.2) and an external justification model. [Vedantam *et al.*, 2017] produce captions that are locally discriminative, in the context of other images.

3.2 Interpretable Models

An alternative to methods for interpreting or justifying otherwise black-box models is to produce models that are inherently interpretable. One family of models that are readily interpretable by humans are shallow rule-based models: decision lists and decision trees. [Rudin *et al.*, 2013] introduced classifiers that use association rules [Agrawal *et al.*, 1993], which can be learned efficiently from sparse data. This family of models includes Bayesian Rule Lists [Letham *et al.*, 2015], an algorithm that generates a posterior distribution of decision lists that encourages sparsity as well as accuracy; Bayesian Or's of And's [Wang *et al.*, 2015], highly efficient disjunctive rule lists; and Falling Rule Lists [Wang and Rudin, 2015], where the order of rules implies both domain-level importance and estimated probability of success. Other approaches have focused on creating sparse models via features selection or extraction that aims to optimize interpretability. Examples include Supersparse Linear Integer Models [Ustun and Rudin, 2016] and Mind-the-Gap Model [Kim *et al.*, 2015].

In deep learning, attention mechanisms which allow a model to focus on a subset of its vector representation were found to improve accuracy on many tasks, particularly within NLP [Bahdanau *et al.*, 2014] and image classification [Xu *et al.*, 2015], and at the same time result in significantly more intuitive hidden states that appear semantically appropriate to humans when inspected.

Finally, there has been some work on *compositional generative models* which are constrained or encouraged to learn hierarchical, semantically meaningful representations of data. [Si and Zhu, 2013] learn compositional models of objects in images: an object (e.g., a cat) contains a mandatory set of parts (e.g., ears), but each part can come in many forms (pointed, round...), and each form is represented using a pixel-level generative model. [Lake *et al.*, 2015] learn a generative model of linguistic character images from sparse data. Their approach infers motor programs (line strokes) from sample images and learns a prior generative model of generative models, which can then produce a reasonable model from even one sample of a new character.

4 Conclusion

While eXplainable AI (XAI) is only now gaining widespread visibility, the ML literature and that of allied fields contain a long, continuous history of work on explanation and can provide a pool of ideas for researchers currently tackling the task of explanation. Despite this history, current efforts face unprecedented difficulties: contemporary models are more complex and less interpretable than ever; they are used for a wider array of tasks, and are more pervasive in everyday life than in the past; and they are increasingly allowed to make (and take) more autonomous decisions (and actions). Justifying these decisions will only become more crucial, and there is little doubt that this field will continue to rise in prominence and produce exciting and much needed work in the future.

References

[Agrawal *et al.*, 1993] R Agrawal, T Imieliński, and A Swami. Mining association rules between sets of items in large databases. *SIGMOD*, 22(2):207–216, June 1993.

[Andreas *et al.*, 2017] Jacob Andreas, Anca Dragan, and Dan Klein. Translating neuralese. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2017.

[Baehrens *et al.*, 2010] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *JMLR*, 11, August 2010.

[Bahdanau *et al.*, 2014] D Bahdanau, K Cho, and Y Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[Barzilay *et al.*, 1998] Regina Barzilay, Daryl Mccullough, Owen Rambow, Jonathan DeCristofaro, Tanya Korelsky, and Benoit Lavoie. A new approach to expert system explanations. In *International Workshop on NLG*, 1998.

[Bilgic and Mooney, 2005] Mustafa Bilgic and Raymond J. Mooney. Explaining recommendations: Satisfaction vs. promotion. In *Workshop on the Next Stage of Recommender Systems Research*, San Diego, CA, 2005.

[Biran and McKeown, 2017] Or Biran and Kathleen McKeown. Human-centric justification of machine learning predictions. In *IJCAI*, Melbourne, Australia, 2017.

[Chajewska and Halpern, 1997] U Chajewska and J Y Halpern. Defining explanation in probabilistic systems. In *Uncertainty in artificial intelligence*, 1997.

[Corfield, 2010] David Corfield. Varieties of justification in machine learning. *Minds and Machines*, 20(2):291–301, 7 2010.

[Craven and Shavlik, 1999] Mark Craven and Jude Shavlik. Rule extraction: Where do we go from here?, 1999.

[Dodson *et al.*, 2011] Thomas Dodson, Nicholas Mattei, and Judy Goldsmith. A natural language argumentation interface for explanation generation in markov decision processes. In *Algorithmic Decision Theory*, 2011.

[Doshi-Velez and Kim, 2017] Finale Doshi-Velez and Been Kim. A roadmap for a rigorous science of interpretability. *arXiv preprint arXiv:1702.08608*, 2017.

[Druzdzel, 1996] Marek J Druzdzel. Qualitiative verbal explanations in bayesian belief networks. *AISB QUARTERLY*, pages 43–54, 1996.

[Elizalde *et al.*, 2007] F. Elizalde, L. E. Sucar, A. Reyes, and P. deBuen. An MDP approach for explanation generation. In *Explanation-Aware Computing Workshop at AAAI*, pages 28–33, Vancouver, BC, Canada, 2007.

[Feraud and Clerot, 2002] Raphael Feraud and Fabrice Clerot. A methodology to explain neural network classification. *Neural Networks*, 15(2):237 – 246, 2002.

[Freuder *et al.*, 2001] E. Freuder, C. Likitvivatanavong, and R. Wallace. Deriving explanations and implications for constraint satisfaction problems. In *Principles and Practice of CP*, pages 585–589. Springer, 2001.

[Gkatzia *et al.*, 2016] D Gkatzia, O Lemon, and V Rieser. Natural language generation enhances human decision-making with uncertain information. In *ACL*, 2016.

[Haddawy *et al.*, 1997] P. Haddawy, J. Jacobson, and C. E. Kahn. BANTER: a Bayesian network tutoring shell. *Artificial Intelligence in Medicine*, 10(2):177–200, June 1997.

[Hendricks *et al.*, 2016] L.A Hendricks, Z Akata, M Rohrbach, J Donahue, B Schiele, and T Darrell. Generating visual explanations. In *ECCV*, 2016.

[Herlocker *et al.*, 2000] J Herlocker, J Konstan, and J Riedl. Explaining collaborative filtering recommendations. In *Computer Supported Cooperative Work (CSCW)*, 2000.

[Hoyer *et al.*, 2009] P Hoyer, D Janzing, J Mooij, J Peters, and B Schölkopf. Nonlinear causal discovery with additive noise models. In *NIPS*, 2009.

[Jakulin *et al.*, 2005] A Jakulin, M Možina, J Demšar, I Bratko, and B Zupan. Nomograms for visualizing support vector machines. In *KDD*, 2005.

[Johnson and Johnson, 1993] H. Johnson and P. Johnson. Explanation facilities and interactive systems. In *IUI*, pages 159–166, New York, NY, USA, 1993.

[Karpathy *et al.*, 2015] Andrej Karpathy, Justin Johnson, and Fei-Fei Li. Visualizing and understanding recurrent networks. *CoRR*, abs/1506.02078, 2015.

[Khan *et al.*, 2009] Omar Zia Khan, Pascal Poupart, and James P. Black. Minimal sufficient explanations for factored markov decision processes. In *ICAPS*, 2009.

[Kim *et al.*, 2015] Been Kim, Julie Shah, and Finale Doshi-Velez. Mind the gap: A generative approach to interpretable feature selection and extraction. In *NIPS*, 2015.

[Kim *et al.*, 2016] Been Kim, Rajiv Khanna, and Oluwasanmi Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *NIPS*. 2016.

[Kononenko *et al.*, 2013] Igor Kononenko, Erik Strumbelj, Zoran Bosnic, Darko Pevec, Matjaz Kukar, and Marko Robnik-Šikonja. Explanation and reliability of individual predictions. *Informatica (Slovenia)*, 37(1):41–48, 2013.

[Lacave and Díez, 2002] C. Lacave and F. J. Díez. A review of explanation methods for Bayesian networks. *Knowledge Engineering Review*, 17:107–127, 2002.

[Lake *et al.*, 2015] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, December 2015.

[Landecker *et al.*, 2013] W. Landecker, M.D. Thomure, L.M.A. Bettencourt, M. Mitchell, G.T. Kenyon, and S.P. Brumby. Interpreting individual classifications of hierarchical networks. In *CIDM*, 2013.

[Lazaridou *et al.*, 2017] A Lazaridou, A Peysakhovich, and M Baroni. Multi-agent cooperation and the emergence of (natural) language. In *ICLR*, Toulon, France, 2017.

[Lei *et al.*, 2016] T Lei, R Barzilay, and T.S Jaakkola. Rationalizing neural predictions. In *EMNLP*, 2016.

[Letham *et al.*, 2015] B Letham, C Rudin, T.H McCormick, and D Madigan. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *CoRR*, abs/1511.01644, 2015.

[Li *et al.*, 2016] Jiwei Li, Xinlei Chen, Eduard H. Hovy, and Dan Jurafsky. Visualizing and understanding neural models in NLP. In *NAACL-HLT*, 2016.

[Lim and Dey, 2010] Brian Lim and Anind Dey. Toolkit to support intelligibility in context-aware applications. In *Ubiquitous Computing (Ubicomp)*, 2010.

[Lubsen *et al.*, 1978] J. Lubsen, J. Pool, and E. van der Does. A practical device for the application of a diagnostic or prognostic function. *Methods of information in medicine*, 17(2):127–129, April 1978.

[Martens and Provost, 2014] David Martens and Foster Provost. Explaining data-driven document classifications. *MIS Q.*, 38(1):73–100, March 2014.

[Martens *et al.*, 2008] D Martens, J Huysmans, R Setiono, J Vanthienen, and B Baesens. Rule extraction from support vector machines: An overview of issues and application in credit scoring. In *Rule Extraction from SVMs*, volume 80 of *Studies in Comp. Int.*, pages 33–63. 2008.

[McGuinness and Borgida, 1995] Deborah L McGuinness and Alexander Borgida. Explaining subsumption in description logics. In *IJCAI (1)*, pages 816–821, 1995.

[Možina *et al.*, 2004] M. Možina, J. Demšar, M. Kattan, and B. Zupan. Nomograms for visualization of naive bayesian classifier. In *PKDD*, 2004.

[Nugent *et al.*, 2009] Conor Nugent, Dónal Doyle, and Pádraig Cunningham. Gaining insight through case-based explanation. *J. Intell. Inf. Syst.*, 32(3):267–295, June 2009.

[Pace and Rosner, 2014] Gordon J. Pace and Michael Rosner. *Explaining Violation Traces with Finite State Natural Language Generation Models*, pages 179–189. Springer International Publishing, 2014.

[Papadimitriou *et al.*, 2012] A. Papadimitriou, P. Symeonidis, and Y. Manolopoulos. A generalized taxonomy of

explanations styles for traditional and social recommender systems. *Data Min. Knowl. Discov.*, 24(3):555–583, 2012.

[Park *et al.*, 2016] D H Park, L A Hendricks, Z Akata, B Schiele, T Darrell, and M Rohrbach. Attentive explanations: Justifying decisions and pointing to the evidence. *CoRR*, abs/1612.04757, 2016.

[Ribeiro *et al.*, 2016] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *KDD*, 2016.

[Robnik-Šikonja and Kononenko, 2008] M. Robnik-Šikonja and I. Kononenko. Explaining classifications for individual instances. *TKDE*, 20(5):589–600, May 2008.

[Robnik-Šikonja *et al.*, 2011] M Robnik-Šikonja, A Likas, C Constantinopoulos, I Kononenko, and E Strumbelj. Efficiently explaining decisions of probabilistic rbf classification networks. In *ICANNGA*, 2011.

[Rudin *et al.*, 2013] C Rudin, B Letham, and D Madigan. Learning theory analysis for association rules and sequential event prediction. *JMLR*, 14:3441–3492, 2013.

[Shortliffe and Buchanan, 1975] Edward H Shortliffe and Bruce G Buchanan. A model of inexact reasoning in medicine. *Mathematical biosciences*, 23(3), 1975.

[Si and Zhu, 2013] Zhangzhang Si and Song-Chun Zhu. Learning and-or templates for object recognition and detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(9):2189–2205, September 2013.

[Simonyan *et al.*, 2013] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013.

[Sinha and Swearingen, 2002] Rashmi Sinha and Kirsten Swearingen. The role of transparency in recommender systems. In *CHI EA*, 2002.

[Strobelt *et al.*, 2016] Hendrik Strobelt, Sebastian Gehrmann, Bernd Huber, Hanspeter Pfister, and Alexander M. Rush. Visual analysis of hidden state dynamics in recurrent neural networks. *CoRR*, abs/1606.07461, 2016.

[Suermondt, 1992] Henri Jacques Suermondt. *Explanation in Bayesian Belief Networks*. PhD thesis, Stanford, CA, USA, 1992. UMI Order No. GAX92-21673.

[Swartout *et al.*, 1991] W Swartout, C Paris, and J Moore. Explanations in knowledge systems: Design for explainable expert systems. *IEEE Expert*, 6(3):58–64, 1991.

[Swartout, 1983] William R. Swartout. Xplain: A system for creating and explaining expert consulting programs. *Artificial Intelligence*, 21(3), September 1983.

[Symeonidis *et al.*, 2009] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. Moviexplain: A recommender system with explanations. In *RecSys*, 2009.

[Szafron *et al.*, 2003] D Szafron, R Greiner, P Lu, D Wishart, C Macdonell, J Anvik, B Poulin, Z Lu, and R Eisner. Explaining naive bayes classifications. Technical report, 2003.

[Teach and Shortliffe, 1981] R. Teach and E. Shortliffe. An Analysis of Physician Attitudes Regarding Computer-Based Clinical Consultation Systems. *Computers and Biomedical Research*, 14:542–558, 1981.

[Thrun, 1995] Sebastian Thrun. Extracting rules from artificial neural networks with distributed representations. In *NIPS*, 1995.

[Timmer *et al.*, 2017] S. Timmer, J. Meyer, H. Prakken, S. Renooij, and B. Verheij. A two-phase method for extracting explanatory arguments from bayesian networks. *Int. J. Approx. Reasoning*, 80(C):475–494, January 2017.

[Tullio *et al.*, 2007] Joe Tullio, Anind Dey, Jason Chalecki, and James Fogarty. How it works: A field study of non-technical users interacting with an intelligent system. In *SIGCHI Human Factors in Computing Systems*, 2007.

[Tzeng and Ma, 2005] F. Y. Tzeng and K. L. Ma. Opening the black box - data driven visualization of neural networks. In *IEEE Visualization*, pages 383–390, 2005.

[Ustun and Rudin, 2016] B Ustun and C Rudin. Supersparse linear integer models for optimized medical scoring systems. *Mach. Learn.*, 102(3):349–391, March 2016.

[Vedantam *et al.*, 2017] Ramakrishna Vedantam, Samy Bengio, Kevin Murphy, Devi Parikh, and Gal Chechik. Context-aware captions from context-agnostic supervision. *CoRR*, abs/1701.02870, 2017.

[Vlek *et al.*, 2016] Charlotte S. Vlek, Henry Prakken, Silja Renooij, and Bart Verheij. A method for explaining bayesian networks for legal evidence with scenarios. *Artificial Intelligence and Law*, 24(3):285–324, 2016.

[Wallace and Freuder, 2001] Richard J Wallace and Eugene C Freuder. Explanations for whom. In *CP01 Workshop on User-Interaction in Constraint Satisfaction*, 2001.

[Wang and Rudin, 2015] Fulton Wang and Cynthia Rudin. Falling rule lists. In *AISTATS*, 2015.

[Wang *et al.*, 2015] T Wang, C Rudin, F Doshi-Velez, Y Liu, E Klampfl, and P MacNeille. Or's of and's for interpretable classification, with application to context-aware recommender systems. *CoRR*, abs/1504.07614, 2015.

[Xu *et al.*, 2015] K Xu, J Ba, R Kiros, K Cho, A Courville, R Salakhudinov, R Zemel, and Y Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, Lille, France, 2015.

[Yap *et al.*, 2008] Ghim-Eng Yap, Ah-Hwee Tan, and Hwee-Hwa Pang. Explaining inferences in bayesian networks. *Applied Intelligence*, 29(3):263–278, 2008.

[Ye and Johnson, 1995] L. Richard Ye and Paul E. Johnson. The impact of explanation facilities on user acceptance of expert systems advice. *MIS Q.*, 19(2):157–172, June 1995.

[Yetim, 2008] Fahri Yetim. A framework for organizing justifications for strategic use in adaptive interaction contexts. In *ECIS*, pages 815–825, 2008.

[Zeiler and Fergus, 2013] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.

A Framework for Explanation of Machine Learning Decisions

Chris Brinton
Mosaic ATM; Leesburg, Virginia; USA
brinton@mosaicatm.com

Abstract

This paper presents two novel techniques to generate explanations of machine learning model results for use in advanced automation-human interaction. The first technique is "Explainable Principal Components Analysis," which creates a framework within a multi-dimensional problem space to support the explainability of model outputs. The second technique is the "Gray-Box Decision Characterization" approach, which probes the output of the machine learning model along the dimensions of the explainable framework. These two techniques are independent of the type of machine learning algorithm. Rather, the intent of these algorithms is to be applicable generally across any type of machine learning algorithm and any application domain of machine learning. The concept and computational steps of each technique are presented in the paper, along with results of experimental implementation and analysis.

1 Introduction

The average person now experiences the results of machine learning (ML) models on a frequent basis through online interaction with news sites that learn reader interests, retail websites that provide automated offers that are customized to individual consumer's buying habits, and credit card fraud detection that warns the card holder when a transaction occurs that is outside of their normal purchasing pattern. Many such applications of machine learning can be performed using an automated approach where human interpretation of the recommendations is not necessary. However, in many other applications, there would be great benefit if the machine learning model could provide an explanation of its output recommendation or other result.

In the realm of many decision support tools for military and other safety- or life-critical applications, it is necessary and appropriate for humans to be involved in decisions using the recommendations and guidance of computer automation and information systems. For example, a surgeon that receives a recommendation from an ML-based medical decision support system is not likely to perform a surgery on a patient based on the recommendation of a computer system alone. Lee and See [2004] provide an extensive description of the need for and methods to achieve user trust in a computer automation system, including the following guidance: "*Show the process and algorithms of the automation by revealing intermediate results in a way that is comprehensible to the operators.*"

Although some ML models can provide limited insight into and explanation of their intermediate results and model outputs, most machine learning model output is opaque. Such opacity can lead users of the technology to doubt the reliability of the information or recommendation that is provided. This lack of understanding of the technology can result in distrust, and to eventual failure of the technology to receive acceptance and use. Even if the technology does receive acceptance and operational use, a machine learning-based information system that can explain itself may allow more efficient and effective use of the technology.

Much previous research has been conducted in explaining ML model output, including [Andrews, et al., 1995; Fung, et al., 2005; Letham, et al., 2012; and Baehrens, et al., 2005]. The innovation that we describe herein is motivated by the explanation approach presented in the Baehrens, et al., [2005] work, but over more complex problem spaces.

2 Explainable Principal Components Analysis

Principal Components Analysis (PCA) [Hotelling, 1933] is a technique that is used extensively within machine learning model development for dimensionality reduction. From an information-theoretic perspective, regular PCA aggregates information contained in a high-dimensional space into a form that can represent an arbitrarily large portion of the information in the data through a lower-dimension vector representation. While this aggregation process identifies the orthogonal dimensions in the data over which the greatest explanation of the variance can be achieved, the "explanation" of the variance in PCA is maximized from a statistical perspective, but not from the perspective of understandability by a human. In fact, the basis vectors created by PCA are one of the primary sources of opacity in many practical machine learning applications. We present herein a formulation of a variant of PCA - which we refer to as Explainable Principal Components Analysis (EPCA) - that computes basis vectors of the problem space with human understandability as a primary objective.

The EPCA technique uses minimal human interaction to identify modes of variation in the input data that can be identified and labeled for use by subsequent explanation algorithms, such as the Gray Box Decision Characterization (GBDC) approach. The EPCA process is performed iteratively, creating one explainable basis vector for the

problem space at a time. After each basis vector is created, the input training samples are projected into a subspace that excludes any contribution from the previously fixed explainable basis vectors. Regular PCA is then run on the training samples in the subspace, and the human uses the results of the regular PCA to inform the design of the next explainable basis vector. The result of this process is a set of basis vectors that are labeled with their meanings in a manner that is intended to be understandable by a human observer of a model decision.

While we have not found this concept in the literature, our EPCA algorithm is motivated by the Kernel Near Principal Components Analysis work [Martin, 2002] and the LASSO Principal Components Analysis concept [Jolliffe, et al., 2003].

Suppose we are given n input training samples, $x_1, \dots, x_n \in \mathbb{R}^d$, for either a classification or regression problem. With any set of orthonormal basis vectors, $u_1, \dots, u_n \in \mathbb{R}^d$ that spans the space of the input training samples, we can represent the input sample data as a linear combination of the basis vectors.

$$x_i = p_{i,1}u_1 + p_{i,2}u_2 + \cdots + p_{i,n}u_n \quad (1)$$

In matrix form for all samples of the training data:

$$X = PU \quad (2)$$

where the *U* matrix is the orthonormal basis, the *X* matrix is the set of all input samples as row vectors, and *P* is the matrix of basis vector coefficients to reconstruct the input, *X*.

To initiate the EPCA procedure, we set the *U* matrix to be the orthonormal basis made up of the eigenvectors as output by a regular PCA process, sorted by the eigenvalues. The model designer then interprets the individual eigenvectors as weighting coefficients on each of the original features in the *x* vectors, and manually identifies the human-understandable concept or combination of features that is generally represented by one of the eigenvectors, favoring those that are associated with a larger eigenvalue. This may indicate that a single feature is the primary contributor to the eigenvector, or perhaps that a combination of a few related features are highlighted in the eigenvector. However, the PCA process likely also results in small, non-zero coefficients on many features in the original feature space, which can obscure the understandability of the eigenvector.

This step in the process requires the model designer to use domain knowledge and other techniques dependent on the unique nature of the problem space to create an interpretable basis vector. Figure 1 shows the eigenvectors generated by the first step in this process for a sample text-analytics problem. For text-analytics problems the Singular Value Decomposition (SVD) is used, rather than PCA, for performance reasons, but the EPCA techniques can be applied analogously to achieve explainable basis vectors.

To generate an explainable vector for this example, words that indicate a distinct concept would be excluded. For example, to distinguish medical doctors and facilities from dentists, the word 'dental' would be excluded.

ei	words
0.295841	patient
0.226496	care
0.156011	health
0.155184	therapy
0.152811	treatment
0.151564	medical
0.145705	dental
0.129402	hospital
0.123112	nurse
0.120978	therapist

ei	words
-0.18745	design
-0.1376	project
-0.12352	designer
-0.11958	graphic
-0.11022	web
-0.10993	business
-0.09552	account
-0.0937	sale
-0.09217	marketing
-0.08637	adobe

Figure 1. Sample words and coefficient values from the first eigenvector for a resume classification problem.

An additional technique that we have used is to identify input data samples that differ from each other in a single, understandable way. A simple linear model can be generated from at least two such input samples that exhibit a single mode of variation in the input feature space to create an explainable basis vector.

The first explainable basis vector, ϕ_1, is created by using these, or other, techniques to identify the input feature vector elements that are to be included as contributors to the first explainable basis vector. All other coefficients in ϕ_1 are set to zero, and ϕ_1 is then normalized to unit length.

To form the remaining explainable basis vectors, the EPCA procedure removes the contribution of the explainable basis vector from each of the input data samples. In linear algebraic terms, the input data samples are projected into the null space of the explainable basis vectors, or orthogonal complement, in the case of a single vector. This modified set of input data samples, X', is then analyzed using regular PCA and the same manual techniques described above to generate a second explainable basis vector.

Although the projection into the null space of the explainable basis vector could be done more directly, the first step that we use in this process is to create a new orthonormal basis that uses ϕ_1 as the first basis vector. This is done by combining ϕ_1 with the original *u* vectors, excluding the *u* vector that was used to form ϕ_1 (or an arbitrary *u* vector if ϕ_1 is not related to any *u* vectors), and then applying the Gram-Schmidt procedure [Wikipedia, 2017] to orthonormalize the basis.

The EPCA process terminates when the model designer is satisfied with the set of explainable basis vectors, or no additional explainable basis vectors can be identified. At this point, the basis matrix is relabeled as Φ, as is declared to be the final orthonormal explainable basis for the problem space.

A very important aspect of using the EPCA approach to establish the basis for explanation of machine learning model decisions is that in addition to obtaining orthogonal dimensions along which the ML decision can be parameterized for explainability, we also obtain explicit measures of the mean and variance of the input data samples along those dimensions. Thus, in the use of the EPCA basis for sensitivity analysis, we can compare the mean value of the entire set of input data along that dimension to the value

of the specific input vector applied to the machine learning model that generated the decision to be explained.

Inverting Equation (2) and replacing the U matrix with the Φ matrix:

$$P = X\Phi^T \quad (3)$$

Since the Φ matrix is orthonormal, its inverse is the same as its transpose. The mean and standard deviation are then computed along the columns of the P matrix to obtain the mean and standard deviation across the entire input data set.

If we are given a new input vector for prediction by the ML model, $z \in \mathbb{R}^d$, the coefficients for each of the explainable basis vectors for this new input vector are computed as:

$$p = z\Phi^T \quad (4)$$

Comparing the p vector from Equation (4) to the column means and standard deviations of the P matrix from Equation (3) provides an immediate, understandable characterization of the input data vector (not the decision, but the input vector). If labels are assigned to the dimensions of the EPCA output basis, textual descriptions could be assigned such as 'the eyes of this face are particularly close to each other,' or 'this flight path uses a particularly long final approach segment.' Although such characterizations of the input data are completely separate from the output of the model, they can provide value to the human model user who desires an explanation of the model decision. The next section, however, describes the method for explanation of the ML model decision itself.

2 Gray Box Decision Characterization

We refer to the second component of our innovation as Gray-Box Decision Characterization. As implied, this approach uses some knowledge of the inner-workings of the machine-learning approach, but does not make changes to the machine learning algorithm itself. Thus, the approach lies in between a black-box and a white-box approach. It is important to note that we refer to this approach as 'decision' characterization, not 'model' characterization. The objective of this technique is to provide an explanation for a single specific output of the machine learning model (at a time), not to provide an explainable characterization of the entire machine learning model's behavior.

The GBDC approach utilizes the results of the EPCA algorithm (or regular PCA if sufficiently explainable) as an orthogonal basis for sensitivity analysis of the output of the machine learning model around the input data vector for a single decision output. This technique simply performs a sensitivity analysis of the behavior of the model in the region of the space around the specific input data feature vector that generated the decision from the machine learning model. The GBDC approach searches for changes along explainable basis vectors that result in a change in the output of the machine learning model. Although this technique is simple in principle, the large number of dimensions (even after dimensionality reduction), and the need to search along each dimension, require additional enhancements beyond the simple concept explanation provided so far. We return to the EPCA approach to describe the additions to the GBDC algorithm.

In addition to using the mean value and standard deviation along each explainable basis vector of the Φ matrix to characterize the input, we also use the standard deviation along each dimension to determine the appropriate step size to use in the sensitivity analysis. For example, if the standard deviation, σ_j, of the coefficients of the input data samples along a particular explainable basis vector, j, is 10, then testing the sensitivity along that dimension by evaluating a change of 30 along that dimension would move the sensitivity test position by 3σ, which would likely be outside of the range of nearly all input samples.

For characterization of a classification machine learning model, the GBDC technique conducts a search to find a change in the output classification of the model. A binary search is used to find the first occurrence of change along that dimension, recognizing that no change may occur at all within the realm of reasonable change values.

Mathematically, this is represented in Equation (5), where $p_{\phi,0}$ represents the mean value of the p coefficient from the input sample data for the ϕ basis vector, y_0 being the label output by the ML model for the test input vector, z, and y being the output of the ML model for the senstivity analysis over coefficients, p:

$$\phi^* = \arg\min_p \frac{|p_\phi - p_{\phi,0}|}{\sigma_\phi}, y \neq y_0, \quad (5)$$

We represent the value of the coefficient that satisfies Equation (5) as p_ϕ^*. Then, the ML model decision can be explained as having the most significant contribution in the ϕ^* explainable basis vector, and that the decision, y_0, was output by the model because the value along the ϕ^* explainable basis vector is less than (or greater than) p_ϕ^*.

For a regression model, the rate of change of the output given a change in the input is calculated.

$$\phi^* = \arg\max_p \frac{\left|\partial y / \partial p_\phi\right|}{\sigma_\phi} \quad (6)$$

In both Equations (5) and (6), additional basis vectors can be selected for use in the explanation if the change in z-score or the partial derivative of the output is below or above a threshold value.

Thus, the GBDC approach provides an explanation of the machine learning model decision by selecting the dimensions that generate the most significant change in the machine learning model output for a regression problem, or that create a change in the classification decision with the smallest change (according to z-score) in the input vector.

3 Results

Experiments have been conducted to study the usefulness and applicability of these two concepts for generating explanations of machine learning model decisions. The research team has implemented the EPCA and GBDC algorithms in prototype software. Initial testing and evaluation of the algorithm has been performed using multiple domains.

In Figure *2*, which shows aircraft arrival tracks for flights entering the Atlanta terminal area from the northwest and landing on Runway 8L at Atlanta Hartsfield International

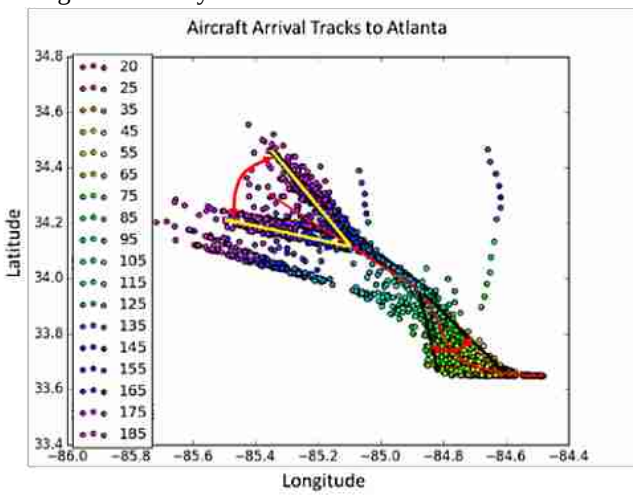


Figure 2. Human-Annotated Explainable Modes of Variation

Airport, some modes of variation are clearly evident through visual analysis of the scatter plot. The points of the scatter plot show individual surveillance positions for 246 flights on a single day. The legend indicates the altitude (in 100s of feet) associated with each color. The figure also shows two primary modes of variation of the data (one in yellow lines and one in black lines) that are clear and understandable to human observation.

In Figure 3 we show the results of both regular PCA and EPCA on this dataset. The top two plots show the first two eigenvectors of the cross-correlation matrix (i.e., regular PCA). Note that the two modes of variation determined by

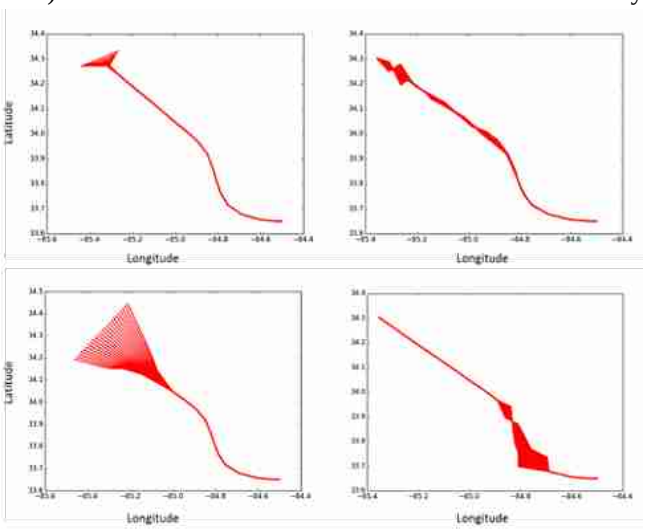


Figure 3. Two Primary Modes of Variation from Regular PCA (top) and EPCA (bottom)

PCA do not closely resemble the two primary modes that can be identified through visual inspection and understood by a human. Using EPCA, two modes of variation were generated that closely resemble the human-annoted modes of variation from Figure *2*.

Finally, we have conducted experiments with the GBDC technique as applied to the Iris dataset. In this case, the input feature vector representation is only 4-dimensional, so we are able to use the four features directly, without applying the EPCA approach first. Table 1 provides sample results that use GBDC to explain the reasons for the model output of one particular data point input to different models. All models gave the same classification, but GBDC found different explanations for each of those classifications.

Table 1. GBDC Results on the Iris Dataset

Model Type	Explanation
Decision Tree	petal width (cm) is greater than 0.79
SVM with Linear Kernel	petal length (cm) is greater than 2.1
SVM with RBF Kernel	sepal length (cm) is greater than 4.3 and petal length (cm) is greater than 2.4
SVM with Polynomial Kernel	petal length (cm) is greater than 1.7

Each of the models learns decision boundaries using different mathematical formulations and parameters. Although the thresholds chosen by GBDC to explain the decision are different in each case, they are still consistent with each other. It is also important to note that this example was selected because of the more significant differences in explanation than many other cases that were tested.

4 Conclusions

The Explainable Principal Components Analysis and Gray-Box Decision Characterization techniques provide a useful framework for analysis and explanation of machine learning model decisions. Human involvement in establishing the framework is required, but the required additional work is performed during design of the model. We have demonstrated the use of the techniques on multiple problem domains. However, additional research is needed to address more complicated problem domains, and to evaluate the feasibility and effectiveness of identifying explainable basis vectors in such problem domains.

Explainability cannot be fully evaluated without including human participants to evaluate the usefulness of the explanations generated. Our future work will include such considerations, as well as the question of whether or not a single set of basis vectors is appropriate for all users of the model, or if different people or types of users would actually need a different set of basis vectors to achieve adequate explanations.

References

[Lee and See, 2004] Lee, J.D., and See, K.A., "Trust in Automation: Designing for Appropriate Reliance," HUMAN FACTORS, Vol. 46, No. 1, Spring 2004, pp. 50–80.

[Andrews, et al., 1995] Andrews, R., Diederich, J., and Tickle A., "A survey and critique of techniques for extracting rules from trained artificial neural networks." Knowledge-Based Systems, 8:373–389, 1995.

[Letham, et al., 2012] Letham, B., Rudin, C., McCormick, T. H., and Madigan, D., "Building Interpretable Classifiers with Rules using Bayesian Analysis," Technical Report no. 609, Department of Statistics, University of Washington, December, 2012.

[Baehrens, et al., 2010] Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and Muller, K., "How to Explain Individual Classification Decisions," Journal of Machine Learning Research, 2010.

[Fung, et al., 2005] Fung. G., Sandilya, S., and Rao, R. B., "Rule Extraction from Linear Support Vector Machines," Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, 2005.

[Hotelling, 1933] H. Hotelling, "Analysis of a complex of statistical variables into principal components," Journal of Educational Psychology, 24:417-441 and 498-520, 1933.

[Martin, 2002] Martin, Shawn, "Kernel Near Principal Components Analysis," Sandia National Laboratory Report SAND2001 -3769, July 2002.

[Jolliffe, et al., 2003] Jolliffe, I.T.; Trendafilov, N.T. and Uddin, M. A modified principal component technique based on the LASSO. Journal of Computational and Graphical Statistics, 12(3) pp. 531–547. 2003.

[Wikipedia, 2017] Wikipedia, "Gram-Schmidt process," https://en.wikipedia.org/wiki/Gram-Schmidt_process, accessed June 1, 2017.

Towards Explainable Text Classification by Jointly Learning Lexicon and Modifier Terms

Jérémie Clos and **Nirmalie Wiratunga** and **Stewart Massie**
Robert Gordon University, Garthdee Road, Aberdeen, United Kingdom
j.clos@rgu.ac.uk

Abstract

Automatically classifying text documents is an active research challenge in document-oriented information systems. It allows such systems to help users browse massive amounts of data with ease by categorizing it on some axis of interest, detect the unknown authors of unsigned work, and analyze corpora along predefined dimensions of interest such as sentiment, emotion or stance. However, current approaches are biased towards building complex black-box algorithms focused on producing high accuracy predictions at the cost of not being able to explain the rationale behind their decisions. Lexicon-based classifiers offer a white-box alternative to these approaches by using a trivially interpretable additive model at the cost of classification accuracy. The contribution of this paper is RELEXNET, a computational architecture that models lexicons as naive gated recurrent networks, allowing us to train them using standard optimization approaches. We evaluate our approach on two tasks: stance detection and sentiment classification, and show that our approach is competitive with standard black-box shallow classifiers.

1 Introduction

Text classification is a core task in natural language processing, with applications ranging from web search to author detection. For example, support vector machines [Hearst *et al.*, 1998], a common family of classification algorithms [Fernández-Delgado *et al.*, 2014] have helped improve document navigation tasks by categorizing web search results [Chen and Dumais, 2000], analyzed corpora to identify anonymous authors [Diederich *et al.*, 2003], and are used to identify spam e-mails [Drucker *et al.*, 1999] at large scale. However, these supervised classification algorithms provide predictions with no means of explanation. Understanding the reason behind a classification allows us to establish trust in further predictions, which has far-reaching consequences in algorithms deployed in production systems such as search engines and document categorization pipelines. Lexicons fill this need by offering a trivially interpretable additive model, where the probability of an instance belonging to a class is modeled as a weighted sum of the probabilities of each term of that instance belonging to that class. An analyst can then examine the terms of an instance and their weights to understand the reason behind a prediction. However, current techniques used to build those lexicons are lacking in many respects compared to standard supervised text classifiers. This paper attempts to conciliate lexicon-based classification and traditional classification models by defining a simple and effective training procedure that can generate lexicons with a classification accuracy that is competitive with standard classification algorithms. We illustrate the explanation step of a lexicon prediction through an example in figure 1.

We first formalize the concept of lexicons and explore the state of the art in the domain of lexicon-based classification. We then detail our contribution, formalizing lexicon-based classification as a gated computational graph and inducing optimal weights using a regularized objective function. We then detail our evaluation protocol on two classification tasks: stance detection and sentiment classification. We perform an evaluation against standard lexicon learning techniques and baselines found in the literature and report that our approach significantly outperforms standard text classification techniques. Finally, we analyze and discuss our results, before exploring the next steps of our work.

2 Related works

Despite its widespread use in real-world applications, text classification heavily relies on black-box models offering little if any explanation on their predictions [Ribeiro *et al.*, 2016]. Lexicon-based classifiers overcome this limitation by constraining the classification to a simple model: each term has a score for each class, those scores get weighted according to the frequency of that term in the instance and then added together, and finally the class with the highest total score for a given instance is chosen as the prediction. Such a classification model offers transparency: each prediction can be explained trivially by analyzing the terms that were present in the text, and any domain expert could revise the model manually with a simple text editing software. This transparency however comes at the cost of some classification accuracy, due to the simplistic nature of its inference scheme.

Example 1. In a binary sentiment classification setting, for a given sentence "*I love horror books*", a lexicon $\mathcal{L}$ referred on the figure, the lexicon could find an aggregated score of $f(love) \times 1.0 + f(horror) \times 0.3 + f(books) \times 0.5 = 1.8$ for the *Positive* class, and $f(love) \times 0 + f(horror) \times 0.7 + f(books) \times 0.5 = 1.2$ for the *Negative* class, where f is a function measuring some notion of local term frequency. The decision function $\mathcal{D}$ would then return the class with the maximum value, i. e., *Positive*. A human reader can read the sentence and identify that the term "*love*" is responsible for tipping the classification towards the *Positive* class.

Example Lexicon		
Term	*Positive*	*Negative*
love	1.0	0.0
horror	0.3	0.7
books	0.5	0.5

Figure 1: Classification and explanation with a sentiment lexicon

2.1 Lexicon-based classification

Lexicons are linguistic tools for classification and feature extraction [Clos *et al.*, 2017; Bandhakavi *et al.*, 2016]. They take the form of a list of terms weighted by their strength of association with a given class. Some lexicons also contain additional contextual information in order to help their users build more complex models [Muhammad *et al.*, 2016] as well as a list of terms which modify the strength (intensifiers, diminishers) or the polarity (negators) of other terms within a predetermined window. We describe the core of a lexicon as follows:

Formal lexicon. A lexicon Lex is a tuple $Lex = \langle \mathcal{L}, \mathcal{A}, \mathcal{D} \rangle$

where: $\mathcal{L}: \quad T \times C \mapsto \mathbb{R}$

$\mathcal{A}: \quad \mathbb{R}^n \mapsto \mathbb{R}$

$\mathcal{D}: \quad \mathbb{R}^n \mapsto \mathbb{R}$

For a given dictionary of terms T and set of classes of interest C, $\mathcal{L}$ is a mapping function that assigns an unbounded value to each pair (t, c) where term $t \in T$ and class $c \in C$. The function $\mathcal{A}$ is an aggregation function that accumulates scores and returns one value, and $\mathcal{D}$ is a decision function that selects and returns a single one of these aggregated values. Concretely, the mapping determines an evidence score for each term using a look-up list (the lexicon), propagates it to the aggregation function which aggregates the evidence into one cumulative score per class. Finally, the decision function evaluates each score to select the one that is the most likely.

This leads us to define a core challenge in lexicon-based classification: the lexicon induction problem. The next section reviews techniques traditionally used to solve the lexicon induction problem.

Lexicon induction problem. The lexicon induction problem is the estimation, given aggregation function $\mathcal{A}$ and decision function $\mathcal{D}$, of the optimal function $\mathcal{L}$ so that the resulting lexicon $Lex = \langle \mathcal{L}, \mathcal{A}, \mathcal{D} \rangle$ minimizes its classification errors on unseen data.

The added complexity in this paper stems from the presence of a new class of terms: **modifiers**. Modifiers are either intensifiers, diminishers or negators, and while they have no class value of their own they alter the value of class-bearing terms in a specific context window. Modifiers can be represented as a single value that is multiplied to the class value of all terms within a certain context window. For instance, the adverb "very" would have a modifier value of 1.3, meaning that a term such as "bad" which would have a negative value of 0.9 in a sentiment analysis setting would end up having a modified value of 1.17 after consideration. We note that modifier terms should ideally be able to interact with each other (e. g., "not very" being different from "not" and "very" applied separately) but that is left for future work.

2.2 Lexicon induction techniques

Research in lexicon induction outlines multiple families of techniques that can be used to generate a computational lexicon. Those techniques are either built on an extensive lexical resource such as an ontology, or on an estimation of strength of association between each term and a class in a reference corpus. Research has shown that merging multiple lexicons produces a reliable feature extractor to augment an existing classifier [Wang and Cardie, 2014], but using those lexicons for direct classification was not explored as it would blur the link between features and prediction.

Traditional hand-crafted lexicons (THCL) Due to the computational cost of building a lexicon from text, early lexicons were hand-crafted by domain experts [Stone *et al.*, 1966] and while higher performance in automated classification tasks has been shown using modern techniques, there still exist handcrafted lexicons in use to this day such as the Linguistic Inquiry and Word Count lexicon [Pennebaker *et al.*, 2001]. The strengths of these approaches are that they generalize well and are highly interpretable due to their human (and not algorithmic) origin. Conversely their weakness are that they tend to be small due to the human labor involved in generating them, and less effective than other methods due to their focus on human interpretability. However they can provide a commonsense knowledge back-up in hybrid lexicons [Muhammad *et al.*, 2014] with some degree of success.

Ontology-based lexicons (OBL) OBL learning techniques use a few human-provided seed words for which the class is known, and leverage some external relationship (typically synonymy, antonymy and hypernymy) in a semantic graph such as WordNet [Miller, 1995] to propagate class values along that graph [Esuli and Sebastiani, 2006]. Because this family of techniques is extremely foreign to the one we are proposing, we do not evaluate against it and only refer to it for the sake of exhaustiveness.

Corpus statistic-based lexicons (CSBL) CSBL learning techniques use a labeled corpus of interest in order to learn a domain-specific lexicon. The two main statistics used for

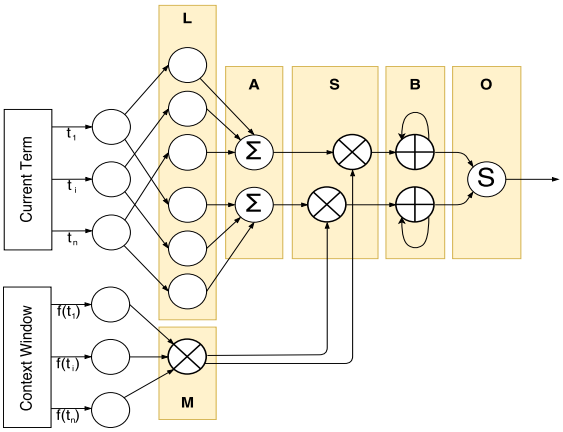


Figure 2: The RELEXNET topology

this purpose are the conditional probability (equation 1) of observing a term given a class, and the pointwise mutual information (PMI, equation 2) between the observation of a term and the observation of a class. These approaches are flawed in that they can overemphasize spurious correlations between terms and classes. For example, if a non-class specific term such as "Monday" accidentally co-occurs too often within one class, it will be misconstrued as being indicative of that class, and the lexicon will overfit. Bandhakavi et al. [Bandhakavi *et al.*, 2014] describe a method for building conditional probability-based lexicons and Turney [Turney, 2002] an approach using PMI and an external search engine to compute lexicon scores. Other works [Clos *et al.*, 2016] have shown some improvement using the normalized PMI measure (NPMI, equation 3) on a stance classification task.

$$P(t;c) = \frac{p(t|c)}{\sum_{i=0}^{|C|} p(t|c_i)} \tag{1}$$

$$PMI(t;c) = \frac{\log(p(t;c))}{p(t)p(c)} \tag{2}$$

$$NPMI(t;c) = \frac{\frac{\log(p(t;c))}{p(t)p(c)}}{-\log\left[p(t;c)\right]} \tag{3}$$

3 The RELEXNET architecture

In this section we present the RELEXNET architecture, presented in figure 2. The goal of RELEXNET is to jointly learn lexicon and modifier terms. We expect modifiers to be adverbs which alter the class valence of neighboring terms to different degrees, so in order to accelerate the learning we limited the sets of candidate terms to adverbs (e. g., very, etc.) and conjunctions (e. g., and, or, etc.). Each term is processed in their own timestep (one at a time), and the current score (aggregating the current timestep with the previous ones) is passed to the next timestep until the final term is processed and a prediction can be produced.

Figure 2 presents an architecture for binary classification. At the top of the diagram is the standard vocabulary layer (first layer), where each term of the instance is fed one at a time, activating the corresponding lexicon units (second layer, marked **L**) before being summed up in the aggregation unit (third layer, marked **A**), which is then modified by modifier terms nearby (fourth layer, marked **S**) and passed on to the next timestep (next term of the instance) via a summation gate (fifth layer, marked **B**). At the end of the sequence we examine the output of the Softmax function (sixth layer, marked **O**) which determines the probability distribution over classes. At the bottom of the diagram we process a context window of arbitrary size. For a context window of size c and for timestep i, all terms from t_{i-c} to t_{i+c} are fed as a bag of words as the context window ($f(t_i)$ denoting the frequency of term i in the context window). A weighted sum of those terms is then multiplied together before being fed in the top part of the network, to modify the score of the current term.

The forward pass of the graph can be represented in the following set of equations:

$$L_{i,j} = w_{i,j} \times f_1(t_i) \tag{4}$$

$$M = \prod_{i=1}^{|t|} (m_i \times f_2(t_i)) \tag{5}$$

$$A_j = \sum_{i=1}^{|t|} L_{i,j} \tag{6}$$

$$S_{i,j} = A_i \times M \tag{7}$$

$$B_{j,t} = B_{j,t-1} + \sum_{i=1}^{|c|} S_{i,j} \tag{8}$$

$$O_t = \bigoplus_{j=1}^{|c|} \frac{e^{-B_{j,t}}}{\sum_{m=1}^{|c|} e^{-B_{m,t}}} \tag{9}$$

In this equation $\bigoplus$ is a function that takes for input a sequence of real numbers and outputs a vector that contains them. The network passes on the data, and at each timestep feeds a one-hot encoding vector of the current term. We detail each part as follows:

- $L_{i,j}$ receives the one-hot encoding vector of the current term i for class j. In figure 2 we can observe that each term is mapped to N lexicon units, where N is the number of classes. It is formalized in equation 4.
- A_j sums the amount of evidence for class j. It is formalized in equation 6.
- M combines all modifiers present within the arbitrary word window into one modification score. It is formalized in equation 5.
- $S_{i,j}$ applies the modification score calculated in equation 5 to the score calculated in equation 6 for term i and class j. It is formalized in equation 7.
- $B_{j,t}$ communicates the output of $S_{i,j}$ at timestep t to the next timestep $t+1$. It is formalized in equation 8.

- O_t receives the output of $B_{j,t}$ (for all classes j) and outputs a probability distribution over the classes at timestep t. It is formalized in equation 9.

4 Experimental setup

4.1 Setting up RELEXNET

The training and testing of our approach was done using a 10-fold cross-validation while fixing the random seed of our algorithm to ensure that the results were consistent. The context size was arbitrarily fixed to the average phrase length in the corpora and not optimized further

Since 3 of the 4 datasets are balanced, we selected classification accuracy as our performance measure.

4.2 Datasets

We performed our evaluation on two tasks, containing a total of 4 datasets for stance classification and sentiment analysis. We describe the tasks and datasets associated in the rest of this section:

- **Stance detection** is the study of local stance of a document with respect to a topic or another stance. For example, if the topic of discussion is "death penalty" and a document d_1 is for the death penalty, then a document d_2 that is against the death penalty is said to be in disagreement with document d_1, while a document d_3 that is also for the death penalty is said to be in agreement with document d_1. In this work, we consider a reduced version of stance classification where the topic is not observed, and the classifiers are not provided with context.
  - **The IAC dataset** is a subset of the Internet Argument Corpus [Walker *et al.*, 2012] containing forum comments crawled from 4FORUMS on different topics: e. g., politics, ... and labeled on a scale from -5 to 5. A subset of comments that ensured disjoint class membership (with an average score far from 0) and containing more than 3 words was binned into 2 classes (agreement and disagreement) and used for our experiments.
  - **The CD dataset** is a dataset collected from the CREATEDEBATE forum dedicated to social argumentation on political and religious topics and labeled using 2 classes (agreement and disagreement). The dataset was used as is with no processing.
- **Sentiment classification** is the study of the sentiment (positive or negative) contained within a piece of text. While many datasets propose finer-grained sentiment classes (including neutral class or numerical sentiment score) we chose to use a binary classification task as our goal.
  - **The AYI dataset** was collected from Amazon, Yelp and IMDB and was built from individual sentences from product, location and movie reviews (respectively).
  - **The AMZ dataset** was collected from Amazon user reviews.

Stance classification			
Baseline lexicons	CPBLEX	0.524	0.441
	PMILEX	0.557	0.529
Baseline classifiers	NAIVEBAYES	0.536	0.474
	SVM	0.589	0.594
	DECISIONTREE	0.582	0.573
Approach	**RELEXNET**	**0.655***	**0.677***
Sentiment classification			
Baseline lexicons	CPBLEX	0.528	0.519
	PMILEX	0.571	0.554
Baseline classifiers	NAIVEBAYES	0.665	0.637
	SVM	0.689	0.671
	DECISIONTREE	0.742	0.707
Approach	**RELEXNET**	**0.751**	**0.719***

Figure 3: Experimental results

4.3 Baselines

We used two families of baselines as comparison points with our approach:

Lexicons: Two lexicons used as a baseline are the CPBLEX and the PMILEX, which are standard methods for building lexicons for other purposes, such as sentiment lexicons [Jurafsky and Martin, 2016]. Section 2.2 on corpus-based lexicons details their implementation ;

Standard classifiers: SVM (with a RBF kernel), which has been shown to perform well in stance detection tasks by Yin et al. [Yin *et al.*, 2012] and is a regular top performer in general classification tasks [Fernández-Delgado *et al.*, 2014], and NAIVEBAYES and DECISIONTREE which are two popular baselines for text classification. Parameters for the classifiers were taken from the default recommendations of the SCIKIT-LEARN [Pedregosa *et al.*, 2011] library.

5 Results and discussion

Table 3 shows that RELEXNET significantly outperforms the baselines (on a two-tailed paired T-test, with $p < 0.05$), while producing a human-readable lexicon that can be used to explain predictions. The hyperparameters were empirically determined on a hold-out set, leading to the choice of a lexicon size of 400 words (selected by corpus frequency), a regularization coefficient of 0.5 and a momentum velocity of 0.7.

6 Conclusion

In this work we showed the viability of using regularized backpropagation to efficiently learn effective lexicon weights, producing a lexicon that is competitive with standard classifiers and outperforms baseline techniques such as SVM. Our future works will focus on improving the performance of RELEXNET by taking into account modifier phrases, which are built from multiple modifier terms (e. g., "not very") and have a modifier valence of their own.

References

[Bandhakavi *et al.*, 2014] Anil Bandhakavi, Nirmalie Wiratunga, P Deepak, and Stewart Massie. Generating a word-emotion lexicon from #emotional tweets. In *Proc of the 3rd Joint Conf. on Lexical and Comp. Sem.*, 2014.

[Bandhakavi *et al.*, 2016] Anil Bandhakavi, Nirmalie Wiratunga, P Deepak, and Stewart Massie. Lexicon based feature extraction for emotion text classification. *Pattern Recognition Letters*, 2016.

[Chen and Dumais, 2000] Hao Chen and Susan Dumais. Bringing order to the web: Automatically categorizing search results. pages 145–152, 2000.

[Clos *et al.*, 2016] Jérémie Clos, Nirmalie Wiratunga, Stewart Massie, and Guillaume Cabanac. Shallow techniques for argument mining. In *ECA'15: Proceedings of the ECA*, volume 63, page 2, 2016.

[Clos *et al.*, 2017] Jérémie Clos, Anil Bandhakavi, Nirmalie Wiratunga, and Guillaume Cabanac. Predicting emotional reaction in social networks. In *European Conference on Information Retrieval*, pages 527–533. Springer, 2017.

[Diederich *et al.*, 2003] Joachim Diederich, Jorg Kindermann, Edda Leopold, and Gerhard Paass. Authorship attribution with support vector machines. *Applied intelligence*, 19(1):109–123, 2003.

[Drucker *et al.*, 1999] Harris Drucker, Donghui Wu, and Vladimir N Vapnik. Support vector machines for spam categorization. *IEEE Transactions on Neural networks*, 10(5):1048–1054, 1999.

[Esuli and Sebastiani, 2006] Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, volume 6, pages 417–422. Citeseer, 2006.

[Fernández-Delgado *et al.*, 2014] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems. *J. Mach. Learn. Res*, 15(1):3133–3181, 2014.

[Hearst *et al.*, 1998] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998.

[Jurafsky and Martin, 2016] Dan Jurafsky and James H Martin. *Lexicons for Sentiment Extraction*, chapter 18. Pearson, 2016.

[Miller, 1995] George A Miller. Wordnet: a lexical db for english. *Comm. of the ACM*, 38(11):39–41, 1995.

[Muhammad *et al.*, 2014] Aminu Muhammad, Nirmalie Wiratunga, and Robert Lothian. A hybrid sentiment lexicon for social media mining. In *Tools with AI (ICTAI), IEEE 26th International Conf. on*, pages 461–468, 2014.

[Muhammad *et al.*, 2016] Aminu Muhammad, Nirmalie Wiratunga, and Robert Lothian. Contextual sentiment analysis for social media genres. *Knowledge-Based Systems*, 108:92–101, 2016.

[Pedregosa *et al.*, 2011] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[Pennebaker *et al.*, 2001] James W Pennebaker, Martha E Francis, and Roger J Booth. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71:2001, 2001.

[Ribeiro *et al.*, 2016] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

[Stone *et al.*, 1966] Philip J Stone, Dexter C Dunphy, and Marshall S Smith. *The general inquirer: A computer approach to content analysis.* MIT press, 1966.

[Turney, 2002] Peter D Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proc. of the 40th annual meeting on ACL*. ACL, 2002.

[Walker *et al.*, 2012] Marilyn A Walker, Jean E Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. A corpus for research on deliberation and debate. In *LREC*, pages 812–817, 2012.

[Wang and Cardie, 2014] Lu Wang and Claire Cardie. Improving agreement and disagreement identification in online discussions with a socially-tuned sentiment lexicon. *ACL 2014*, page 97, 2014.

[Yin *et al.*, 2012] Jie Yin, Paul Thomas, Nalin Narang, and Cecile Paris. Unifying local and global agreement and disagreement classification in online debates. In *Proc. of the 3rd Workshop in Comp. Approaches to Subjectivity and Sentiment Analysis*, pages 61–69. ACL, 2012.

Explainable Planning

Maria Fox, Derek Long, Daniele Magazzeni
King's College London
firstname.lastname@kcl.ac.uk

Abstract

As AI is increasingly being adopted into application solutions, the challenge of supporting interaction with humans is becoming more apparent. Partly this is to support integrated working styles, in which humans and intelligent systems cooperate in problem-solving, but also it is a necessary step in the process of building trust as humans migrate greater responsibility to such systems. The challenge is to find effective ways to communicate the foundations of AI-driven behaviour, when the algorithms that drive it are far from transparent to humans. In this paper we consider the opportunities that arise in AI planning, exploiting the model-based representations that form a familiar and common basis for communication with users, while acknowledging the gap between planning algorithms and human problem-solving.

1 Introduction

DARPA recently launched the *Explainable AI (XAI) program*[1] that aims to create a suite of AI systems able to explain their own behaviour. This program is mainly concerned with machine/deep learning techniques, as they are currently treated almost as a black box. For example, it is not possible to fully understand why alphaGo selected a specific move at each turn, or on what basis a neural network recognises an image as an "image of a cat".

The need for explainable AI is motivated mainly by three reasons:

- the need for trust;
- the need for interaction;
- the need for transparency.

If doctors want to use a neural network to make a diagnosis, they need to be confident that there is a clear rationale for the NN to diagnose a cancer, in order to build *trust*. As autonomy gathers traction, in many scenarios, instead of full autonomy, Human-Autonomy Teaming (HAT) is required, where humans interact with the AI systems, and for this humans need to understand why the AI system is suggesting something that the human would not do: this requires *interaction*. There are growing legal implications in the use of AI, and in the cases where the AI system makes the wrong decision, or simply disagrees with the human, it is important to understand why a wrong or different decision was made: this is *transparency*.

Explainable AI is harder to achieve than the good decision-making that underlies it. The need to explain decisions forces them to be made in ways that can be subsequently justified in human terms. Entirely trustworthy and theoretically well-understood algorithms can still yield decisions that are hard to explain. For example, linear programming is a well-established tool, but explaining the results it generates without simply 'appealing to authority' remains hard. Part of the difficulty lies in understanding what an explanation should actually contain.

On one hand it is evident that there has been amazing progress in machine/deep learning research and there is a huge proliferation of ML and DL learning. On the other hand, Deep Neural Networks are still far away from being explainable.

In contrast, AI Planning is potentially well placed to be able to address the challenges that motived the DARPA project on AI: planners can eventually be trusted; planners can allow an easy interaction with humans; planners are transparent (at least, the process by which the decisions are made are understood by their programmers).

This paper presents Explainable Planning (XAIP), describing some initial results, and proposing a roadmap for making XAIP more effective and efficient.

Of course the challenge of Explainable AI and the need of making machine/deep learning explicable remain of critical importance. At the same time, we think that XAIP is an important contribution in this direction, as Planning is an important area of AI with applications in domains where learning is not an option.

The paper is structured as follows. We give an overview of related work in the next section. In Section 3 we list some important questions that XAIP should address and in Section 4 we discuss the features of planning that facilitate explanations. In Section 5 we present initial results and suggest future directions. In Section 6 we show two illustrative examples. Section 7 concludes the paper.

[1] http://www.darpa.mil/attachments/DARPA-BAA-16-53.pdf

2 Related Work

For a survey of recent works in the broader area of Explainable AI, we refer to the IJCAI-17 XAI workshop website[2]. Here we briefly highlight some recent works that are related and can contribute to Explainable Planning. *Plan Explanation* is an area of Planning where the main goal is to help humans to understand the plans produced by the planners (e.g., [Sohrabi *et al.*, 2011]. This involves the translation of the planner outputs (e.g., PDDL plans) in forms that humans can easily understand; the design of interfaces that help this understanding (e.g., spoken language dialog systems [Bidot *et al.*, 2010]); and the description of causal and temporal relations for plan steps (e.g., [Seegebarth *et al.*, 2012]). Note that making sense of a plan (plan explanation) is different from explaining why a planner made decisions (XAIP).

Plan Explicability [Zhang *et al.*, 2017] focuses on human's interpretation of plans. Learning is used to create a model of the interpretations, which is then used to measure the explicability and predictability of plans.

Veloso and her team look at the problem of generating narrations for autonomous mobile robot navigations. They contribute with *verbalization*, where the robot experience is described via natural language [Rosenthal *et al.*, 2016].

In *Model reconciliation* [Chakraborti *et al.*, 2017], the focus is on the agent and the human having two different models, hence the explanations must identify and reconcile the relevant differences between the models.

David Smith in his AAAI invited talk presented *Planning as an Iterative Process* [Smith, 2012], and he discussed the broad problem of users interacting with the planning process, which also includes questions about choices made by the planner. Pat Langley et al. more recently used *Explainable Agency* to refer to the ability of autonomous agents to explain their decisions, and in [Langley *et al.*, 2017] they discuss some functions that agents should exhibit.

In this paper, we go beyond a discussion of the questions that need to be answered, and by focusing on AI planning we provide initial results on how to address some of the questions and we point to concrete works in the community to address the others.

3 Things to Be Explained

As mentioned in the introduction, one of the challenges of XAI is to understand what constitutes an explanation. In general, rewriting the steps of the decision-making algorithm in natural language is not what is required. For example, despite it being the case that many planners select actions in their plan-construction process in order to minimize a heuristic distance to goal, based on a relaxed plan, even these terms are inappropriate vocabulary to explain the process to a human. In any case, a real danger in XAI is to reduce an explanation to the statement of the obvious. It is clearly not the answer to the question 'why did you do that?' to say 'because it got me closer to the goal'. A request for an explanation is really an attempt to uncover a piece of knowledge that the questioner believes must be available to the system and that the questioner does not have.

In this section we list some of the questions that characterise what it means for the behaviour of a planner to be *explainable*, and we discuss what constitutes a response to these questions.

- Q1: Why did you do that?
  This is one of the most fundamental questions that can be asked about a plan. It is also an excellent example of how complex the intention behind the question can be. In a sufficiently long and complex plan, it is plausible that the questioner is unable to immediately see which later action in the plan is supported by the target action. Thus, the answer could be as simple as 'action A is in the plan to allow this application of action B'. However, for shorter and more easily assimilated plans, the question is far more likely to be an implicit question: 'why did you do action A? *I would have done action B*'
- Q2: And why didn't you do *something else* (that I would have done)?
  This question is similar to the intention in Q1, but makes the alternative action explicit. An answer to this question would normally be a demonstration of a flaw in a plan that uses the proposed alternative action compared with the plan actually produced. It would usually be acceptable to demonstrate that the plan actually produced was no worse than a plan using the proposed alternative action. In order to respond with either a flaw or else a demonstration of neutral cost, it is necessary to infer by what metric the alternatives are to be compared (one plan might be longer but cheaper than a second — depending on the relative values of time and money, either plan might be considered better).
- Q3: Why is what you propose to do more efficient/safe/cheap than something else (that I would have done)?
  This question refines Q2 by being explicit about the metric being used to evaluate the plans. If the metric is different to the one used in constructing the original plan, then the answer might be to point out the different basis for evaluation of plans. This is a valid explanation provided the original plan is better under the original metric than the rival proposal.
- Q4: Why can't you do that?
  Here we consider the form of this question arising when a planner fails to find a plan for a problem. Planners are typically not very effective at proving unsolvability of planning problems, but model-checking techniques can be applied to planning domain models in order to attempt to prove the non-existence of plans. Unfortunately, converting the exhaustive search of a space (albeit supplemented with careful relaxation-based approaches that bundle parts of the search space) into a transparent and succinct argument is extremely challenging.
- Q5: Why do I need to replan at this point?
  During execution, plan failure will be caused by a deviation between the expected behaviour and the observed

[2] http://home.earthlink.net/∼dwaha/research/meetings/ijcai17-xai/

behaviour of the world. This question can be directed at discovering what has diverged from expectation, or it might be that the deviation is understood, but the significance of the deviation is not. Thus, this question seeks to know what is it that the executing plan was depending on being true that has been observed not to be.

- Q6: Why do I not need to replan at this point?

  There are two reasons possible for this question. One is that the observer has seen a divergence in expected behaviour and does not understand why it should not cause plan failure and the other is that the observer has seen non-diverging behaviour that is not what was expected by the observer. In other words, the divergence could be observable for the executive (and then the explanation is to show why it does not cause plan failure) or else there might be no divergence observable by the executive, in which case the explanation (assuming there is a valid explanation) is to show why the observed behaviour was expected at this point.

Of course there are other questions related to Explainable Planning, including those arising when we consider probabilistic planning or planning under uncertainty, as well as anytime planning (e.g., *will I get a significantly better plan if I give the planner 10 more minutes?*).

4 Unique Features of Planning

AI Planning exploits a collection of techniques that have the potential to make it easier to understand the decision process (even if it is very complex and requires sophisticated algorithms and heuristics).

First of all, AI Planning is based on *models*. Models are used to create plans, and can also be used during plan execution as well as after a plan has been executed. One of the driving principles for a large part of the work carried out in planning is that planners should encapsulate the machinery of planning independently of the domain of application. This means that models capture the dynamics of domains. However, a second guiding principle has been that domain models should not attempt to direct the decision process in the planner (this is not a universally adopted principle, but it has been a consequence of the international planning competition series that domain models have been developed to be 'pure' descriptions of what can be done, not how to do it). McDermott articulated this as the maxim that domain models should contain 'axioms, not advice'. The implication of this for modellers is that they do not need to understand how a planner works, but only the behaviour of their domain. This leads to more intuitive and accessible models for users and facilitates their use in explanation.

Second, plan execution provides its *execution trace*, as a set of pairs (observation, action) which can be used to explore the reasons behind the choices of actions and allows explanations to focus on aspects of state or of action choice, depending on the question.

Third, some progress has been made in explaining plans, in order to help humans to understand the meaning of plans, with a long history based on mixed-initiative planning.

Most planners are based on transparent algorithms, where the planner choice at each decision point is deterministic, repeatable and based on a specific choice mechanism. The fact that the reason for the choice of an action is transparent to the programmer, at least, makes it plausible that we can construct an articulation of parts of that reason in a form a human user might appreciate.

5 Providing Explanations

In this section we address the questions presented in Section 3. For each question, we highlight the main challenges, present some preliminary results and propose a roadmap for achieving the goal of providing reasonable answers and explanations.

5.1 Explaining why the planner chose an action

This explanation introduces two main challenges, that are also inherited by all the following explanations. First, an explanation needs to show *causality* among actions. While this is obvious in many cases (e.g., I get the key first so that I can open the door later), there are examples where action A early in the plan is needed to support action B much later in the plan. This causal relationship might not be evident.

As a practical example, in the electricity domain [Piacentini *et al.*, 2015], the planner reverses current through a transformer *early* in the afternoon in order to support the achievement of supply within thermal constraints in the peak demand period *later* in the evening. This was performed early because there was more flexibility to reconfigure the flows when the load was lighter and by the time the supply was required, demand was so high that the network had no longer got flexibility to change the configuration.

The second issue is that the plan must be understandable to humans, who are not supposed to be planning researchers or experts. Hence, planning formalisms such as PDDL need to be presented to humans in a more natural-language fashion, and the works on plan explanation go in this direction (e.g., [Seegebarth *et al.*, 2012]).

5.2 Explaining why the planner did not choose an action

When a planner decision is confronted with an alternative suggested by the human, an explanation should be a demonstration that the alternative action would prevent from finding a valid plan or would lead to a plan that is no better than the one found by the planner. However, just showing the different heuristic value is not a valid explanation, unless it is translated into a value which is more informative for the user than for example the RPG heuristic value.

What is needed, instead, is an algorithm that executes the plan up to the point where the human suggests the alternative, then injects the human decision, and finally replans from the state obtained after applying the action suggested by the human.

Here there is a first issue, though, as one possible behaviour of the planner could be to just do an *undo* of the human action and produce the original plan (see Figure 1 part (a)).

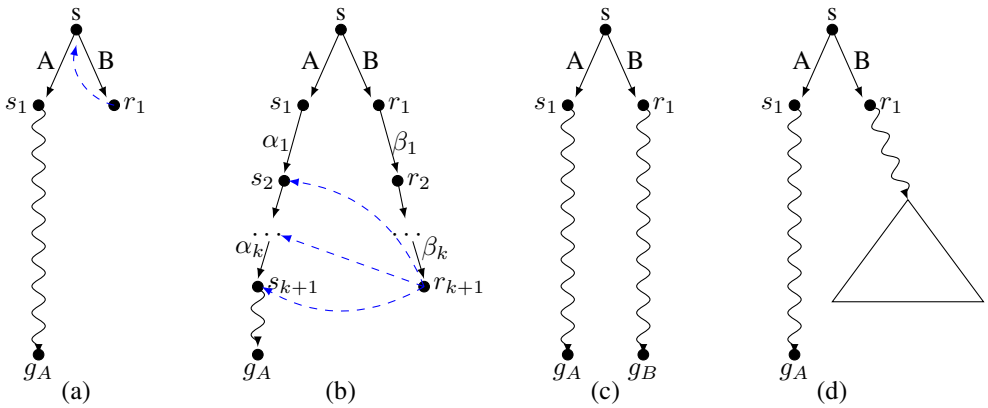


Figure 1: Possible plan behaviours after human-decision injection.

One possible fix is to forbid the planner to revisit the state where the human decision was injected. This, however, does not prevent the planner to get back to the original plan after k steps, as shown in Figure 1 part (b). In this case, the explanation can provide the different costs of the two alternatives, that is $C_A = c(A) + c(\alpha_1) + \ldots + c(\alpha_k)$ and $C_B = c(B) + c(\beta_1) + \ldots + c(\beta_k)$. More in general, the human may want to inject a longer plan, and in this sense the explanation must enable the interaction with the human, by allowing the human to inject more than one action, and provide an explanation after each injection.

The two remaining possible outcomes after the human-decision injections are either that the planner finds a plan for a different goal, or it fails to find a plan, as shown in Figure 1 parts (c) and (d), respectively.

5.3 Explaining why the planner decisions are *better*

The third question we listed was: *why is the planner decision more efficient/safe/cheap than what I would do?* The focus here is that different *metrics* can be used to evaluate a plan. The more interesting case is when one wants to evaluate a plan using a metric which is different from the one used when searching for the plan. This is of practical importance, given that when dealing with complex domains (temporal/numeric domains) there are a number of planners able to minimise the makespan, while almost none are able to optimise other metrics (e.g., total cost). Hence, for example one would like to understand whether a plan found using POPF or FastDownward is actually more efficient (in terms of total cost), than an alternative plan suggested by the human.

This explanation is a refinement of previous explanations, and our proposed solution is to integrate the previous explanation approach with the validator VAL [Fox *et al.*, 2005], where different metrics can be specified to evaluate plans. After each injection of human decision, the validator is used to execute the alternative plan against the new metric (total cost for example).

5.4 Explaining why things can *not* be done

There are two reasons why one action can not be applied in a given state. Either because the current state does not satisfy the action precondition; or because the application of that action would prevent achieving the goal from the resulting state. To provide an explanation for the first case is straightforward, and the validator VAL already provides this. Explanations for the second case are more challenging, and anyway would also be used for providing more general justifications for why the goal cannot be achieved at all (i.e., the planning problem is unsolvable). To this end, we suggest that a promising direction is given by the work being done in proving plan non-existence (see, among others, [Bäckström *et al.*, 2013], [Steinmetz and Hoffmann, 2016], [Hoffmann *et al.*, 2014]).

Model-checking algorithms and tools can prove very suitable [Clarke *et al.*, 2001]. Indeed, in the planning-as-model-checking paradigm [Giunchiglia and Traverso, 1999], a planning problem is cast as a verification problem, where the safety property to be verified is set as the negation of the goal of the planning problem. In this way, if the model checker returns an error trace, that would correspond to the plan. On the contrary, if the model checker states that the property (not goal) is satisfied for all the reachable states, this is a proof that there is no plan. Recently, this paradigm has been applied to complex planning problems with temporal and numeric features [Bogomolov *et al.*, 2014; 2015].

Another very relevant topic is the research around Simple Temporal Networks [Dechter *et al.*, 1991]. STN can be used, for example, to show why an action taking longer than expected can invalidate the plan. There is a large amount of research around STN, STNU [Vidal and Ghallab, 1996], and controllability of STN. For a survey on this field, we refer to [Micheli, 2016].

While proving plan-non-existence (or STN inconsistency) is not yet explaining why the problem is unsolvable, we highlight here that a roadmap for this question should build on the cited works.

5.5 Explaining why one needs to replan

This kind of explanation is needed at plan-execution time. In many real-world scenarios, it is not obvious that the plan being executed will fail for some changes in the environment and/or for mismatching about the model of the environment and the real environment. In most of the cases, plan failure is discovered only when it is too late for replanning in an efficient way.

Explanations for when replanning is needed must take into account the whole plan being executed and check its validity when monitoring the environment. To this end, one possible approach is to use the *filter violation* techniques, as described in [Cashmore *et al.*, 2015] and implemented in ROSPlan.

ROSPlan uses a Knowledge Base to store information about the environment and the plan being executed. The Knowledge Base is updated as soon as new information becomes available. Each change to the Knowledge Base is checked against a filter that is created as follows. Once the plan is generated, the filter is constructed by taking the intersection of static facts in the problem instance with the union of all preconditions of actions in the plan. In addition, each object instance involved in these facts is added to the filter. For example, suppose we have a PDDL domain with the object type `waypoint`, the static fact `(connected ?from ?to - waypoint)` and the action `navigate ( ?v vehicle ?from ?to waypoint)` whose preconditions include `(connected ?from ?to)`. For each `navigate` action scheduled in the plan, the waypoint instances bound to `from` and `to` and the ground fact `(connected ?from ?to)` are included in the filter. If these objects are removed, or altered in the Knowledge Base, a notification will be sent to the Planning System. An example of this type of explanation is provided in Section 6.

Another promising approach in this direction is Discover-History, described in [Molineaux *et al.*, 2012].

5.6 Explaining why one does *not* have to replan

This explanation is of practical importance, as it is concerned with the situation where the environment being observed (including the plan execution) is different from what was anticipated. In this very common case, it is important to avoid the naive approach of continuous replanning. Rather, it would be ideal to have a way to understand why the plan is still valid, despite the differences between what expected and what being observed. The most common scenario is when actions are taking longer than expected (again, one can think of an underwater mission, where an unexpected current is slowing down the AUV, hence navigate actions are taking longer than anticipated). We highlight here that a promising research direction is represented by the work on dynamic controllability of STN (e.g., [Vidal and Fargier, 1999], [Morris *et al.*, 2001]).

6 Illustrative Examples

In this section we provide two examples of how the approach described in the previous section can be used to explain a plan and to explain why replanning is needed.

6.1 The Rover Domain

We consider the `rover time` domain from IPC-4 and problem 3. Here is the plan found by POP-F [Coles *et al.*, 2010].

```
0.000: (navigate r1 wp3 wp0)  [5.0]
0.000: (navigate r0 wp1 wp0)  [5.0]
5.001: (calibrate r1 camera1 obj0 wp0)  [5.0]
5.001: (sample_rock r0 r0store wp0)  [8.0]
10.002: (take_image r1 wp0 obj0 camera1 col)  [7.0]
13.001: (navigate r0 wp0 wp1)  [5.0]
17.002: (navigate r1 wp0 wp3)  [5.0]
18.001: (comm_rock_data r0 general1 wp0 wp1 wp0)  [10.0]
22.003: (navigate r1 wp3 wp2)  [5.0]
27.003: (sample_soil r1 r1store wp2)  [10.0]
28.002: (comm_image_data r1 general obj0 col wp2 wp0) [15.0]
43.003: (comm_soil_data r1 general wp2 wp2 wp0)  [10.0]

[Duration = 53.003]
```

An instance of Q1 could be: *"Why did you use Rover0 to take the rock sample at waypoint0?"*

A naive answer could be: *so that I can* `communicate_rock_data` *from Rover0 later in the plan (at 18.001)*. This is naive because the plan is so short that the user can easily see that this is performed and, presumably, will realise that the data can only be communicated by the rover that has it. A better way to interpret the question would be to consider alternative ways to achieve the goal this action supports: to communicate the rock data from Waypoint0. It turns out the only way to communicate the rock data is to first have the rock analysis from Waypoint0. And there are only two ways to do this, either to sample rock with Rover0 and or with Rover1.

Hence, an instance of Q2 could be: *Why didn't Rover1 take the rock sample at waypoint0?* In order to provide an answer to this question, we need to force the planner to second the human input. To this end, we remove the ground action instance for Rover0 from those available to the planner and ask it to replan, and here is the new plan:

```
0.000: (navigate r1 wp3 wp0)  [5.0]
5.001: (calibrate r1 camera1 obj0 wp0)  [5.0]
10.002: (take_image r1 wp0 obj0 camera1 col)  [7.0]
10.003: (sample_rock r1 r1store wp0)  [8.0]
18.003: (navigate r1 wp0 wp3)  [5.0]
18.004: (drop r1 r1store)  [1.0]
23.004: (navigate r1 wp3 wp2)  [5.0]
28.004: (comm_image_data r1 general obj0 col wp2 wp0) [15.0]
28.005: (sample_soil r1 r1store wp2)  [10.0]
43.005: (comm_soil_data r1 general wp2 wp2 wp0)  [10.0]
53.006: (comm_rock_data r1 general wp0 wp2 wp0)  [10.0]

[Duration = 63.006]
```

Clearly this is far worse quality than the first plan (the metric is specified as makespan for these plans). So the answer could be: *Because not using Rover0 for this action leads to a worse plan.* It could be argued that this is not a very satisfactory answer, although it is better than the naive answer above, because it does not seem to explain why Rover1 does everything.

One option is for the human to follow up with another question: *Why does Rover1 do everything?* In order to answer this question, we can require the plan to contain at least one action that has Rover0 as an argument. This could be encoded in the domain automatically, by adding a dummy effect to all actions using Rover0 and then adding this as a goal, but here we use a plan generated by remodelling the domain manually, and this is the new plan found by the planner:

```
0.000: (navigate r0 wp1 wp0)  [5.0]
0.000: (navigate r1 wp3 wp0)  [5.0]
5.001: (calibrate r1 camera1 obj0 wp0)  [5.0]
10.002: (take_image r1 wp0 obj0 camera1 col)  [7.0]
10.003: (sample_rock r1 r1store wp0)  [8.0]
18.003: (navigate r1 wp0 wp3)  [5.0]
18.004: (drop r1 r1store)  [1.0]
23.004: (navigate r1 wp3 wp2)  [5.0]
28.004: (comm_image_data r1 general obj0 col wp2 wp0)  [15.0]
28.005: (sample_soil r1 r1store wp2)  [10.0]
43.005: (comm_soil_data r1 general wp2 wp2 wp0)  [10.0]
53.006: (comm_rock_data r1 general wp0 wp2 wp0)  [10.0]
```

Hence, an explanation is that this plan, while not being any longer, contains more actions, so is even worse than the last plan (and in fact contains all the actions of the last plan, so is actually a simple extension of that plan).

However, this is also not entirely satisfactory, because it only shows us that the planner cannot find a useful way to incorporate Rover0 into the plan, but not why. If we restrict the actions that can be used to achieve the dummy condition (that Rover0 acted in the plan) to the set of actions that achieve goals, then the planner cannot find a plan. So, the answer to the question could be slightly improved to: *I cannot find a plan in which Rover0 does not sample the rock at Waypoint0, but achieves a goal in the plan*.

In fact, it turns out that the problem specification prevents Rover0 from reaching Waypoint2, so the soil data there cannot be collected by Rover0, while only Camera1 can be calibrated for Objective0, so only Rover1 that carries Camera1 can be used to take that image. Therefore, the only task that Rover0 can perform is the rock sample mission at Waypoint0.

6.2 The AUV Domain

We consider the AUV domain from the PANDORA project [Cashmore *et al.*, 2014; Palomeras *et al.*, 2016], where an AUV has to complete an inspection mission, by navigating (`do_hover`) between waypoints and making observation of a set of inspection points. Here is a fragment of a plan for this scenario:

```
0.000: (observe auv wp1 ip3)  [10.000]
10.001: (correct_position auv wp1)  [10.000]
20.002: (do_hover auv wp1 wp2)  [71.696]
91.699: (observe auv wp2 ip4)  [10.000]
101.700: (correct_position auv wp2)  [10.000]
111.701: (do_hover auv wp2 wp23)  [16.710]
128.412: (observe auv wp23 ip5)  [10.000]
138.413: (correct_position auv wp23)  [10.000]
148.414: (observe auv wp23 ip1)  [10.000]
158.415: (correct_position auv wp23)  [10.000]
168.416: (do_hover auv wp23 wp22)  [16.710]
185.127: (do_hover auv wp22 wp26)  [30.201]
215.329: (observe auv wp26 ip7)  [10.000]
225.330: (correct_position auv wp26)  [10.000]
235.331: (do_hover auv wp26 wp21)  [23.177]
258.509: (observe auv wp21 ip2)  [10.000]
268.510: (correct_position auv wp21)  [10.000]
278.511: (do_hover auv wp21 wp27)  [21.255]
299.767: (observe auv wp27 ip8)  [10.000]
309.768: (correct_position auv wp27)  [10.000]
319.769: (observe auv wp27 ip6)  [10.000]
329.770: (correct_position auv wp27)  [10.000]
339.771: (do_hover auv wp27 wp17)  [23.597]
363.369: (do_hover auv wp17 wp25)  [21.413]
384.783: (do_hover auv wp25 wp32)  [16.710]
401.494: (do_hover auv wp32 wp36)  [21.451]
422.946: (observe auv wp36 ip9)  [10.000]
432.947: (correct_position auv wp36)  [10.000]
442.948: (observe auv wp36 ip15)  [10.000]
```

In the PANDORA project, the plans are generated and dispatched through the ROSPlan framework [Cashmore *et al.*, 2015] which is also used to monitor plan execution. ROSPlan implements the filter violation described in Section 5.5.

Figure 2 shows the execution of the plan above.

The blue lines represent the Probabilistic Road Map used to determine accessibility for the AUV, while the yellow lines represent the AUV trajectory. At time-point 215.329, while the AUV is observing the inspection point `ip7` it also observes that waypoints `wp32` and `wp36` are actually not connected (this is represented by the red line in Figure 5.5). Given that `(connected wp32 wp36)` was in the filter and after this observation the predicate is removed, a filter violation is triggered, which provides a justification for why replanning is needed, explaining that an action later in the plan (precisely at time-point 401.494) will no longer be executable, and also highlighting which condition has changed from its expected value to one that prevents the plan from being executable.

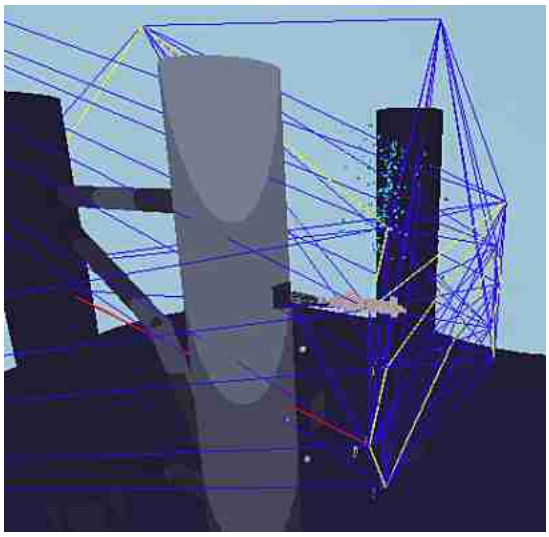


Figure 2: Plan Execution in the AUV Domain.

7 Conclusion

We have introduced Explainable Planning (XAIP), as a promising contribution to the Explainable AI (XAI) challenge. We characterised some of the questions that need to be explained, and provided initial results and a roadmap for achieving the objective of providing effective explanations. The next steps include a full formalisation of the XAIP problem, and a formulation of the user/planner interaction in terms of new constraints to add and alternatives to explore.

This work opens up a number of future directions in explanations for plans as well as plan execution. For example temporal planning introduces interesting planning choices about the order in which (sub)goals are achieved. Another interesting problem is to understand whether to expect improvement in giving the planner a given additional amount time for planning. For plan execution, especially with probabilistic planning and planning under uncertainty, one of the problems is to explain what has been observed at execution time that made the planner make a particular choice.

There is no clear way to define what constitutes a good explanation. As we argued in the paper, XAIP should not focus on explaining the obvious. However, defining a good metric for explanation is an important issue.

More in general, the literature in planning contains many works that could contribute to Explainable Planning. On the other hand, nowadays plans are much more complex than before, and are also used in many new critical domains. Existing works should be revisited and leveraged in order to make XAIP more effective and efficient.

References

[Bäckström *et al.*, 2013] Christer Bäckström, Peter Jonsson, and Simon Ståhlberg. Fast detection of unsolvable planning instances using local consistency. In *Proceedings of SOCS*, 2013.

[Bidot *et al.*, 2010] Julien Bidot, Susanne Biundo, Tobias Heinroth, Wolfgang Minker, Florian Nothdurft, and Bernd Schattenberg. Verbal plan explanations for hybrid planning. In *Proceedings MKWI*, 2010.

[Bogomolov *et al.*, 2014] Sergiy Bogomolov, Daniele Magazzeni, Andreas Podelski, and Martin Wehrle. Planning as model checking in hybrid domains. In *Proceedings of AAAI*, 2014.

[Bogomolov *et al.*, 2015] Sergiy Bogomolov, Daniele Magazzeni, Stefano Minopoli, and Martin Wehrle. PDDL+ planning with hybrid automata: Foundations of translating must behavior. In *Proceedings of ICAPS*, 2015.

[Cashmore *et al.*, 2014] Michael Cashmore, Maria Fox, Tom Larkworthy, Derek Long, and Daniele Magazzeni. AUV mission control via temporal planning. In *Proceedings of ICRA*, 2014.

[Cashmore *et al.*, 2015] Michael Cashmore, Maria Fox, Derek Long, Daniele Magazzeni, Bram Ridder, Arnau Carrera, Narcis Palomeras, Natalia Hurtos, and Marc Carreras. ROSPlan: Planning in the robot operating system. In *Proceedings of ICAPS*, 2015.

[Chakraborti *et al.*, 2017] T. Chakraborti, S. Sreedharan, Y. Zhang, and S. Kambhampati. Plan explanations as model reconciliation: Moving beyond explanation as soliloquy. In *Proceedings of IJCAI*, 2017.

[Clarke *et al.*, 2001] Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model checking*. MIT Press, 2001.

[Coles *et al.*, 2010] Amanda Jane Coles, Andrew Coles, Maria Fox, and Derek Long. Forward-chaining partial-order planning. In *Proceedings of ICAPS*, 2010.

[Dechter *et al.*, 1991] Rina Dechter, Itay Meiri, and Judea Pearl. Temporal constraint networks. *Artif. Intell.*, 49(1-3):61–95, 1991.

[Fox *et al.*, 2005] Maria Fox, Richard Howey, and Derek Long. Validating plans in the context of processes and exogenous events. In *Proceedings of IAAI*, 2005.

[Giunchiglia and Traverso, 1999] Fausto Giunchiglia and Paolo Traverso. Planning as model checking. In *Proceedings of ECP*, 1999.

[Hoffmann *et al.*, 2014] Jörg Hoffmann, Peter Kissmann, and Álvaro Torralba. Distance? who cares? tailoring merge-and-shrink heuristics to detect unsolvability. In *Proceedings of ECAI*, 2014.

[Langley *et al.*, 2017] Pat Langley, Ben Meadows, Mohan Sridharan, and Dongkyu Choi. Explainable agency for intelligent autonomous systems. In *Proceedings of AAAI*, 2017.

[Micheli, 2016] Andrea Micheli. *Planning and Scheduling in Temporally Uncertain Domains*. PhD thesis, University of Trento, Italy, 2016.

[Molineaux *et al.*, 2012] Matthew Molineaux, Ugur Kuter, and Matthew Klenk. Discoverhistory: understanding the past in planning and execution. In *Proceedings of AAMAS*, 2012.

[Morris *et al.*, 2001] Paul H. Morris, Nicola Muscettola, and Thierry Vidal. Dynamic control of plans with temporal uncertainty. In *Proceedings of IJCAI*, 2001.

[Palomeras *et al.*, 2016] Narcís Palomeras, Arnau Carrera, Natàlia Hurtós, George C. Karras, Charalampos P. Bechlioulis, Michael Cashmore, Daniele Magazzeni, Derek Long, Maria Fox, Kostas J. Kyriakopoulos, Petar Kormushev, Joaquim Salvi, and Marc Carreras. Toward persistent autonomous intervention in a subsea panel. *Autonomous Robots*, 40(7):1279–1306, 2016.

[Piacentini *et al.*, 2015] Chiara Piacentini, Varvara Alimisis, Maria Fox, and Derek Long. An extension of metric temporal planning with application to AC voltage control. *Artif. Intell.*, 229:210–245, 2015.

[Rosenthal *et al.*, 2016] Stephanie Rosenthal, Sai P. Selvaraj, and Manuela M. Veloso. Verbalization: Narration of autonomous robot experience. In *Proceedings of IJCAI*, 2016.

[Seegebarth *et al.*, 2012] Bastian Seegebarth, Felix Müller, Bernd Schattenberg, and Susanne Biundo. Making hybrid plans more clear to human users - A formal approach for generating sound explanations. In *Proceedings of ICAPS*, 2012.

[Smith, 2012] David Smith. Planning as an iterative process. In *Proceedings of AAAI*, pages 2180–2185, 2012.

[Sohrabi *et al.*, 2011] Shirin Sohrabi, Jorge A. Baier, and Sheila A. McIlraith. Preferred explanations: Theory and generation via planning. In *Proceedings of AAAI*, 2011.

[Steinmetz and Hoffmann, 2016] Marcel Steinmetz and Jörg Hoffmann. Towards clause-learning state space search: Learning to recognize dead-ends. In *Proceedings of AAAI*, 2016.

[Vidal and Fargier, 1999] Thierry Vidal and Hélène Fargier. Handling contingency in temporal constraint networks: from consistency to controllabilities. *J. Exp. Theor. Artif. Intell.*, 11(1):23–45, 1999.

[Vidal and Ghallab, 1996] Thierry Vidal and Malik Ghallab. Dealing with uncertain durations in temporal constraint networks dedicated to planning. In *Proceedings of ECAI*, 1996.

[Zhang *et al.*, 2017] Y. Zhang, S. Sreedharan, A. Kulkarni, T. Chakraborti, H. Zhuo, and S. Kambhampati. Plan explicability and predictability for robot task planning. In *Proceedings of ICRA*, 2017.

Towards Compact Interpretable Models: Shrinking of Learned Probabilistic Sentential Decision Diagrams

Yitao Liang
Computer Science Department
University of California, Los Angeles
yliang@cs.ucla.edu

Guy Van den Broeck
Computer Science Department
University of California, Los Angeles
guyvdb@cs.ucla.edu

Abstract

Probabilistic sentential decision diagrams (PSDDs) were recently introduced as a tractable and interpretable representation of discrete probability distributions. PSDDs are tractable because they support a wide range of queries efficiently. They are interpretable because each parameter in the PSDD represents a conditional probability, as in Bayesian networks. This paper summarizes ongoing research that aims to answer two questions that are important to employ PSDDs as an explainable AI model. First, as a tractable and interpretable model, can PSDDs compete with more general machine learning models for density estimation? We answer this question positively, reporting state-of-the-art results on standard benchmarks. Second, can we effectively reduce the number of parameters in a learned PSDD to simplify its interpretation, without harming the quality of the learned model? For this task, we present an algorithm that merges PSDD substructures that are similar in KL-divergence, which we show can be done efficiently on PSDDs.

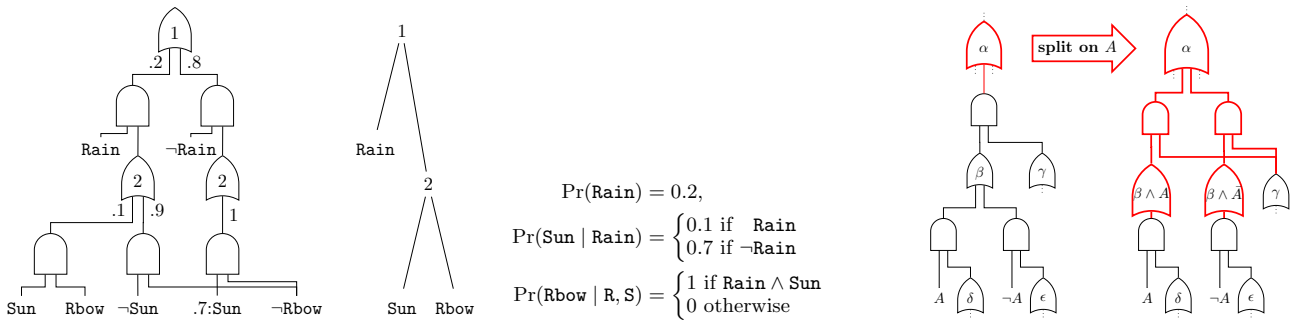
1 Introduction

Tractable learning aims to induce complex, yet tractable probability distributions from data (Domingos *et al.*, 2014; Mauro and Vergari, 2016). The learned tractable model serves as a certificate to the user that any query that arises can always be answered efficiently. Tractable learning initially targeted sparse graphical models (Meila and Jordan, 2000; Narasimhan and Bilmes, 2004; Chechetka and Guestrin, 2007). More recently, tractable circuit representations of probability distributions, such as *arithmetic circuits* (ACs) (Darwiche, 2003), have become the chosen target representation for these learners (Lowd and Domingos, 2008; Lowd and Rooshenas, 2013; Gens and Domingos, 2013; Dennis and Ventura, 2015; Bekker *et al.*, 2015), spurring innovation in arithmetic circuit dialects such as *sum-product networks* (SPNs) (Poon and Domingos, 2011; Peharz *et al.*, 2014) and *cutset networks* (Rahman *et al.*, 2014).

While closely related, these representations differ significantly in their properties, both in terms of their interpretability and their support for tractable queries. Our work considers the *probabilistic sentential decision diagram* (PSDD) (Kisa *et al.*, 2014a), which is perhaps the most powerful circuit proposed to date. Owing to their intricate structure, PSDDs stand out as being exceptionally interpretable: each PSDD parameter represents a conditional probability in the distribution, and the PSDD structure encodes an abundance of conditional independencies (Kisa *et al.*, 2014a). At the computational level, PSDDs support closed-form parameter learning, MAP inference, complex queries (Bekker *et al.*, 2015), and even efficient multiplication of distributions (Shen *et al.*, 2016), which are all exceedingly rare capabilities.

With these desirable properties, a key question is whether the PSDD representation can effectively be learned from data, and be competitive with other models for density estimation, such as Bayesian and Markov networks, or weaker types of tractable circuits. Liang *et al.* (2017) develop the first structure learning algorithm for PSDDs, called LearnPSDD. It uses local operations on the PSDD circuit that maintain the desired circuit properties, while steadily increasing model fit. LearnPSDD achieves state-of-the-art results on a large set of standard benchmarks. In this paper, we give a brief overview of LearnPSDD and its empirical performance.

A second question directly pertains to the explainability of PSDDs. While each PSDD parameter is individually interpretable as a conditional probability, LearnPSDD routinely learns circuits with tens of thousands of parameters, which hinders the interpretability of the model as a whole. Even though tractable learners trade off the likelihood of the model and its parameter count (a proxy for tractability), the learned model may be too large to interpret. To mitigate, we propose an algorithm to shrink PSDD circuits, reducing the number of parameters without affecting the likelihood. Our algorithm finds similar PSDD substructures, as measured by the KL-divergence between their distributions, and merges them. It is supported by an efficient algorithm to compute the KL-divergence between PSDDs.



(a) PSDD circuit (b) Vtree (c) Equivalent distribution (d) Minimal Split on A.

Figure 1: Examples of a PSDD, its vtree and distribution, and a split on an abstract PSDD. (After Liang *et al.* (2017).)

2 Background: PSDDs

Uppercase letters denote Boolean random variables. A lowercase complete instantiation $\mathbf{x}$ of variables $\mathbf{X}$ is a world, and $\mathbf{x} \models \alpha$ denotes that $\mathbf{x}$ satisfies sentence α.

Syntax and Semantics A probabilistic sentential decision diagram (PSDD) is a circuit representations of a joint probability distribution over binary variables. We refer to Kisa *et al.* (2014a) for a technical exposition and give a brief overview here. A PSDD is a parameterized directed acyclic graph, as depicted in Figure 1a. Each inner node is either a logical AND gate with two inputs, or a logical OR gate with an arbitrary number of inputs. The types of nodes alternate. Each terminal (input) node is a univariate distribution: X when X is always true, $\neg X$ when it is always false, or $(\theta : X)$ when it is true with probability θ. Each combination of an OR gate with its AND inputs is called a decision node. The left input to an AND gate is its prime (denoted p) and the right input is its sub (denoted s). The n wires in each decision node are annotated with a normalized probability distribution $\theta_1, \ldots, \theta_n$. Equivalently a decision node is represented as a set $\{(p_1, s_1, \theta_1), \ldots, (p_n, s_n, \theta_n)\}$.

Each PSDD node represents a probability distribution over its random variables. The inputs to an AND gate must represent *decomposable* distributions (i.e. over disjoint sets of variables). This is enforced uniformly throughout the circuit by a variable tree (vtree): a full, binary tree, whose leaves are labeled with variables. Intermediate vtree nodes partition variables into those appearing in the primes and subs of the corresponding PSDD decision nodes; see Figure 1b. Each PSDD node's distribution has an intricate support over which it defines a non-zero probability distribution. We refer to this set of worlds as the *base* of the PSDD node q, denoted $[q]$. For any single possible world and decision node, there is at most one prime input that assigns a non-zero probability to the world. That is, the support of each decision node's prime distributions has to be disjoint (a property called *determinism*). The probability of world $\mathbf{xy}$ according to decision node q factorizes recursively as

$$\Pr_q(\mathbf{xy}) = \theta_i \cdot \Pr_{p_i}(\mathbf{x}) \cdot \Pr s_i(\mathbf{y}) \text{ for } i \text{ s.t. } \mathbf{x} \models [p_i]$$

until it reduces to the univariate distributions at the terminals. Intuitively, each decision node branches based on which sentence $[p_i]$ is true, similar to how decision trees branch on the value of a single variable. We invite the reader to verify that the PSDD in Figure 1a represents the distribution shown in Figure 1c.

Interpretability From a top-down perspective, a PSDD repeatedly decomposes the distribution by conditioning it on the prime bases $[p_i]$. In each conditioned distribution, the prime and sub variables are independent. Independence given a logical sentence is called *context-specific independence* (Boutilier *et al.*, 1996). Moreover, to reach a node q through some path, all the primes on that path must be satisfied; they form the *sub-context* of the node. The disjunction of all sub-contexts forms the node's *context* γ_q. It gives us a way of precisely characterizing the parameter semantics of PSDD. Parameters θ_i of node q are conditional probabilities in root node r's overall distribution:

$$\theta_i = \Pr_r([p_i] \mid \gamma_q).$$

Inference and Parameter Learning PSDDs are a tractable representation: the probability of any assignment $\mathbf{x}$ can be computed in time linear in the PSDD size (its number of parameters), in a single bottom-up pass (Kisa *et al.*, 2014a). Second, PSDDs support efficient complex queries, such as count queries (Bekker *et al.*, 2015) and can be multiplied efficiently (Shen *et al.*, 2016). The maximum-likelihood estimates for the PSDD parameters are calculated in closed form by observing the fraction of complete examples flowing through the wire. That is, out of all the examples that agree with the node context γ_q, the fraction that also agrees with the prime base $[p_{q,i}]$ (Kisa *et al.*, 2014a).

3 PSDD Structure Learning

Liang *et al.* (2017) recently developed the first PSDD structure learning algorithm, called LearnPSDD. The objective of LearnPSDD is to obtain a compact PSDD that fits the data well. This section provides a high-level overview of that work (adapted from Liang *et al.* (2017)).

Operations Two local operations are proposed for LearnPSDD that change the PSDD structure: splitting and cloning. Splitting creates copies of an AND gate by constraining its prime and thereby changing its base. Cloning creates a structurally identical copy of a node but redirects some of the parents of the original node to the copy. A depth parameter d is used to specify the recursion depth to which these nodes are copied. When, $d = 0$, we call the operation minimal, and when d is infinity, we call the operation complete. Figure 1d depicts a minimal split. Node γ is still shared among the copies, as it exceeds depth d.

LearnPSDD Algorithm LearnPSDD incrementally improves the structure of an existing PSDD to better fit the data. In each iteration, the operation to execute is greedily chosen based on the best test-set likelihood improvement per size increment:

$$\text{score} = (\ln \mathcal{L}(r' \mid \mathcal{D}) - \ln \mathcal{L}(r \mid \mathcal{D}))/(\text{size}(r') - \text{size}(r))$$

where r is the original and r' the updated PSDD. The depth parameter d is fixed during learning. It is a critical parameter to tune in order to balance the learning speed and the tractability/explainability of the learned model.

Experiments After being extended to learn ensembles of PSDDs with bagging and EM, LearnPSDD achieves state-of-art results on standard benchmarks for density estimation (Liang *et al.*, 2017). An ensemble of PSDDs is equivalent to a single PSDD with a latent variable. LearnPSDD surpasses the state of the art[1] on 6 out of 20 datasets; see Table 1. This experiment shows that LearnPSDD performs competitively, despite the fact that PSDDs are a more interpretable, tractable, and restrictive representation than their alternatives.

4 Shrinking PSDDs

Both the interpretability and the tractability of a learned PSDD depend critically on its size. In LearnPSDD, this is largely a function of parameter d. This section reports ongoing work to control the size of the PSDD during learning by merging similar substructures with an algorithm called MergePSDD. This helps find the right trade-off between the number of parameters and the data fit, and eliminates the need to tune d.

Merge Operation Our merge operation takes as input two PSDD decision nodes that respect the same vtree and have the same base. It removes the larger node and redirects its parents to the remaining one; see Figure 2. The parameters of the modified substructure need to be re-estimated on the union of the datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ that flowed through the original nodes.

[1]Best-to-date is the best of ACMN (Lowd and Rooshenas, 2013), ID-SPN (Rooshenas and Lowd, 2014), SPN-SVD (Tameem Adel, 2015), ECNet (Rahman and Gogate, 2016a) and Merged L-SPN (Rahman and Gogate, 2016b).

Table 1: Comparison of test-data log-likelihood between LearnPSDD and the state of the art († denotes best).

Dataset	|Var|	LearnPSDD Ensemble	Best-to-Date
NLTCS	16	$-5.99^{\dagger}$	-6.00
MSNBC	17	$-6.04^{\dagger}$	$-6.04^{\dagger}$
KDD	64	$-2.11^{\dagger}$	-2.12
Plants	69	-13.02	$-11.99^{\dagger}$
Audio	100	-39.94	$-39.49^{\dagger}$
Jester	100	-51.29	$-41.11^{\dagger}$
Netflix	100	$-55.71^{\dagger}$	-55.84
Accidents	111	-30.16	$-24.87^{\dagger}$
Retail	135	$-10.72^{\dagger}$	-10.78
Pumsb-Star	163	-26.12	$-22.40^{\dagger}$
DNA	180	-88.01	$-80.03^{\dagger}$
Kosarek	190	$-10.52^{\dagger}$	-10.54
MSWeb	294	-9.89	$-9.22^{\dagger}$
Book	500	-34.97	$-30.18^{\dagger}$
EachMovie	500	-58.01	$-51.14^{\dagger}$
WebKB	839	-161.09	$-150.10^{\dagger}$
Reuters-52	889	-89.61	$-80.66^{\dagger}$
20NewsGrp.	910	-155.97	$-150.88^{\dagger}$
BBC	1058	-253.19	$-233.26^{\dagger}$
AD	1556	-31.78	$-14.36^{\dagger}$

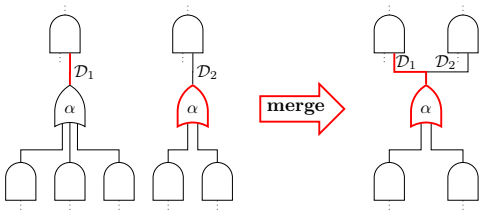


Figure 2: A merge operation. To-be-merged nodes have the same base α. The node with smaller size is retained.

Merge Heuristic MergePSDD incrementally improves the tractability of the learned model by repeatedly invoking the merge operation. Whereas each merge decreases the size of the learned model, it may increase or decrease the test-set likelihood. A heuristic is needed to select merges that are least likely to decrease the quality of the model. It is natural to only merge PSDDs that represent similar probability distributions. Similarity between distributions is commonly measured by their KL-divergence. However, KL-divergence cannot be directly applied to PSDDs: even if two PSDDs share the same set of variables, they may not share the same base (support), which invalidates the definition of KL-divergence. Therefore, we generalize the KL-divergence to the *Intersectional Divergence*, which is always defined between PSDDs that respect the same vtree.

Definition 1. (Intersectional Divergence D_I)
Given two PSDDs respecting the same vtree, m and n

$$D_I(m \parallel n) \stackrel{def}{=} \sum_{\mathbf{x} \models [m] \wedge [n]} \Pr_m(\mathbf{x}) \log \frac{\Pr_m(\mathbf{x})}{\Pr_n(\mathbf{x})}$$

This definition applies beyond PSDDs. Intuitively, it is the KL-divergence computed on the intersection of the supports of the two distributions. For nodes with the same base, intersectional and KL-divergence are equivalent. This is the way we use it in MergePSDD. Al-

Algorithm 1 `intersectional-divergence`(m, n)

input: PSDDs m and n that respect the same vtree.
output: Intersectional divergence $D_I(m, n)$
note: `pr-constraint`$(a, [b])$ is the probability of $[b]$ in PSDD a's induced distribution. See algorithm in Choi *et al.* (2015).
note: `cache` is loaded with divergences between terminals.
main:

if $(m, n) \in$ in `cache` **then return** `cache`$[(m, n)]$
$\rho \leftarrow 0$
for each (p_i, s_i, θ_i) in decision node m **do**
  for each (r_j, t_j, β_j) in decision node n **do**
    $\rho_{11} \leftarrow$ `pr-constraint`$(s_i, [t_j])$
    $\rho_{12} \leftarrow$ `pr-constraint`$(p_i, [r_j])$
    $\rho_{13} \leftarrow \theta_i \log \frac{\theta_i}{\beta_j}$
    $\rho_{21} \leftarrow$ `intersectional-divergence`(p_i, r_j)
    $\rho_{31} \leftarrow$ `intersectional-divergence`(s_i, t_j)
    $\rho \leftarrow \rho + \rho_{11}\rho_{12}\rho_{13} + \theta_i\rho_{11}\rho_{21} + \theta_i\rho_{12}\rho_{31}$
`cache`$[(m, n)] \leftarrow \rho$
return ρ

gorithm 1 computes D_I efficiently (in quadratic time) using the following recursion.

Theorem 1. (D_I Calculation) Given a PSDD node $m = \{(p_1, s_1, \theta_1), (p_2, s_2, \theta_2) \dots\}$ and PSDD node $n = \{(r_1, t_1, \beta_1), (r_2, t_2, \beta_2)\}$ respecting the same vtree,

$$D_I(m \parallel n) = \sum_{i,j} \sum_{x \models [p_i] \wedge [r_j]} \sum_{y \models [s_i] \wedge [t_j]} \Pr_{p_i}(x) \Pr_{s_i}(y) \theta_i \left\{ \log \frac{\Pr_{p_i}(x)\theta_i}{\Pr_{r_j}(x)\beta_j} + \log \frac{\Pr_{s_i}(y)}{\Pr_{t_j}(y)} \right\}$$

$$= \sum_{i,j} \Pr_{s_i}([t_j]) \Pr_{p_i}([r_j]) D_{KL}(\theta_i \parallel \beta_j) + \theta_i \Pr_{s_i}([t_j]) D_I(p_i \parallel r_j) + \theta_i \Pr_{p_i}([r_j]) D_I(s_i \parallel t_j)$$

where $D_{KL}(\theta_i \parallel \beta_j) = \theta_i \log \frac{\theta_i}{\beta_j}$.

Merging Algorithm The MergePSDD algorithm starts from a (large) initial learned PSDD, for example obtained from LearnPSDD. It considers vtree nodes bottom-up. In each iteration, it finds all pairwise combinations of PSDD decision nodes that (i) respect the considered vtree node and (ii) have the same base. These pairs are candidates for a merge: the pair that yields the lowest intersectional divergence is chosen. The merge is first simulated and only permanently executed if the likelihood on validation data does not decrease.

Experiments We evaluate the effectiveness of MergePSDD with a focus on reduction in size. Experiments were conducted on a 16-core 2.6GHz Intel Xeon server with 256GB RAM. For each dataset in Table 2, two different PSDDs were obtained from LearnPSDD, using either greedy operations (complete split) or frugal operations (80% minimal operations and 20% depth-3 operations). Greedy LearnPSDD maximizes the learning speed, but PSDD size may be wasted. Frugal LearnPSDD better balances between size and learning speed. LearnPSDD was run until reaching the desired test-set likelihood, with a maximum of 24 hours. As expected, greedy LearnPSDD learns much larger models than frugal LearnPSDD; see Table 2.

Table 2: Number of parameters in PSDDs learned by LearnPSDD using frugal or greedy operations, and MergePSDD. LL is the desired test-set log-likelihood.

Dataset	Target LL	Frugal	Greedy	MergePSDD
NLTCS	−6.08	7491	31669	23471
MSNBC	−6.05	11074	17687	12943
KDD	−2.19	13814	29429	18921
Plants	−16.98	12021	12398	11574
Audio	−44.64	5804	5494	5389
Jester	−56.21	11774	16149	12349

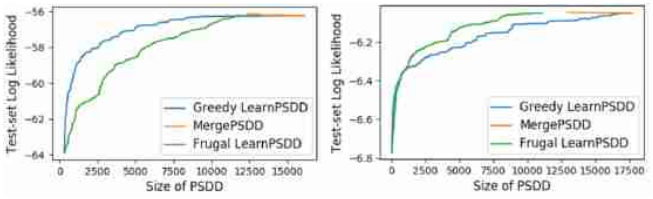


Figure 3: MergePSDD prunes away "unnecessary" PSDD structures while slightly improving performance.
Left: on dataset Jester. Right: on dataset MSNBC.

MergePSDD was run on the models learned by greedy LearnPSDD for until all potential merge operations were exhausted, with a maximum of 6 hours. As shown in Figure 3 (comparing the left end of the brown line with the right end of the green line), MergePSDD effectively shrank the gap in size between the models learned by greedy LearnPSDD and frugal LearnPSDD. A full result on 6 datasets is reported in Table 2. It shows that MergePSDD is able to effectively reduce PSDD size, making the models more tractable and interpretable, without sacrificing too much model quality, by virtue of its KL-divergence heuristic.

5 Conclusions

The two questions raised in this paper both received positive answers. First, LearnPSDD demonstrates the competitiveness of PSDDs in density estimation, even with structure learning from data instead of logical constraints (Kisa *et al.*, 2014b). Second, MergePSDD finds a better trade-off between learning speed and model size. Moreover, it was able to simplify PSDDs without a significant loss in quality. For future work, we hope to further decrease the size of PSDDs to any number required to make the model interpretable in practice.

Acknowledgements The authors thank Arthur Choi and Jessa Bekker for helpful discussions. This work is partially supported by NSF grants #IIS-1657613, #IIS-1633857 and DARPA XAI grant #N66001-17-2-4032.

References

Jessa Bekker, Jesse Davis, Arthur Choi, Adnan Darwiche, and Guy Van den Broeck. Tractable learning for complex probability queries. In *Proceedings of NIPS*, pages 2242–2250, 2015.

Craig Boutilier, Nir Friedman, Moises Goldszmidt, and Daphne Koller. Context-specific independence in Bayesian networks. In *Proceedings of UAI*, pages 115–123, 1996.

A. Chechetka and C. Guestrin. Efficient principled learning of thin junction trees. In *Proceedings of NIPS*, pages 273–280, 2007.

Arthur Choi, Guy Van den Broeck, and Adnan Darwiche. Tractable learning for structured probability spaces: A case study in learning preference distributions. In *Proceedings of IJCAI*, 2015.

A. Darwiche. A differential approach to inference in Bayesian networks. *JACM*, 50(3):280–305, 2003.

Aaron Dennis and Dan Ventura. Greedy structure search for sum-product networks. *Proceedings of IJCAI*, 2015.

P. Domingos, M. Niepert, and D. Lowd (Eds.). ICML workshop on learning tractable probabilistic models. 2014.

Robert Gens and Pedro Domingos. Learning the structure of sum-product networks. In *Proceedings of ICML*, pages 873–880, 2013.

Doga Kisa, Guy Van den Broeck, Arthur Choi, and Adnan Darwiche. Probabilistic sentential decision diagrams. In *Proceedings of KR*, pages 1–10, 2014.

Doga Kisa, Guy Van den Broeck, Arthur Choi, and Adnan Darwiche. Probabilistic sentential decision diagrams: Learning with massive logical constraints. In *ICML Workshop on Learning Tractable Probabilistic Models (LTPM)*, 2014.

Yitao Liang, Jessa Bekker, and Guy Van den Broeck. Learning the structure of probabilistic sentential decision diagrams. In *Proceedings of UAI*, 2017.

D. Lowd and P. Domingos. Learning arithmetic circuits. In *Proceedings of UAI*, pages 383–392, 2008.

Daniel Lowd and Amirmohammad Rooshenas. Learning markov networks with arithmetic circuits. In *Proceedings of AISTATS*, pages 406–414, 2013.

Nicola Di Mauro and Antonio Vergari. PGM tutorial on learning sum-product networks. 2016.

Marina Meila and Michael I Jordan. Learning with mixtures of trees. *JMLR*, 1:1–48, 2000.

M. Narasimhan and J. Bilmes. PAC-learning bounded tree-width graphical models. In *Proceedings of UAI*, 2004.

Robert Peharz, Robert Gens, and Pedro Domingos. Learning selective sum-product networks. In *LTPM workshop*, 2014.

Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 689–690. IEEE, 2011.

Tahrima Rahman and Vibhav Gogate. Learning ensembles of cutset networks. In *Proceedings of AAAI*, pages 3301–3307, 2016.

Tahrima Rahman and Vibhav Gogate. Merging strategies for sum-product networks: From trees to graphs. In *Proceedings of UAI*, 2016.

T. Rahman, P. Kothalkar, and V. Gogate. Cutset networks: A simple, tractable, and scalable approach for improving the accuracy of Chow-Liu trees. In *Proceedings of ECML PKDD*, pages 630–645, 2014.

Amirmohammad Rooshenas and Daniel Lowd. Learning sum-product networks with direct and indirect variable interactions. In *Proceedings of ICML*, pages 710–718, 2014.

Yujia Shen, Arthur Choi, and Adnan Darwiche. Tractable operations for arithmetic circuits of probabilistic models. In *Proceedings of NIPS*, 2016.

Ali Ghodsi Tameem Adel, David Balduzzi. Learning the structure of sum-product networks via an svd-based algorithm. *Proceedings of UAI*, pages 32–41, 2015.

Explainable AI: Beware of Inmates Running the Asylum

Or: How I Learnt to Stop Worrying and Love the Social and Behavioural Sciences

Tim Miller[*] and **Piers Howe**[†] and **Liz Sonenberg**[*]
[*]School of Computing and Information Systems
[†]Melbourne School of Psychological Sciences
University of Melbourne, Australia
{tmiller,pdhowe,l.sonenberg}@unimelb.edu.au

Abstract

In his seminal book *The Inmates are Running the Asylum: Why High-Tech Products Drive Us Crazy And How To Restore The Sanity* [2004, Sams Indianapolis, IN, USA], Alan Cooper argues that a major reason why software is often poorly designed (from a user perspective) is that programmers are in charge of design decisions, rather than interaction designers. As a result, programmers design software for themselves, rather than for their target audience; a phenomenon he refers to as the '*inmates running the asylum*'. This paper argues that explainable AI risks a similar fate. While the reemergence of explainable AI is positive, this paper argues most of us as AI researchers are building explanatory agents for ourselves, rather than for the intended users. But explainable AI is more likely to succeed if researchers and practitioners understand, adopt, implement, and improve models from the vast and valuable bodies of research in philosophy, psychology, and cognitive science; and if evaluation of these models is focused more on people than on technology. From a light scan of literature, we demonstrate that there is considerable scope to infuse more results from the social and behavioural sciences into explainable AI, and present some key results from these fields that are relevant to explainable AI.

1 Introduction

> "*Causal explanation is first and foremost a form of social interaction. One speaks of giving causal explanations, but not attributions, perceptions, comprehensions, categorizations, or memories. The verb to explain is a three-place predicate:* Someone *explains* something *to* someone. *Causal explanation takes the form of conversation and is thus subject to the rules of conversation.*" — Hilton [1990].

The term "explainable AI" has regained traction again recently, after being considered important in the 80s and 90s in expert systems particularly; see [Chandrasekaran *et al.*, 1989], [Swartout and Moore, 1993], and [Buchanan and Shortliffe, 1984]. High visibility of the term, sometimes abbreviated XAI, is seen in grant solicitations [DARPA, 2016] and in the popular press [Nott, 2017]. One area of explainable AI receiving attention is explicit *explanation*, on which we say more below.

While the title of the paper is deliberately tongue-in-cheek, the parallels with Cooper [2004] are real: leaving decisions about what constitutes a good explanation of complex decision-making models to the experts who understand these models the best is likely to result in failure in many cases. Instead, models should be built on an understanding of explanation, and should be evaluated using data from human behavioural studies.

In Section 2, we describe a simple scan of the 23 articles posted as 'Related Work' on the workshop web page. We looked at two attributes: whether the papers were built on research from philosophy, psychology, cognitive science, or human factors; and whether the reported evaluations involved human behavioural studies. The outcome of this scan supports the hypothesis that ideas from social sciences and human factors are not sufficiently visible in the field.

In Section 3, we present some key bodies of work on explanation and related topics from social and behavioural sciences that will be of interest to those in explainable AI, and briefly discuss what their impact could be.

2 Explainable AI Survey

To gather some data to test the hypothesis that the social sciences and human behavioural studies are not having enough impact in explainable AI, a short literature survey was undertaken. This survey is not intended to be even close to comprehensive – it is merely illustrative. However, the results that it shows are reflective of many other papers in the area that the authors have read.

2.1 Selected Papers

The articles surveyed were taken from the 'Related Work' list that was posted on the website for the IJ-

CAI 2017 Explainable AI workshop[1] as of 16 May 2017 — the workshop to which this paper is submitted. In total 23 articles were on the list, although one was not included in the results as described later. This list can be found in Appendix A.

As noted already, this list is far from comprehensive, however, it is a useful list for two reasons:

1. First, it was compiled by the explainable AI community: the organisers of the conference requested that people send related papers to be added to the list. As such, it represents at least a subset of what the community see at the moment as highly relevant papers for researchers in explainable AI.
2. Second, it is objective from the perspective of the authors of this paper. We did not contribute to the list, so the selection is not biased by our argument.

While the authors of some of the listed papers may not consider their work as explainable AI, almost all of the papers were describing methods for automatically generating explanations of some type.

The paper that was excluded is Tania Lombrozo's survey paper on explanation research in cognitive science [Lombrozo, 2012]. This is not an explainable AI paper – indeed, it summarises one of the bodies of work of which we argue people should be more aware.

2.2 Survey Method

The survey was lightweight: it only looked for evidence that the presented research was somehow influenced by a scientific understanding of explanations, and that the evaluations were performed using data derived from human behaviour studies or similar. We categorised the papers on the three items of interest, with the criteria for the scores as follows:

1. *On topic*: Each paper was categorised as either being about explainable AI or not, based on our understanding of the topic. It is possible that some papers were included on the workshop website because they presented good challenges or potentially useful approaches, but were not papers about explanation *per se*, in which case they were 'off topic'.
2. *Data Driven*: Each paper was given a score from 0–2 inclusive.

   A score of 1 was given if and only if one or more of the *references* of the paper was an article on explanation in social science, meaning that: (a) explanation or causal attribution as done by humans is one of the main topics of the referenced article(s); (b) the referenced article(s) validated their claims using data collected from human behaviour experiments; and (c) the referenced article(s) appear in a non-computer science venue *or* in a computer science venue but contributed to the understanding of explanation in general (outside of AI).

   A score of 2 was given if and only if (a), (b), and (c) above held, and the survey article (not the referenced article) described an algorithm for automatically generating explanations and this algorithm was derived from data from the social sciences. In other words, the algorithm is explicitly based on a model from one or more of the references.

   A score of 0 was given for any other paper; that is, no references satisfying (a), (b), and (c).
3. *Validation*: Each paper was given a binary 0/1. A score of 1 was given if and only if the evaluation in the survey article (note, not the referenced article) was based on data from human behavioural studies. Even if the algorithm is categorised as data driven, we argue that it is still important to test that the assumptions and trade-offs made are suitable. It is therefore necessary to (eventually) perform behavioural studies to test if the explanations produced by the algorithm are appropriate for humans.

2.3 Results

Table 1 shows the results for the survey. Results for each of the surveyed articles are available in Appendix B. Five papers were deemed 'off topic', however, the results are included because we could not know the intent of those who submitted articles to the reading list. For the 'Data driven' entry, column 'N' means that we were unsure about the reference. In this case, one paper had a reference to a cognitive science article of which we were unable to locate a copy. For the 'Validation' entry, column 'N' means 'not applicable': three papers were categorised as not applicable because their status were not research articles, but review articles or position papers, and thus, they did not present any algorithm or model to evaluate.

	On topic (17 articles)				**Off topic** (5 articles)			
Criterion	N	0	1	2	N	0	1	2
Data driven	1	11	4	1	0	4	0	1
Validation	3	10	4	—	0	4	1	—

Table 1: Results on small survey

These results show that for the on-topic papers, only four articles referenced relevant social science research, and only one of them truly built a model on this. Further, serious human behavioural experiments are not currently being undertaken. For off topic papers, the results are similar: limited input from social sciences and limited human behavioural experiments.

2.4 Discussion

The results, while only on a small set of papers, provide evidence that many models being used in explainable

[1] See `http://home.earthlink.net/~dwaha/research/meetings/ijcai17-xai/`.

AI research are not building on current scientific understanding of explanation. Further, human behavioural experiments are rare — something that needs to change for us to produce useful explanatory agents.

It is important to note that we are not interpreting the above observations to say that there is not a lot of excellent research on explainable AI. For example, consider Ribeiro *et al.* [2016], who have done some remarkable work on explaining classifiers, and yet scored '0' on the 'Data Driven' criteria. Instead, they have constructed their own understanding of how people evaluate explanations for their particular field over a series of human behavioural experiments. However, developing such an understanding will not always be required or even possible for many researchers, so in these cases, building on social science research is a sound place to start.

3 Where to? A Brief Pointer to Relevant Work

In the different sub-fields of social sciences, there are several hundred articles on explanation, not to mention another entire field on causality. It is not feasible to expect that AI researchers and practitioners can navigate this entire field in addition to their own field of expertise, especially considering that the relevant literature is written for a different audience. However, there are some key areas that should be of interest to those in explainable AI, which we outline in this section.

Miller [2017] provides an in-depth survey of all articles cited in this section plus many other relevant articles, and draws parallels between this work and explainable AI. Here, we present several key ideas from that work to demonstrate ways that models of explainable AI can benefit from models of human explanation.

3.1 Contrastive Explanation

Perhaps the most important result from this work is that explanations are *contrastive*; or more accurately, *why–questions* are contrastive. That is, why–questions are of the form "*Why* P *rather than* Q*?*", where P is the *fact* that requires explanation, and Q is some *foil* case that was expected. Most philosophers, psychologists, and cognitive scientists in this field assert that *all* why–questions are contrastive (e.g. see [Hilton, 1990; Lombrozo, 2012; Miller, 2017]), and that when people ask for an explanation "*Why* P*?*", there is an implicit contrast case. Importantly, the contrast case helps to frame the possible answers and make them relevant [Hilton, 1990]. For example, explaining "*Why did Mr. Jones open the window?*" with the response "*Because he was hot*" is not useful if the implied foil is Mr. Jones turning on the air conditioner, as this explains both the fact and the foil; or if the implied foil was why Ms. Smith, who was sitting closer to the window, did not open it instead, as the cited cause does not refer to a cause of Ms. Smith's lack of action.

This is a challenge for explainable AI, because it may not be easy to elicit a contrast case from an observer. However, it is also an opportunity: as Lipton [1990] argues, answering a contrastive question is often easier than giving a full cause attribution because one only needs to understand the difference between the two cases, so one can provide a complete explanation without determining or even knowing all causes of the event.

3.2 Attribution Theory

Attribution theory is the study of how people attribute causes to events; something that is necessary to provide explanations. It is common to divide the types of attribution into two classes: (1) causal attribution of social behaviour (called *social attribution*); and (2) general causal attribution.

Social Attribution The book from Malle [2004], based on a large body of work from himself and other researchers in the field, describes a mature model of how people explain behaviour of others using folk psychology. He argues that people attribute behaviour based on the beliefs, desires, intentions, and traits of people, and presents theories for why failed actions are described differently than successful actions; the former often referring to some precondition that could not be satisfied.

Malle's work provides a solid foundation on which to build social attribution and explainable AI models for many sub-fields of artificial intelligence. Social attribution is important for systems in which *intentional action* will be cited as a cause; in particular, it is important for systems doing deliberative reasoning, and the concepts used in his work are closely linked to that of systems such as *belief-desire-intention* models [Rao and Georgeff, 1995] and AI planning.

Causal Connection Research on how people connect causes shows that they do so by undertaking a mental simulation of what *would have happened* had some other event turned out differently [Kahneman and Tversky, 1982; Hilton *et al.*, 2005; McCloy and Byrne, 2000].

However, simulating an entire causal chain is infeasible in most cases, so cognitive scientists and social psychologists have studied how people decide which events to 'undo' (the counterfactuals) to determine cause. For example, people tend to undo more proximal causes over more distal causes [Miller and Gunasegaram, 1990], abnormal events over normal events [Kahneman and Tversky, 1982], and events that are considered more 'controllable' [Girotto *et al.*, 1991].

For explainable AI models, these heuristics are useful from a computational perspective in large causal chains, in which causal attribution is intractable in many cases [Eiter and Lukasiewicz, 2002]. Effectively, they can be used to 'skip-over' or *discount* some events and not consider their counterfactuals, while being consistent with what an explainee would expect.

3.3 Explanation Selection

An important body of work is concerned with explanation *selection*. People rarely expect an explanation that consists of an actual and complete cause of an event. Instead, explainers select one or two causes and present

these as *the* explanation. Explainees are typically able to 'fill in' their own causal understanding from just these. Thus, some causes are better explanations than others: events that are 'closer' to the fact in question in the causal chain are preferred over more distal events [Miller and Gunasegaram, 1990], but people will 'trace through' closer events to more distal events if those distal events are human actions [Hilton *et al.*, 2005] or abnormal events [Hilton and Slugoski, 1986].

In AI, perhaps some models are simple enough that explanation selection would not be valuable, or visualisation would provide a powerful medium to show many causes at once. However, for causal chains with than a handful of causes, we argue that explanation selection can be used to simplify and/or prioritise explanations.

3.4 Explanation Evaluation

The work discussed in this section so far looks at how explainees generate and select explanations. There is also a body of work that studies how people evaluate the quality of explanations provided to them. The most important finding from this work is that the probability that the cited cause is actually true is not the most important criteria people use [Hilton, 1996]. Instead, people judge explanations based on so-called *pragmatic influences* of causes, which include criteria such as usefulness, relevance, etc. [Slugoski *et al.*, 1993].

Recent work shows that people prefer explanations that are *simpler* (cite few causes) [Lombrozo, 2007], more *general* (they explain more events) [Lombrozo, 2007], and *coherent* (consistent with prior knowledge) [Thagard, 1989]. In particular, Lombrozo [2007] shows that the people disproportionately prefer simpler explanations over more likely explanations.

These criteria are important to any work in explainable AI. Giving simpler explanations that increase the likelihood that the observer both *understands* and *accepts* the explanation may be more useful to establish trust, if this is the primary goal of the explanation. Learning from these and adding them as objective criteria to models of explainable AI is important.

3.5 Explanation as Conversation

Finally, it is important to remember that explanations are interactive conversations, and that people typically abide by certain rules of conversation [Hilton, 1990]. *Grice's maxims* [Grice, 1975] are the most well-known and widely accepted rules of conversation. In short, they say that in a conversation, people consider the following: (a) quality; (b) quantity; (c) relation; and (d) manner. Coarsely, these respectively mean: only say what you believe; only say as much as is necessary; only say what is relevant; and say it in a nice way. Hilton [1990] argues that as explanations are conversations, they follow these maxims. There is body of research that demonstrates people do follow these maxims, as discussed by Miller [2017].

Note that we are not arguing that explanations must be text or verbal. However, explanations presented in a visual way, for example, should have similar properties, and these maxims offer a useful set of objective criteria.

3.6 Where not to go

Finally, we discuss work that we believe should be discounted in explainable AI. Specifically, two well-known theories of explanation, sometimes cited and used in explainable AI articles, are the *logically deductive model* of explanation [Hempel and Oppenheim, 1948], and the *co-variation model* [Kelley, 1967]; both of which have had significant impact. However, since its publication, researchers found that the logically-deductive model was inconsistent in many ways, and instead derived new models of explanation. Similarly, the co-variation model was found be problematic and did not account for many facets of human explanation [Malle, 2011], so was refined into other models, such as those of abnormality described in Section 3.3.

While these models are still cited as part of the history of research in explanation, they are no longer considered valid models of human explanation in cognitive and social science. We contend, therefore, that explainable AI models should build on these newer models, which are widely accepted, rather than these earlier models.

4 Conclusions

We argued that existing models of how people generate, select, present, and evaluate explanations are highly relevant to explainable AI. Via a brief survey of articles, we provide evidence that little research on explainable AI draws on such models. Although the survey was limited, it is clear from our readings that the observation holds more generally. We pointed to a handful of key articles that we believe could be important, but for a proper presentation and discussion of these, see Miller [2017].

We encourage researchers and practitioners in explainable AI to collaborate with researchers and practitioners from the social and behavioural sciences, to inform both model design and human behavioural experiments. We do not advocate that every paper on explainable AI should be accompanied by human behavioural experiments — proxy studies are valid ways to evaluate models of explanation, especially those in early development, and computational problems are also of interest. However, we support the emphasis in the recent DARPA solicitation [DARPA, 2016] on reaching "human-in-the-loop techniques that developers can use ... for more intensive human evaluations,' and agree with Doshi-Velez and Kim [2017] that to have a real-world impact, "it is essential that we as a community respect the time and effort involved to do such evaluations."

We hope that readers of this paper and participants in the workshop agree with our position and, where feasible, adopt existing models and methods to reduce the risk that it is only the inmates that are running the asylum.

References

[Buchanan and Shortliffe, 1984] Bruce Buchanan and Edward Shortliffe. *Rule-based expert systems: the MYCIN experiments of the Stanford Heuristic Programming Project*. Addison-Wesley, 1984.

[Chandrasekaran *et al.*, 1989] B Chandrasekaran, Michael C. Tanner, and John R. Josephson. Explaining control strategies in problem solving. *IEEE Expert*, 4(1):9–15, 1989.

[Cooper, 2004] Alan Cooper. *The inmates are running the asylum: Why high-tech products drive us crazy and how to restore the sanity*. Sams, IN, USA, 2004.

[DARPA, 2016] DARPA. Explainable artificial intelligence (XAI) program. `http://www.darpa.mil/program/explainable-artificial-intelligence`, 2016. Full solicitation at `http://www.darpa.mil/attachments/DARPA-BAA-16-53.pdf`.

[Doshi-Velez and Kim, 2017] F. Doshi-Velez and B. Kim. Towards A Rigorous Science of Interpretable Machine Learning. *ArXiv e-prints*, 1702.08608, 2017.

[Eiter and Lukasiewicz, 2002] Thomas Eiter and Thomas Lukasiewicz. Complexity results for structure-based causality. *Artificial Intelligence*, 142(1):53–89, 2002.

[Girotto *et al.*, 1991] Vittorio Girotto, Paolo Legrenzi, and Antonio Rizzo. Event controllability in counterfactual thinking. *Acta Psychologica*, 78(1):111–133, 1991.

[Grice, 1975] Herbert P Grice. Logic and conversation. In *Syntax and semantics 3: Speech arts*, pages 41–58. New York: Academic Press, 1975.

[Hempel and Oppenheim, 1948] Carl G Hempel and Paul Oppenheim. Studies in the logic of explanation. *Philosophy of Science*, 15(2):135–175, 1948.

[Hilton and Slugoski, 1986] Denis J Hilton and Ben R Slugoski. Knowledge-based causal attribution: The abnormal conditions focus model. *Psychological review*, 93(1):75, 1986.

[Hilton *et al.*, 2005] Denis J. Hilton, John L. McClure, and R. Slugoski, Ben. The course of events: Counterfactuals, causal sequences and explanation. In *The Psychology of Counterfactual Thinking*. 2005.

[Hilton, 1990] Denis J Hilton. Conversational processes and causal explanation. *Psychological Bulletin*, 107(1):65–81, 1990.

[Hilton, 1996] Denis J Hilton. Mental models and causal explanation: Judgements of probable cause and explanatory relevance. *Thinking & Reasoning*, 2(4):273–308, 1996.

[Kahneman and Tversky, 1982] Daniel Kahneman and Amos Tversky. The simulation heuristic. In P. Slovic D. Kahneman and A. Tversky, editors, *Judgment under Uncertainty: Heuristics and Biases*. New York: Cambridge University Press, 1982.

[Kelley, 1967] Harold H Kelley. Attribution theory in social psychology. In *Nebraska symposium on motivation*, pages 192–238. Uni. Nebraska Press, 1967.

[Lipton, 1990] Peter Lipton. Contrastive explanation. *Royal Institute of Philosophy Supplement*, 27:247–266, 1990.

[Lombrozo, 2007] Tania Lombrozo. Simplicity and probability in causal explanation. *Cognitive psychology*, 55(3):232–257, 2007.

[Lombrozo, 2012] Tania Lombrozo. Explanation and abductive inference. *Oxford handbook of thinking and reasoning*, pages 260–276, 2012.

[Malle, 2004] Bertram F Malle. *How the mind explains behavior: Folk explanations, meaning, and social interaction*. MIT Press, 2004.

[Malle, 2011] Bertram F Malle. Time to give up the dogmas of attribution: An alternative theory of behavior explanation. *Advances in Experimental Social Psychology*, 44(1):297–311, 2011.

[McCloy and Byrne, 2000] Rachel McCloy and Ruth MJ Byrne. Counterfactual thinking about controllable events. *Memory & Cognition*, 28(6):1071–1078, 2000.

[Miller and Gunasegaram, 1990] Dale T Miller and Saku Gunasegaram. Temporal order and the perceived mutability of events: Implications for blame assignment. *Journal of personality and social psychology*, 59(6):1111, 1990.

[Miller, 2017] Tim Miller. Explainable AI: Insights from the social sciences. *ArXiv e-prints*, 1706.07269, 2017. `https://arxiv.org/abs/1706.07269`.

[Nott, 2017] George Nott. 'Explainable Artificial Intelligence': Cracking open the black box of AI. *Computer World*, 4 2017. `https://www.computerworld.com.au/article/617359/`.

[Rao and Georgeff, 1995] Anand S Rao and Michael P Georgeff. Bdi agents: From theory to practice. In *ICMAS*, volume 95, pages 312–319, 1995.

[Ribeiro *et al.*, 2016] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the Int. Conf.x on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

[Slugoski *et al.*, 1993] Ben R Slugoski, Mansur Lalljee, Roger Lamb, and Gerald P Ginsburg. Attribution in conversational context: Effect of mutual knowledge on explanation-giving. *European Journal of Social Psychology*, 23(3):219–238, 1993.

[Swartout and Moore, 1993] William R Swartout and Johanna D Moore. Explanation in second generation expert systems. In *Second generation expert systems*, pages 543–585. Springer, 1993.

[Thagard, 1989] Paul Thagard. Explanatory coherence. *Behavioral and Brain Sciences*, 12(03):435–467, 1989.

A List of Papers Surveyed

Taken from the 'Related Work' list posted on the website for the IJCAI 2017 Explainable AI workshop[2] as of 16 May 2017.

1. Chakraborti, T., Sreedharan, S., Zhang, Y., & Kambhampati, S. (2017). Plan explanations as model reconciliation: Moving beyond explanation as soliloquy. To appear in Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. Melbourne, Australia: AAAI Press.
2. Cheng, H., et al. (2014) SRI-Sarnoff Aurora at TRECVid 2014: Multimedia event detection and recounting.
3. Doshi-Velez, F., & Kim, B. (2017). A roadmap for a rigorous science of interpretability. (arXiv:1702.08608)
4. Elhoseiny, M., Liu, J., Cheng, H., Sawhney, H., & Elgammal, A. (2015). Zero-shot event detection by multimodal distributional semantic embedding of videos. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (pp. 3478-3486). Phoenix, AZ: AAAI Press.
5. Hendricks, L.A, Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., & Darrell, T. (2016). Generating visual explanations. (arXiv:1603.08507v1)
6. Kofod-Petersen, A., Cassens, J., & Aamodt, A. (2008). Explanatory capabilities in the CREEK knowledge-intensive case-based reasoner. Frontiers in Artificial Intelligence and Applications, 173, 28-35.
7. Kulesza, T., Burnett, M., Wong, W. K., & Stumpf, S. (2015). Principles of explanatory debugging to personalize interactive machine learning. Proceedings of the Twentieth International Conference on Intelligent User Interfaces (pp. 126-137). Atlanta, GA: ACM Press.
8. Lake, B.H., Salakhutdinov, R., & Tenenbaum, J.B. (2015). Human-level concept learning through probabilistic program induction. Science, 350, 1332-1338.
9. Langley, P., Meadows, B., Sridharan, M., & Choi, D. (2017). Explainable agency for intelligent autonomous systems. In Proceedings of the Twenty-Ninth Annual Conference on Innovative Applications of Artificial Intelligence. San Francisco: AAAI Press.
10. Lécué, F. (2012). Diagnosing changes in an ontology stream: A DL reasoning approach. In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence. Toronto, Ontario, Canada: AAAI Press.
11. Letham, B., Rudin. C., McCormick, T., and Madigan, D. (2015). Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. Annals of Applied Statistics, 9(3), 1350-137.
12. Lombrozo, T. (2012). Explanation and abductive inference. Oxford Handbook of Thinking And Reasoning (pp. 260-276).
13. Martens, D., & Provost, F. (2014). Explaining data-driven document classifications. MIS Quarterly, 38(1), 73-99.
14. Ribeiro, M.T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. Human Centered Machine Learning: Papers from the CHI Workshop. (arXiv:1602.04938v1)
15. Rosenthal, S., Selvaraj, S. P., & Veloso, M. (2016). Verbalization: Narration of autonomous mobile robot experience. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. New York, NY: AAAI Press.
16. Sheh, R.K. (2017). "Why did you do that?" Explainable intelligent robots. In K. Talamadupula, S. Sohrabi, L. Michael, & B. Srivastava (Eds.) Human-Aware Artificial Intelligence: Papers from the AAAI Workshop (Technical Report WS-17-11). San Francisco, CA: AAAI Press.
17. Si, Z. and Zhu, S. (2013). Learning AND-OR templates for object recognition and detection. IEEE Transactions On Pattern Analysis and Machine Intelligence, 35(9), 2189-2205.
18. Shwartz-Ziv, R. & Tishby, N. (2017). Opening the black box of deep neural networks via information. (arXiv:1703.00810 [cs.LG])
19. Sormo, F., Cassens, J., & Aamodt, A. (2005). Explanation in case-based reasoning: Perspectives and goals. Artificial Intelligence Review, 24(2), 109-143.
20. Swartout, W., Paris, C., & Moore, J. (1991). Explanations in knowledge systems: Design for explainable expert systems. IEEE Expert, 6(3), 58-64.
21. van Lent, M., Fisher, W., & Mancuso, M. (2004). An explainable artificial intelligence system for small-unit tactical behavior. Proceedings of the Nineteenth National Conference on Artificial Intelligence (pp. 900-907). San Jose, CA: AAAI Press.
22. Zahavy, T., Zrihem, N.B., & Mannor, S. (2017). Graying the black box: Understanding DQNs. (arXiv:1602.02658 [cs.LG])
23. Zhang, Y., Sreedharan, S., Kulkarni, A., Chakraborti, T., Zhuo, H.H., & Kambhampati, S. (2017). Plan explicability and predictability for robot task planning. To appear in Proceedings of the International Conference on Robotics and Automation. Singapore: IEEE Press.

[2] See `http://home.earthlink.net/~dwaha/research/meetings/ijcai17-xai/`.

B Detailed Results

Paper	On topic	Data Driven	Validation	Comments
1	1	1	0	
2	0	0	0	
3	1	1	N/A	A position paper, so Validation not applicable.
4	0	0	0	
5	1	0	1	
6	1	1	0	
7	1	0	1	
8	0	2	1	Off topic, but is mature work
9	1	0	N/A	
10	0	0	0	
11	1	?	0	Could not locate reference Jennings et al. (1982)
12	N/A	N/A	N/A	Survey paper on explanation in the social sciences
13	1	0	0	
14	1	0	1	
15	1	0	0	
16	1	0	0	
17	0	0	0	
18	1	0	0	
19	1	2	N/A	Survey paper, so Validation not applicable
20	1	0	0	
21	1	0	1	
22	1	0	0	
23	1	1	0	

Using Explanations to Improve Ensembling of Visual Question Answering Systems

Nazneen Fatema Rajani
Department of Computer Science
University of Texas at Austin
nrajani@cs.utexas.edu

Raymond J. Mooney
Department of Computer Science
University of Texas at Austin
mooney@cs.utexas.edu

Abstract

We present results on using explanations as auxiliary features to improve stacked ensembles for Visual Question Answering (VQA). VQA is a challenging task that requires systems to jointly reason about natural language and vision. We present results applying a recent ensembling approach to VQA, Stacking with Auxiliary Features (SWAF), which learns to combine the results of multiple systems. We propose using features based on explanations to improve SWAF. Using explanations we are able to improve ensembling of three recent VQA systems.

1 Introduction

In recent years, deep-learning has led to unprecedented breakthroughs in many avenues of Artificial Intelligence and most notably in computer vision. Even though the results produced by these deep networks have been groundbreaking, they lack transparency, making them hard to understand and interpret [Lipton, 2016]. Consequently, when such intelligent models that provide no explanation for their decisions fail, it becomes very difficult to do any root cause analysis. Transparency is also important in order to build human trust in systems. Recently, there has been some work by the deep learning community on generating explanations as a way to better understand and interpret the decisions made by deep neural networks [Hendricks *et al.*, 2016; Goyal *et al.*, 2016; Selvaraju *et al.*, 2016].

Visual Question Answering (VQA) is a challenging task that requires systems to attend to regions of an image or question or both for producing an output. VQA addresses open-ended questions about images [Antol *et al.*, 2015] and has attracted significant attention in the past year [Andreas *et al.*, 2016; Goyal *et al.*, 2016; Agrawal *et al.*, 2016]. It requires visual and linguistic comprehension, language grounding as well as commonsense knowledge. A variety of methods to address these challenges have been developed in recent years [Fukui *et al.*, 2016; Xu and Saenko, 2016; Lu *et al.*, 2016; Chen *et al.*, 2015]. The vision component of a typical VQA system extracts visual features using a deep convolutional neural network (CNN), and the linguistic component encodes the question into a semantic vector using a recurrent neural network (RNN). An answer is then generated conditioned on the visual features and the question vector. Some VQA models have an explicit attention component in the architecture [Fukui *et al.*, 2016; Lu *et al.*, 2016], whereas systems that use CNNs without attention [Antol *et al.*, 2015], the gradient for the desired class is backpropagated through the convolutional feature maps to obtain a visualization of the focus regions in an image.

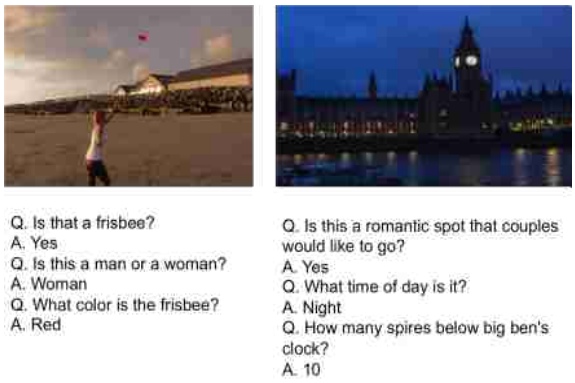


Figure 1: Random sample of images with related three questions and ground truth answers taken from the VQA dataset

Explanation can be helpful in not just understanding and interpreting systems' output but also leveraging systems for improving performance. For example, a system that generates an explanation that is not coherent with its output is not reliable. For VQA, explanation can be of two types – visual or textual. The regions in an image that a model attends to while generating an output can be considered a visual explanation. The words in the question that a model attends to can be considered a textual explanation. Most VQA systems use visual attention, however there are some that use both visual and textual attention while generating an output. The visual explanation is generally represented using heat-maps that use color intensities to highlight the regions in that image that a model attends to. In this paper, we use visual explanation for improving the accuracy of VQA systems.

Most VQA systems have a single underlying method that optimizes a specific loss function and do not leverage the advantage of using multiple diverse models. Ensembling

systems intelligently is crucial to optimizing overall performance. In this paper, we use Stacking with Auxiliary Features (SWAF) [Rajani and Mooney, 2017] to more effectively combine diverse VQA models. The key idea is that we trust systems' agreement on an answer more if they also agree on its explanation. Traditional stacking [Wolpert, 1992] trains a supervised meta-classifier to appropriately combine multiple system outputs. SWAF further enables the stacker to exploit additional relevant knowledge of both the component systems and the problem by providing "auxiliary features" to the meta-classifier. Our key contribution is using visual explanations to create additional useful auxiliary features for SWAF applied to VQA. We demonstrate that ensembling three leading VQA systems using this approach outperforms a variety of baselines and ablations.

2 Background and Related Work

VQA is the task of answering a natural language question about the content of an image by returning an appropriate word or phrase. Figure 1 shows a sample of images and questions from the VQA 2016 challenge. The dataset consists of images taken from the MS COCO dataset [Lin *et al.*, 2014] and three questions per image obtained through Mechanical Turk. Table 1 gives some statistics on the dataset.

	Images	Questions
Training	82,783	248,349
Validation	40,504	121,512
Test	81,43	244,302

Table 1: VQA dataset size

In stacking, a meta-classifier is learned to combine the outputs of multiple underlying systems. The stacker learns a classification boundary based on the confidence scores provided by individual systems for each possible output. Stacking With Auxiliary Features (SWAF) provides the meta-classifier additional information, such as features of the current problem and provenance information for the output from individual systems. We use visual explanation that provides information about regions in an image that are crucial for generating the output. This allows SWAF to *learn* which systems are reliable based on what regions of the image they attend to, on which types of problems and when to trust agreements between specific systems. It has previously been applied effectively to information extraction and entity linking [Viswanathan *et al.*, 2015; Rajani and Mooney, 2016; 2017]. To the best of our knowledge, there has been no prior work on using explanation for improving ensembles. Figure 2 gives an overview of the SWAF approach.

We use SWAF to combine three diverse VQA systems such that the final ensemble performs better than any individual component model even on questions with low agreement. The three component models are trained on the VQA training set and the stacker is trained on the validation data.

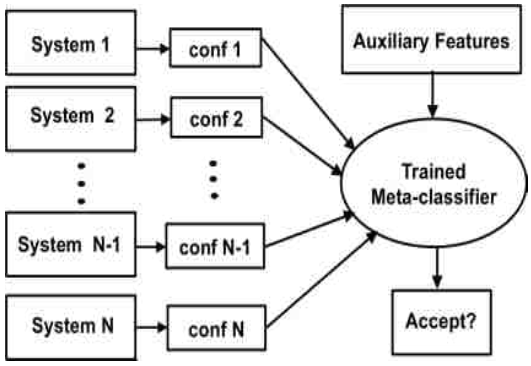


Figure 2: Ensemble Architecture using Stacking with Auxiliary Features. Given an input, the ensemble judges every possible question-answer pair produced by the component systems and determines the final output answer.

2.1 Long Short Term Memory (LSTM)

The LSTM model is one of the original baseline models used to establish a benchmark for the VQA dataset [Antol *et al.*, 2015]. It combines an LSTM [Hochreiter and Schmidhuber, 1997] for the question with a CNN for the image to generate an answer and uses one-hot encoding for the words in the question and the penultimate layer of the VGGNet [Simonyan and Zisserman, 2015] as image features fused together using element-wise multiplication. We note that this model does not have an explicit attention.

2.2 Multimodal Compact Bilinear pooling (MCB)

Traditionally, systems that combine vision and language vector representations use concatenation or element-wise product or sum. [Fukui *et al.*, 2016] argue that such methods are not as effective as an outer product of the visual and textual vectors. To overcome the challenge of high dimensionality due to the outer product, the authors propose using Multimodal Compact Bilinear pooling (MCB) to efficiently and expressively combine multimodal features. The MCB model extracts representations for the image using the 152-layer Residual Network [He *et al.*, 2015] and an LSTM embedding of the question. The two vectors are pooled using MCB and the answer is obtained by treating the problem as a multi-class classification problem with $3,000$ possible answers.

2.3 Hierarchical Question-Image Co-Attention (HieCoAtt)

The idea behind the HieCoAtt model is that in addition to using visual attention to focus on where to look, it is equally important to model what words to attend to in the question (question attention) [Lu *et al.*, 2016]. The model jointly reasons about the visual and language components using "co-attention." Question attention is modeled using a hierarchical architecture including word, phrase, and question levels. HieCoAtt uses two types of co-attention – parallel and sequential at all three levels of the question hierarchy.

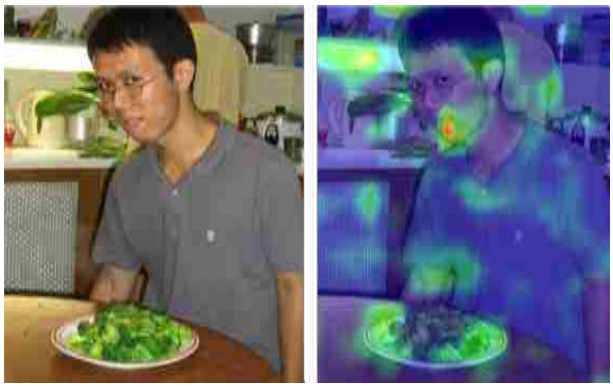


Figure 3: On the left is an image from the VQA dataset and on the right is the heat-map overlaid on the image for the question - 'What is the man eating?'

3 Auxiliary Features for SWAF

For stacking the VQA systems, we first form unique question-answer pairs across all outputs before passing them through the stacker. If a system generates a given I/O pair, then we use its probability estimate for that output, otherwise the confidence is considered zero. If a question-answer pair is classified as correct by the stacker, but there is also another answer that is classified correct for the same question, the pair with higher classifier confidence is chosen. For questions that did not have any answer classified as correct by the stacker, we choose the answer with lowest classifier confidence, which means it is least likely to be incorrect.

The confidence scores along with explanation as auxiliary features are used by the stacker, as shown in Figure 2, to classify each question-answer pair. The auxiliary features are the backbone of the SWAF approach, enabling the stacker to intelligently learn to rely on systems' outputs conditioned on the supporting evidence. Along with explanation, we also use three other types of features that enable the stacker to get more context while making a decision.

3.1 Explanation

Recently, there has been work on analyzing regions of the image that VQA models focus on while answering the question [Goyal *et al.*, 2016]. The authors concluded that deep learning models attend to relevant parts of the image while answering the question. The parts of images that the models focus on can be thought of as visual explanations for answering the question. We use these visual explanations to construct auxiliary features for SWAF.

The part of image to which the model attends can be visualized using a heat-map. Figure 3 shows an image and its heat-map for a given question. The idea is to trust the agreement between systems when they also agree on the heat-map explanation. The heat-map of a given system is compared to every other system's heat-map using using the rank correlation protocol described in [Das *et al.*, 2016]. This generates n choose 2 "explanation agreement" auxiliary features for SWAF. The idea behind using such features is that it enables the stacker to learn to rely on systems that "look" at the right region of the image when generating an answer.

We use the GradCAM algorithm [Goyal *et al.*, 2016] to generate explanatory heat-maps for each answer. Given an image and category, the image is forward propagated through the CNN part of the model. The gradients are set to zero for all categories except the one under consideration, which is set to 1. This signal is then backpropagated to the convolutional feature maps of interest and is combined to compute the heat-map.

3.2 Question and Answer Types

[Antol *et al.*, 2015] analyzed the VQA data and found that most questions fall into several types based on the first few words. For example, questions beginning with "What is...", "Is there...", "How many...", or "Does the...". Using the validation data, we discover such lexical patterns that define a set of question types. The questions were tokenized and a question type was formed by adding one token at a time, up to a maximum of 5, to the current substring. The question "What is the color of the vase?" has the following types "What", "What is", "What is the", "What is the color", "What is the color of". The prefixes that contain at least 500 questions were then retained as types. We added a final type "other" for questions that do not fall into any of the predefined types, resulting in a total of 70 question types. A 70-bit vector is used to encode the question type as a set of auxiliary features.

The original analysis of VQA answers found that they are 38% "yes/no" and 12% numbers. There is clearly a pattern in the VQA answers as well and we use the questions to infer some of these patterns. We considered three answer types – "yes/no", "number" and "other". The answer-type auxiliary features are encoded using a one-hot vector. We classify all questions beginning with "Does","Is","Was","Are", and "Has" as "yes/no". Ones beginning with "How many", "What time", "What number" are assigned "number" type. These inferred answer types are not exhaustive but have good coverage.

3.3 Question Features

We also use a bag-of-words (BOW) representation of the question as auxiliary features. Words that occur at least five or more times in the validation set were included. The final sparse vector of dimension $3,391$ representing a question was normalized by the number of unique words in the question. [Goyal *et al.*, 2016] showed that attending to specific words in the question is important in VQA. Including a BOW in the auxiliary features equips the stacker to efficiently learn which words are important to classifying answers.

3.4 Image Features

Along with the aforementioned features, we also use "deep visual features" of the image as additional auxiliary features. Specifically, we use the $4,096$ features from VGGNet's $fc7$ layer [Simonyan and Zisserman, 2015]. Using such image features enables the stacker to learn to rely on systems that are good at identifying answers for particular types of images.

Method	All	Yes/No	Number	Other
Voting (MCB + HieCoAtt + LSTM)	60.31	80.22	34.92	48.83
iBOWIMG [Zhou *et al.*, 2015]	55.72	76.55	35.03	42.62
DPPNet [Noh *et al.*, 2016]	57.36	80.28	36.92	42.24
LSTM [Antol *et al.*, 2015]	58.20	80.60	36.50	43.70
HieCoAtt [Lu *et al.*, 2016]	61.80	79.70	38.70	51.70
MCB [Fukui *et al.*, 2016]	62.56	80.68	35.59	52.93
Stacking	62.59	81.79	34.58	51.72
+ Q/A types	62.73	82.09	35.47	52.10
+ Question Features	63.12	81.61	36.07	53.77
+ Image Features	65.44	82.08	38.08	57.15
+ Explanation*	**65.54**	**82.28**	**38.63**	**57.32**

Table 2: Accuracy results on the VQA open-ended *test-standard* set (except for the explanation features)

4 Experimental Results

We present experimental results on various baselines and ablations of the Stacking With Auxiliary Features (SWAF) approach. The VQA challenge splits the test set into *test-dev* and *test-standard*. Evaluation on either split requires submitting the output to the competition's online server.† However, there are less restrictions on the number of submissions that can be made to the *test-dev* compared to the *test-standard*. The *test-dev* set is a subset of the standard test set consisting of randomly selected $60,864$ questions.

We note that generating explanations is computationally expensive and we were only able to get results on the test-dev set with the explanation features. All the other results are reported on the entire test set. We use $L1$ regularized SVM classification for generic stacking and stacking with only question/answer types as auxiliary features. For the question, image, and explanation features, we found that a neural network with two hidden layers works best. The first layer is fully connected and the second has approximately half the number of neurons as the first hidden layer. We used Keras with Tensorflow back-end [Chollet, 2015] for implementing the network.

We compare our approach to a voting baseline which maximizes precision by only accepting an answer to be correct if all the component systems predicted the exact same answer for a given question. For questions that do not have a consensus, the answer that has maximum agreement is taken with ties broken in favor of systems with higher confidence. We also compare against two other state-of-the-art VQA systems not used in our ensemble: iBOWIMG [Zhou *et al.*, 2015] and DPPNet [Noh *et al.*, 2016]. iBOWIMG uses softmax over the bag-of-words representation of the question concatenated with GoogleNet [Szegedy *et al.*, 2015] image features and gives comparable performance to models using deep or recurrent neural networks. VGG has lower error rate compared to GoogleNet for CNNs and is thus our choice for image features [Johnson, 2016]. DPPNet uses a CNN with a dynamic parameter layer whose weights are determined adaptively based on questions using a gated recurrent unit (GRU).

The VQA server along with reporting accuracies on the full question set, also reports a break-up of accuracy based on three answer categories. Table 2 shows the full set and category-wise accuracies. Although the results using explanation are on the test-dev subset of the test set and not directly comparable, they do show a small improvement in accuracy. The number of explanation features is small compared to all the other feature types. So to avoid over-fitting to the other features, we also plan on trying neural architectures in which the different feature sets are fused at later layers in the network.

5 Conclusion and Future Work

This paper has proposed and evaluated the novel idea of using explanations to improve ensembling of multiple systems. It has demonstrated how visual explanations for visual question answering (represented as heat-maps) can be used to aid stacking with auxiliary features. This approach effectively utilizes information on the degree to which systems agree on the *explanation* of their answers. We also described three other types of auxiliary features obtained from VQA problems and showed that the combination of all of these auxiliary features, including explanation, gives the best results.

We believe that integrating explanation with ensembling has a two-fold advantage. First, as discussed in this paper, explanations can be used to improve the accuracy of an ensemble. Second, explanations from the component systems can be used to build an explanation for the overall ensemble. That is, by combining multiple component explanations, SWAF could also produce more comprehensible results. Therefore, in the future, we would like to focus on explaining the results of an ensemble. Another issue we plan to explore is using textual explanations for VQA. We believe that the words in the question to which a system attends can also be used to improve ensembling. Finally, we hope to apply our approach to additional problems beyond VQA.

Acknowledgement

This research was supported by the DARPA DEFT program under AFRL grant FA8750-13-2-0026.

* Result obtained on test-dev set

† `http://www.visualqa.org/challenge.html`

References

[Agrawal *et al.*, 2016] Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. Analyzing the behavior of visual question answering models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP-16)*, 2016.

[Andreas *et al.*, 2016] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to compose neural networks for question answering. In *Proceedings of the Conference on Natural language learning (NAACL2016)*, pages 1545–1554, 2016.

[Antol *et al.*, 2015] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[Chen *et al.*, 2015] Kan Chen, Jiang Wang, Liang-Chieh Chen, Haoyuan Gao, Wei Xu, and Ram Nevatia. ABC-CNN: An attention based convolutional neural network for visual question answering. *arXiv preprint arXiv:1511.05960*, 2015.

[Chollet, 2015] Franois Chollet. Keras. `https://github.com/fchollet/keras`, 2015.

[Das *et al.*, 2016] Abhishek Das, Harsh Agrawal, C Lawrence Zitnick, Devi Parikh, and Dhruv Batra. Human Attention in Visual Question Answering: Do Humans and Deep Networks Look at the Same Regions? *arXiv preprint arXiv:1606.03556*, 2016.

[Fukui *et al.*, 2016] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal Compact Bilinear pooling for Visual Question Answering and Visual Grounding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP-16)*, 2016.

[Goyal *et al.*, 2016] Yash Goyal, Akrit Mohapatra, Devi Parikh, and Dhruv Batra. Towards Transparent AI Systems: Interpreting Visual Question Answering Models. In *International Conference on Machine Learning (ICML) Workshop on Visualization for Deep Learning, 2016*, 2016.

[He *et al.*, 2015] K. He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[Hendricks *et al.*, 2016] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating Visual Explanations. *arXiv preprint arXiv:1603.08507*, 2016.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Johnson, 2016] Justin Johnson. cnn-benchmarks. `https://github.com/jcjohnson/cnn-benchmarks`, 2016.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[Lipton, 2016] Zachary C Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

[Lu *et al.*, 2016] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*, pages 289–297, 2016.

[Noh *et al.*, 2016] Hyeonwoo Noh, Paul Hongsuck Seo, and Bohyung Han. Image question answering using convolutional neural network with dynamic parameter prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 30–38, 2016.

[Rajani and Mooney, 2016] Nazneen Fatema Rajani and Raymond J. Mooney. Combining Supervised and Unsupervised Ensembles for Knowledge Base Population. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP-16)*, 2016.

[Rajani and Mooney, 2017] Nazneen Fatema Rajani and Raymond J. Mooney. Stacking With Auxiliary Features. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI2017)*, Melbourne, Australia, August 2017.

[Selvaraju *et al.*, 2016] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *arXiv preprint arXiv:1610.02391*, 2016.

[Simonyan and Zisserman, 2015] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-scale Image Recognition. In *Proceedings of ICLR*, 2015.

[Szegedy *et al.*, 2015] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[Viswanathan *et al.*, 2015] V. Viswanathan, N. Rajani, Y. Bentor, and R. Mooney. Stacked ensembles of information extractors for knowledge-base population. In *Proceedings of ACL 2015*, Beijing, China, 2015.

[Wolpert, 1992] D. Wolpert. Stacked generalization. *Neural Networks*, 5, 1992.

[Xu and Saenko, 2016] Huijuan Xu and Kate Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *European Conference on Computer Vision*, pages 451–466. Springer, 2016.

[Zhou *et al.*, 2015] Bolei Zhou, Yuandong Tian, Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Simple baseline for visual question answering. *arXiv preprint arXiv:1512.02167*, 2015.

A Characterization of Monotone Influence Measures for Data Classification

Jakub Sliwinski
National Univ. of Singapore
dcsjaku@nus.edu.sg

Martin Strobel
Nanyang Technological Univ.
str0001@e.ntu.edu.sg

Yair Zick
National Univ. of Singapore
dcsyaz@nus.edu.sg

Abstract

In this work we focus on the following question: how important was the i-th feature in determining the outcome for a given datapoint? We identify a family of *influence measures*; functions that, given a datapoint $\vec{x}$, assign a value $\phi_i(\vec{x})$ to every feature i, which roughly corresponds to that i's importance in determining the outcome for $\vec{x}$. This family is uniquely derived from a set of axioms: desirable properties that any reasonable influence measure should satisfy. Departing from prior work on influence measures, we assume no knowledge — or access — to the underlying classifier labeling the dataset. In other words, our influence measures are based on the dataset alone, and do not make any queries to the classifier. While this requirement naturally limits the scope of explanations we provide, we show that it is effective on real datasets.

1 Introduction

Alice applied for a bank loan and was denied; knowing that she is in good financial standing, she demands that the bank explains its decision. However, the bank has recently implemented an ML algorithm that filters some applications, and has automatically rejected Alice's. How should the bank explain its decision? This example is more than anectdotal; recent years have seen the widespread implementation of data-driven algorithms making decisions in increasingly high-stakes domains, such as healthcare, transportation and public safety. Using novel ML techniques, algorithms are able to process massive amounts of data and make highly accurate predictions; however, their inherent complexity makes it increasingly difficult for humans to understand *why* certain decisions were made. By obfuscating the underlying decision making processes, such algorithms run the risk of exposing human stakeholders to risks. These risks could include incorrect decisions (e.g. Alice's application was wrongly rejected due to a system bug), information leaks (e.g. the algorithm was inadvertently given information about Alice that it should not have seen), or discrimination (e.g. the algorithm is biased against female applicants). Indeed, government bodies and regulatory authorities have recently begun calling for algorithmic transparency: providing human-interpretable explanations of the underlying reasoning behind large-scale decision making algorithms.

1.1 Our Contribution

In this work, we investigate *influence measures*: these are functions that, given a dataset, assign a value to every feature, roughly corresponding to its importance in affecting the classification outcome for individual datapoints. We identify specific properties that any reasonable influence measure should satisfy (Section 3); next, we mathematically derive a class of influence measures, dubbed *monotone influence measures* (MIM), which uniquely satisfy these axioms (Section 4). Unlike most existing influence measures in the literature, we assume neither knowledge of the underlying decision making algorithm, nor of its behavior on points outside the dataset. Indeed, some methodologies (see Section 1.2) are heavily reliant on having access to counterfactual information: what would the classifier have done if some features were changed? This is a rather strong assumption, as it assumes not only access to the classifier, but also the potential ability to use it on nonsensical data points[1]. By making no such assumptions, we are able to provide a far more general methodology for measuring influence; indeed, many of the tools described in Section 1.2 will simply not be usable when queries to the classifier are not available, or when the underlying classification algorithm is not known. Finally, grounding the measure in the dataset ensures the distribution of data is accounted for, rather than explaining the classification in terms of arbitrarily chosen datapoints. The points can be very unlikely or impossible to occur in practice, and using them can demonstrate a behavior the algorithm will never exhibit in its actual domain. Despite their rather limiting conceptual framework, our influence measures do surprisingly well on real datasets. We show that the outputs of our influence measure are comparable to those of other measures, and provide interpretable results.

1.2 Related Work

Algorithmic transparency has been called for by several government agencies [Hollande, 2016; de Rosnay, 2016; Custers

[1]For example, if the dataset consists of medical records of men and women, the classifier might need to answer how it would handle pregnant men

et al., 2012; Smith *et al.*, 2016a; 2016b]; in addition, recent court rulings have also required the opacity and neutrality of automatic decision systems [Roggensack and Abrahamson, 2016; Blue, 2015; Suzor, 2015; Charruault, 2013]. Last but not least, algorithmic transparency has been widely discussed in the media [Smith, 2016; Citron, 2016; Angwin *et al.*, 2016; Angwin, 2016; Winerip *et al.*, 2016]. The AI and ML community has answered this call. Researchers are designing better explainable AI systems, as well as developing tools to explain the behavior of existing systems; our work is focused on the latter.

Datta *et al.* [2015] axiomatically characterize an influence measure for datasets; however, they interpret influence as a global measure (e.g., what is the overall importance of gender in making decisions); on the other hand, we measure feature importance for individual data-points. Moreover, as Datta *et al.* [2016] show, the measure proposed by Datta *et al.* [2015] outputs undesirable values (e.g. zero influence) in many real instances. Baehrens *et al.* [2010] propose an empirical influence measure that relies on a potential vector like approach. However, as we show, their methodology fails to satisfy our axioms on simple datasets. Other approaches in the literature either rely on black-box access to the classifier [Datta *et al.*, 2016; Ribeiro *et al.*, 2016], or assume domain knowledge (e.g. that the classifier is a neural network whose layers are observable) [Sundararajan *et al.*, 2017].

2 Preliminaries

A dataset $\mathcal{X} = \langle \vec{x}_1, \ldots, \vec{x}_m \rangle$ is given as a list of vectors in $\mathbb{R}^n$ (each dimension $i \in \{1, \ldots, n\}$ is a feature), where for every $\vec{x}_j \in \mathcal{X}$ there is a unique label $c_j \in \{-1, 1\}$; given a vector $\vec{x} \in \mathcal{X}$, we often refer to the label of $\vec{x}$ as $c(\vec{x})$. For example, $\mathcal{X}$ can be a dataset of bank loan applications, with $\vec{x}$ describing the applicant profile (age, gender, credit history etc.), and $c(\vec{x})$ being a binary decision (accepted/rejected). An *influence measure* is simply a function ϕ whose input is a dataset $\mathcal{X}$, the labels of the vectors in $\mathcal{X}$ denoted by c, and a specific point $\vec{x} \in \mathcal{X}$; its output is a value $\phi_i(\vec{x}, \mathcal{X}, c) \in \mathbb{R}$; we often omit the inputs $\mathcal{X}$ and c when they are clear from context. The value $\phi_i(\vec{x})$ should roughly correspond to the importance of the i-th feature in determining the outcome $c(\vec{x})$ for $\vec{x}$.

3 Axioms for Empirical Influence Measurement

We are now ready to define our axioms. We take a geometric interpretation of the dataset $\mathcal{X}$; thus, several of our axioms are phrased in terms of geometric operations on $\mathcal{X}$.

1. **Shift Invariance:** let $\mathcal{X} + \vec{b}$ be the dataset resulting from adding the vector $\vec{b} \in \mathbb{R}^n$ to every vector in $\mathcal{X}$ (not changing the labels). An influence measure ϕ is said to be *shift invariant* if for any vector $\vec{b} \in \mathbb{R}^n$, any $i \in [n]$ and any $\vec{x} \in \mathcal{X}$,

$$\phi_i(\vec{x}, \mathcal{X}) = \phi_i(\vec{x} + \vec{b}, \mathcal{X} + \vec{b}).$$

In other words, shifting the entire dataset by some vector $\vec{b}$ should not affect feature importance.

2. **Rotation and Reflection Faithfulness:** let A be a rotation (or reflection) matrix, i.e. an $n \times n$ matrix with $\det(A) \in \pm 1$; let $A\mathcal{X}$ be the dataset resulting from taking every point $\vec{x}$ in $\mathcal{X}$ and replacing it with $A\vec{x}$. An influence measure ϕ is said to be *faithful to rotation and reflection* if for any rotation matrix A, and any point $\vec{x} \in \mathcal{X}$, we have

$$A\phi(\vec{x}, \mathcal{X}) = \phi(A\vec{x}, A\mathcal{X}).$$

In other words, rotating or reflecting the entire dataset results in the influence vector rotating in the same manner.

3. **Continuity:** an influence measure ϕ is said to be *continuous* if it is a continuous function of $\mathcal{X}$.

4. **Flip Invariance:** let $-c$ be the labeling resulting from replacing every label $c(\vec{x})$ with $-c(\vec{x})$. An influence measure is *flip invariant* if for every point $\vec{x} \in \mathcal{X}$ and every $i \in [n]$ we have

$$\phi_i(\vec{x}, \mathcal{X}, c) = \phi_i(\vec{x}, \mathcal{X}, -c).$$

5. **Monotonicity:** a point $\vec{y} \in \mathbb{R}^n$ is said to *strengthen* the influence of feature i with respect to $\vec{x} \in \mathcal{X}$ if $c(\vec{x}) = c(\vec{y})$ and $y_i > x_i$; similarly, a point $\vec{y} \in \mathbb{R}^n$ is said to *weaken* the influence of i with respect to $\vec{x} \in \mathcal{X}$ if $y_i > x_i$ and $c(\vec{x}) \neq c(\vec{y})$. An influence measure ϕ is said to be *monotonic*, if for any data set $\mathcal{X}$, any feature i and any data point $\vec{x} \in \mathcal{X}$ we have $\phi_i(\vec{x}, \mathcal{X}) \leq \phi_i(\vec{x}, \mathcal{X} \cup \{\vec{y}\})$ whenever $\vec{y}$ strengthens i w.r.t. $\vec{x}$, and $\phi_i(\vec{x}, \mathcal{X}) \geq \phi_i(\vec{x}, \mathcal{X} \cup \{\vec{y}\})$ whenever $\vec{y}$ weakens i w.r.t. $\vec{x}$.

6. **Random Labels:** an influence measure ϕ is said to satisfy the *random labels* axiom, if for any dataset $\mathcal{X}$, if all labels are assigned i.i.d. uniformly at random (i.e. for all $\vec{x} \in \mathcal{X}$, $\Pr[c(\vec{x}) = 1] = \Pr[c(\vec{x}) = -1]$) then for all $\vec{x} \in \mathcal{X}$ and all i we have

$$\mathbb{E}[\phi_i(\vec{x}, \mathcal{X}, c)] = 0.$$

Let us briefly discuss the latter two axioms. Monotonicity is key in defining influence: intuitively, if one is to argue that Alice's old age caused her loan rejection, then finding *older* persons whose loans were similarly rejected should strengthen this argument; however, finding older persons whose loans were not rejected should weaken the argument. The Random Labels axiom states that when labels are randomly generated, no feature should have any influence in expectation; any influence measure that fails this test may assign influence to some features when labels are data independent.

4 Characterization result

In what follows, we show that influence measures satisfying the Axioms in Section 3 must follow a simple formula, described in Theorem 4.2. Below, $\mathbb{1}(p)$ is a $\{1, -1\}$-valued indicator (i.e. 1 if p is true and -1 otherwise), and $\|\vec{x}\|_2$ is the euclidean length of $\vec{x}$.

We begin by showing a simple technical lemma (proof omitted due to space constraints).

Lemma 4.1. *If an influence measure ϕ satisfies both monotonicity and rotation faithfulness, then for any dataset $\mathcal{X}$, any*

datapoint $\vec{x} \in \mathcal{X}$, and any $\vec{y}$ where $\vec{y}$ and $\vec{x}$ differ in some feature, there exists some $a \in \mathbb{R}$ such that

$$\phi(\vec{x}, \mathcal{X} \cup \{\vec{y}\}) - \phi(\vec{x}, \mathcal{X}) = a(\vec{y} - \vec{x}); \quad (1)$$

furthermore, $a \geq 0$ if $c(x) = c(y)$, and $a \leq 0$ otherwise.

Theorem 4.2. *Axioms 1 to 6 are satisfied iff ϕ is of the form*

$$\phi(\vec{x}, \mathcal{X}) = \sum_{\vec{y} \in \mathcal{X} \setminus \vec{x}} (\vec{y} - \vec{x}) \alpha(\|\vec{y} - \vec{x}\|_2) \mathbb{1}(c(\vec{x}) = c(\vec{y})) \quad (2)$$

where α is any non-negative-valued function.

Proof. Suppose ϕ statisfies Axioms 1 to 6. We prove the statement by induction on $k = |\mathcal{X}|$; some technical points are omitted due to space constraints. When the dataset contains a single point (i.e. $k = 1$), the axioms imply that all features have an influence of 0.

When $k = 2$, we have $\mathcal{X} = \langle \vec{x}, \vec{y} \rangle$. If $\vec{x} = \vec{y}$ all features have zero influence. Further, note that any set of two points can be translated by shift and rotation to any other set of two points with the same labels and the same euclidean distance between them. Hence, by shift invariance, rotation faithfulness and Lemma 4.1,

$$\phi(\vec{x}) = \begin{cases} (\vec{y} - \vec{x}) \alpha_1(\|\vec{y} - \vec{x}\|_2) & \text{if } c(\vec{x}) = c(\vec{y}) \\ (\vec{y} - \vec{x}) \alpha_2(\|\vec{y} - \vec{x}\|_2) & \text{if } c(\vec{x}) \neq c(\vec{y}), \end{cases}$$

where α_1 (α_2) is some non-negative (non-positive) valued function. By labels-expectation and flip faithfulness, $\alpha_1 = -\alpha_2$, and then $\phi(\vec{x}, \mathcal{X}) = (\vec{y} - \vec{x}) \alpha(\|\vec{y} - \vec{x}\|_2) \mathbb{1}(c(\vec{x}) = c(\vec{y}))$, where α depends only on $\|\vec{y} - \vec{x}\|_2$.

Suppose the hypothesis holds when $|\mathcal{X}| \leq k$. Consider any dataset $\mathcal{Y}$ of size $k + 1$. The cases where the dataset $\mathcal{Y}$ does not contain at least three different points are handled in a manner similar to when $k = 1, 2$. Suppose $\mathcal{Y}$ contains at least two distinct datapoints $\vec{y}, \vec{z} \neq \vec{x}$. We prove the hypothesis for the case where $\vec{y} - \vec{x}$ and $\vec{z} - \vec{x}$ are linearly independent; the case where they are linearly dependent follows from continuity (we can 'perturb' the points slightly to avoid linear dependency).

By Lemma 4.1 we have

$$\begin{aligned} \phi(\vec{x}, Y) \in A = &\{\phi(\vec{x}, Y \setminus \{\vec{y}\}) + a(\vec{y} - \vec{x}) : a \in \mathbb{R}\} \\ \text{and } \phi(\vec{x}, Y) \in B = &\{\phi(\vec{x}, Y \setminus \{\vec{z}\}) + a(\vec{z} - \vec{x}) : a \in \mathbb{R}\}. \end{aligned}$$

Further by the inductive hypothesis we have:

$$\begin{aligned} \phi(\vec{x}, Y \setminus \{\vec{y}\}) = &\phi(\vec{x}, Y \setminus \{\vec{y}, \vec{z}\}) \\ &+ (\vec{z} - \vec{x}) \alpha(\|\vec{z} - \vec{x}\|_2) \mathbb{1}(c(\vec{x}) = c(\vec{z})) \\ \text{and } \phi(\vec{x}, Y \setminus \{\vec{z}\}) = &\phi(\vec{x}, Y \setminus \{\vec{y}, \vec{z}\}) \\ &+ (\vec{y} - \vec{x}) \alpha(\|\vec{y} - \vec{x}\|_2) \mathbb{1}(c(\vec{x}) = c(\vec{y})). \end{aligned}$$

Hence, since $\vec{y} - \vec{x}$ and $\vec{z} - \vec{x}$ are linearly independent we get,

$$\begin{aligned} \phi(\vec{x}, Y) \in A \cap B = &\{\phi(\vec{x}, Y \setminus \{\vec{y}, \vec{z}\}) \\ &+ (\vec{z} - \vec{x}) \alpha(\|\vec{z} - \vec{x}\|_2) \mathbb{1}(c(\vec{x}) = c(\vec{z})) \\ &+ (\vec{y} - \vec{x}) \alpha(\|\vec{y} - \vec{x}\|_2) \mathbb{1}(c(\vec{x}) = c(\vec{y}))\} \end{aligned}$$

concluding the inductive step. □

5 Axiomatic analysis of existing measures

As mentioned above, several feature influence measures were proposed in prior work. Most of them, however, fundamentally rely on black-box access to the underlying classifier and cannot be immediately applied to our setting; for example, QII [Datta *et al.*, 2016] cannot be easily applied without some heavy modifications. In this section we discuss two popular proposed methods: LIME [Ribeiro *et al.*, 2016] and PARZEN [Baehrens *et al.*, 2010]. These methods can be applied to our setting without departing much from their original definition; moreover, they can be seen as typical examples of two fundamentally different ways of looking at this problem.

5.1 Parzen

The main idea behind the approach followed by Baehrens *et al.* [2010] is to approximate the labeled dataset with a *potential function* and then use the derivative of this function to locally assign influence to features. Given a locality measure σ and a kernel function

$$k_\sigma(\vec{x}) = \frac{1}{\sqrt{\pi \sigma^2}} \exp\left(\frac{\sum_{i=1}^n x_i^2}{2\sigma^2}\right),$$

we can derive the influence measure

Definition 5.1 (Parzen)**.** The *parametric parzen influence measure* $\phi_{\text{Parzen}_\sigma}(\vec{x}, \mathcal{X})$ is given by the derivative at $\vec{x}$ of the potential function

$$\mathbb{P}(c(\vec{x}) = 1 | \vec{x}) = \frac{\sum_{\vec{y} \in \mathcal{X} c(\vec{y}) = 1} k_\sigma(\vec{x} - \vec{y})}{\sum_{\vec{y} \in \mathcal{X}} k_\sigma(\vec{x} - \vec{y})}.$$

It is easy to check that $\phi_{\text{Parzen}_\sigma}$ satisfies Axioms 1 to 4. However, Parzen is neither monotonic, nor can it efficiently detect random labels. To understand why Parzen fails monotonicity it helps to look at the potential function. In Figure 1, we have a single feature ranging from 0 to 2; we are measuring influence for the point $\vec{x}_0$ (marked with a green circle). When we add two more positive labels slightly to its right, the value of $\phi_{\text{Parzen}_\sigma}(\vec{x}_0, \mathcal{X})$ should not decrease; however, this addition 'flattens' the potential function, decreasing the influence of the feature. The violation of the random label axiom can easily be checked on any dataset with two points. The underlying problem is the same: $\phi_{\text{Parzen}_\sigma}$ measures only change in labels, so data points of the same label lead to zero influence and not positive influence. This leads to problems, since $\phi_{\text{Parzen}_\sigma}$ assigns influence to noise, since noise leads to change.

5.2 LIME

The measure developed by Ribeiro *et al.* [2016] has been shown to work well in some instances. Unfortunately, at its' core is a discretization step which makes it unsuitable for an axiomatic analysis. Through the discretization alone it violates almost all axioms. On the other hand, based on the underlying idea of locally approximating the classification with a linear function, one can design an SVM-like measure more fit for theoretical analysis. However, the more adjustments one makes, the more the measure resambles a monotone influence measure, so the motivation for experimental comparison becomes unclear.

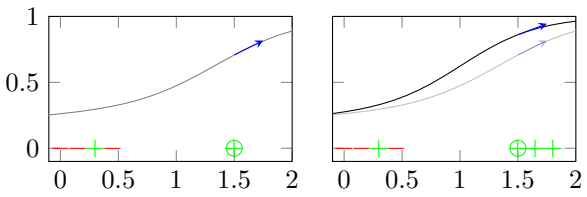


Figure 1: Parzen violates monotonicity; the point of interest $\vec{x}_0$ is marked with a green circle. Its influence is the slope of the blue arrow above it.

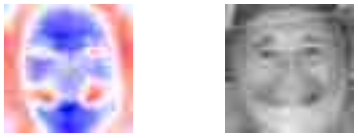
6 Experimental results

The dataset used to produce the experimental results is a part of the Facial Expression Recognition 2013 dataset described in [Goodfellow *et al.*, 2013]. The data consists of 12156 48×48 pixel grayscale images of faces, evenly divided between happy and sad facial expressions. Each pixel is a feature; its brightness level is its parametric value. A parametric Parzen influence measure with $\sigma = 4.7$ and a monotone influence measure with $\alpha(d) = \frac{1}{d^2}$ were run on some of the images.

(a) The point of interest

(b) Influence of pixels, MIM

(c) POI shifted according to MIM influence



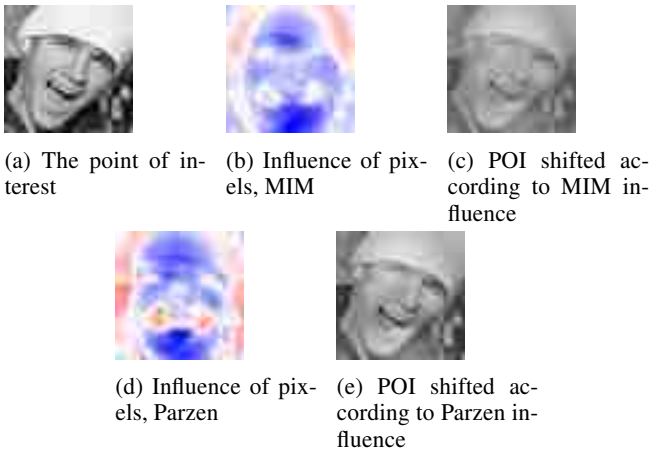
(d) Influence of pixels, Parzen

(e) POI shifted according to Parzen influence

Figure 2: Example point of interest

Figure 2 shows an example picture of a happy face from the dataset, along with a visualization of the influence vertors as produced by MIM and Parzen. In the images of influence vectors, the color blue (red) indicates positive (negative) influence; that is, for every pixel, the measures indicate that the brighter (darker) the pixel in the original image, the more 'happy' ('sad') the face. Subfigures 2c and 2e show the point of interest shifted according to the influence vector, i.e. the pixels with positive influence were brightened, and darkened if their influence was negative.

According to the MIM influence vector, the factors that contribute to this face looking happy, are a bright mouth with darkened corners, bright eyebrows, bright tone of the face, and a darkened background. Shifting the picture along the influence vector seems to make the person in the picture smile wider, and open their mouth slightly. The Parzen vector differs from the MIM vector mainly in that it suggests dark eyes as indicative of the label and does not indicate the eyebrows as strongly.



(a) The point of interest

(b) Influence of pixels, MIM

(c) POI shifted according to MIM influence

(d) Influence of pixels, Parzen

(e) POI shifted according to Parzen influence

Figure 3: Example point of interest

Figure 3 shows another example of a picture from the dataset and its MIM/Parzen influence vectors; however, both measures fail to offer a meaningful explanation. This is likely to be since the face in the image is tilted, unlike the majority of images in the dataset. This is due to the fact that the dataset does not describe the locality of the image well enough; one can expect this to be the case for many images if the dataset is so small (12000) for such a complex feature space ($48 \times 48 = 2304$ features, with each potentially taking 256 different shades of gray). This exemplifies how the influence measures are based only on the dataset provided and indicates it needs to describe the locality of the point of interest reasonably well, if black-box access to the classifier or any domain knowledge cannot be assumed.

7 Conclusions and Future Work

In this paper we present a novel characterization of empirical influence measurement. Axiomatic analysis of influence in data domains is an important research direction, as it allows one to discuss *underlying desirable properties*. QII [Datta *et al.*, 2016] is axiomatically characterized, but LIME and PARZEN are not. We believe that an axiomatic characterization of other measures would help the research community to better understand the benefits and drawbacks of each method.

Monotone influence measures have interesting connections to other domains. One can show that our measures generalize influence measures in empirical game-theoretic domains [Balkanski *et al.*, 2017]; furthermore, our measures are related to mathematical formulations of responsibility and blame, described by Chockler and Halpern [2004]. These connections are encouraging, as they pave the way towards a general theory of causal influence across domains.

Acknowledgements

Sliwinski and Zick are supported by a Singapore MOE Grant #R-252-000-625-133; Zick is also supported by a Singapore NRF Fellowship Grant #R-252-000-643-281.

References

[Angwin *et al.*, 2016] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias theres software used across the country to predict future criminals. and its biased against blacks. *ProPublica*, May 2016.

[Angwin, 2016] J. Angwin. Make algorithms accountable. *New York Times*, August 2016.

[Baehrens *et al.*, 2010] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010.

[Balkanski *et al.*, 2017] E. Balkanski, U. Syed, and S. Vassilvitskii. Statistical cost sharing. *CoRR*, abs/1703.03111, 2017.

[Blue, 2015] The Honourable Justice Blue. Duffy v. Google Inc, 2015. [2015] SASC 170.

[Charruault, 2013] M. Charruault. N° de pourvoi: 12-17591. Cour de cassation, June 2013.

[Chockler and Halpern, 2004] H. Chockler and J. Y. Halpern. Responsibility and blame: A structural-model approach. *Journal of Artificial Intelligence Research*, 22:93–115, 2004.

[Citron, 2016] D. Citron. (Un)fairness of risk scores in criminal sentencing. *Forbes*, July 2016.

[Custers *et al.*, 2012] T. Custers, B.and Calders, B. Schermer, and T. Zarsky. *Discrimination and Privacy in the Information Society: Data Mining and Profiling in Large Databases*, volume 3. Springer Science & Business Media, 2012.

[Datta *et al.*, 2015] A. Datta, A. Datta, A. D. Procaccia, and Y. Zick. Influence in classification via cooperative game theory. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[Datta *et al.*, 2016] A. Datta, S. Sen, and Y. Zick. Algorithmic transparency via quantitative input influence. In *Proceedings of 37th IEEE Symposium on Security and Privacy*, 2016.

[de Rosnay, 2016] M. D. de Rosnay. Algorithmic transparency and platform loyalty or fairness in the french digital republic bill, April 2016. Accessed: 2016-11-28.

[Goodfellow *et al.*, 2013] Ian Goodfellow, Dumitru Erhan, Pierre-Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, Yingbo Zhou, Chetan Ramaiah, Fangxiang Feng, Ruifan Li, Xiaojie Wang, Dimitris Athanasakis, John Shawe-Taylor, Maxim Milakov, John Park, Radu Ionescu, Marius Popescu, Cristian Grozea, James Bergstra, Jingjing Xie, Lukasz Romaszko, Bing Xu, Zhang Chuang, and Yoshua Bengio. Challenges in representation learning: A report on three machine learning contests, 2013.

[Hollande, 2016] F. Hollande. Pour une république numérique (1), October 2016. LOI n 2016-1321 NOR: ECFI1524250L.

[Ribeiro *et al.*, 2016] M. T. Ribeiro, S. Singh, and C. Guestrin. " Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1513–1522. ACM, 2016.

[Roggensack and Abrahamson, 2016] C. J. Roggensack and J. Abrahamson. Wisconsin v. Loomis, 2016. Case No.: 2015AP157 - CR.

[Smith *et al.*, 2016a] M. Smith, D. Patil, and Muoz C. Big data: A report on algorithmic systems, opportunity, and civil rights. White House Report, May 2016.

[Smith *et al.*, 2016b] M. Smith, D. Patil, and Muoz C. Big risks, big opportunities: the intersection of big data and civil rights. *White House Blog*, 2016.

[Smith, 2016] M. Smith. A case is putting the use of data to predict defendants futures on trial. *New York Times*, June 2016.

[Sundararajan *et al.*, 2017] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*, 2017.

[Suzor, 2015] N. Suzor. Google defamation case highlights complex jurisdiction problem. *The Conversation*, October 2015.

[Winerip *et al.*, 2016] M. Winerip, M. Schwirtz, and R. Gebeloff. For blacks facing parole in new york state, signs of a broken system. *New York Times*, December 2016.

Regulatory Mechanisms and Algorithms towards Trust in AI/ML

Eva Thelisson
University of Fribourg, Switzerland
eva.thelisson@unifr.ch

Kirtan Padh
EPFL, Switzerland
kirtan.padh@epfl.ch

L. Elisa Celis
EPFL, Switzerland
elisa.celis@epfl.ch

Abstract

Recent studies suggest that automated processes that are prevalent in machine learning (ML) and artificial intelligence (AI) can propagate and exacerbate systemic biases in society. This has led to calls for regulatory mechanisms and algorithms that are transparent, trustworthy, and fair. However, it remains unclear what form such mechanisms and algorithms can take. In this paper we survey recent formal advances put forth by the EU, and consider what other mechanisms can be put in place in order to avoid discrimination and enhance fairness when it comes to algorithm design and use. We consider this to be an important first step – enacting this vision will require a concerted effort by policy makers, lawyers and computer scientist alike.

1 Introduction

Computer science has developed a wealth of algorithms for increasingly difficult problems, creating efficiency in the world around us, and making the unimaginable possible. Machine learning (ML) and Artificial Intelligence (AI) in particular are projected to yield the highest economic benefits for the United-States, on a worldwide comparison, culminating in a 4.6% growth rate by 2035 [Purdy and Daugherty, 2016]. Using ML/AI, Japan could triple its gross value added growth during the same period, raising it from 0.8% to 2.7%, and Germany, Austria, Sweden and the Netherlands could see their annual economic growth rates double. This is all due to AI/ML's unique ability to drastically improve efficiency by making use of the vast amounts of data currently being generated, collected, and stored in a myriad of business applications. Besides its immense contribution to economic growth, AI/ML has found its place in the daily fabric of our lives, pervading everything from our social interactions (e.g., Facebook) to our news consumption (e.g., Google News and Twitter) to our entertainment (e.g., YouTube and Netflix). Furthermore, decision-making based on algorithms has disseminated to fundamental aspects of everyday life from the finance industry (e.g., credit scoring), to transportation, housing, education, policing, insurance, health, and political systems.

Despite the incredible boon that computational techniques have been to society, certain red flags have recently appeared which demonstrate that algorithms, in particular AI/ML techniques that rely on data, can be biased. A growing number of global leaders and experts including Bill Gates, Elon Musk, Georges Church and Stephen Hawking have publicly voiced their concern regarding the speed and pervasiveness of the developments of AI/ML. In the US, President Obama's administration produced a report which states that "*big data technologies can cause societal harms beyond damages to privacy*" [Executive office of the President *et al.*, 2014]. In particular, it expressed concerns about the possibility that decisions informed by big data could have discriminatory effects, even in the absence of discriminatory intent. The 2017 edition of the World Economics Forum Global Risks Report, which surveyed 745 leaders in business, government, academia and members of the Institute of Risk Management, listed AI as "the emerging technology with the greatest potential for negative consequences over the coming decade".

Many negative instances have now been demonstrated [O'Neil, 2016; Kirkpatrick, 2016; Barocas and Selbst, 2015]. For instance, Google's online advertising system displayed ads for high-income jobs to men much more often than it did to women [Datta *et al.*, 2015], and ads for arrest records were significantly more likely to show up on searches for distinctively black names or a historically black fraternity [Sweeney, 2013]. Recent events have shown that such algorithmic bias is affecting society in a multitude of ways, e.g., exacerbating systemic bias in the racial composition of the American prison population [Angwin *et al.*, 2016], inadvertently promoting extremist ideology [Costello *et al.*, 2016] and affecting the results of elections [Baer, 2016; Bakshy *et al.*, 2015]. Despite these serious concerns, algorithms, at a fundamental level, pervade everything we do. Simply eliminating them is not an option. Hence it is essential to design algorithmic tools and regulatory mechanisms to empower society at large to mitigate any resulting discrimination, inequality and bias.

For AI/ML to remain beneficial, we must build trust in the systems that are transforming our social, political and business environments and are making decisions on our behalf. We consider at the technical aspect of how bias and discrimination can creep into decisions made by AI, often despite the best intentions of the developers of the algorithm, and how can we prevent such negative outcomes. We then outline the

necessary regulatory mechanisms and techniques that must be developed in order to prevent such biases in the future.

2 Algorithmic Bias

One must first understand how such biases occur. Indeed, computers are inherently impartial, and computer scientists and programmers are not malicious. The problem lies at all points in the cycle of collecting, encoding, modeling and optimizing the data.

2.1 Sources of Algorithmic Biases

Input Data

The problem begins with the data that the algorithms build upon, or even the realities of the world itself. Unconscious and systemic biases, rather than intentional choices, account for a large part of the disparate treatment observed in employment, housing, credit, and consumer markets [Pager and Shepherd, 2008]. Such biases can lead to misrepresentation of particular groups in the training data. If the set of examples in the training data do not fairly represent the data on which the algorithm is supposed to run then misrepresented groups could be disadvantaged [Barocas and Selbst, 2014].

Data Vectorization and Cleaning

The raw data must be converted into a digital form (i.e., represented by some kind of *vector*) that an algorithm can use. This process can also introduce biases. This effect is most striking when the training data is labeled manually; the inherent subjectivity in labeling the data can naturally lead to a bias in the dataset. Consider the real life example of St. Georges Hospital in the United Kingdom in where an algorithm for admission decision was developed based on the previous decisions by the admissions committee [Lowry and Macpherson, 1988]. This algorithm simply learned existing biases in the admissions process and resulted in being systematically unfavorable towards minorities.

Model Building

AI/ML algorithms then take as input a subset of vectorized and/or labeled data, and output a model that can take decisions or make predictions. In making these predictions, algorithms can not only propagate biases as discussed above, but in fact amplify them. One potential solution would be to strip away any identifying information that could lead to discrimination, intended or otherwise. However, this could unnecessarily (or undesirably) hamstring the algorithm itself, rendering it useless.

Behavioral Impact

This in turn affects users' actions, feeding back into the real world. For example, it has been hypothesized that increasingly polarized content in search results and online feeds such as Facebook and Twitter can lead to increasingly polarized opinions and behavior [Epstein and Robertson, 2015].

Hence, the steps in the AI/ML life cycle become a destructive feedback loop that can not only propagate, but also exacerbate, societal biases. Thus, if approached without care, algorithms can end up duplicating or even aggravate existing patterns of discrimination that persist in society.

2.2 A Rising Level of Awareness in the EU

On 25 May 2018, the General Data Protection Regulation (GDPR) will be directly applicable in all Member States of the European Union. It brings some substantial changes on data protection and decision making based on algorithms. The GDPR aims at creating a free data flow market in the EU, while making the rules on data protection in the EU consistent, reinforcing data subject's fundamental rights and increasing the liability of companies that control and process such data. Its scope is global (Art. 3, §1). In particular, it reaffirms the data subject's right to explanation and places restrictions on automated decision-making. The GDPR will be applicable in all EU countries and will introduce EU-wide maximum penalties of €20 million or 4% of Global revenue, whichever is greater (Art. 83, Paragraph 5).

Data processors (i.e., entities who process personal data) will now be obliged to comply with data protection requirements which previously only applied to data controllers (i.e., entities who determine why and how personal data are processed). The GDPR will apply regardless whether the processing takes place in the EU or not, and applies processing activities that are related to the offering of goods or services and monitoring their behavior. This regulation gives data subjects the right to access information collected about them, and also requires data processors to ensure data subjects are notified about the data collected (Articles 13 – 15).

It further recognizes that transparency is a key principle. Data must be treated in a transparent manner (Art. 5, §1a)), transparency may occur in the treatment itself (Art. 13, §2 and Art. 14, §2), and the information communicated by the data controller to the data subject must be transparent (Art. 12, §1). The codes of conduct and certification mechanisms must also respect this transparency principle (Art. 40, §2a) and (Art. 42, §3), and transparency also applies to decision-making (Art. 22). Furthermore, this article gives individuals the right to object to decisions made about them purely on the basis of automated processing when those decisions have significant/legal effects. Other provisions in the Regulation gives data subjects the right to obtain information about the existence of an automated decision making system, the logic involved and its significance and envisaged consequences. In addition, the article 22 of the regulation provides the obligation for the data processor to add additional "safeguards for the rights and freedoms of the data subject", when profiling takes place. Although the article does not elaborate what these safeguards are beyond "the right to obtain human intervention", Articles 13 and 14 state that, when profiling takes place, a data subject has the right to "meaningful information about the logic involved".

Towards satisfying various points of this regulation, and more generally ensuring that the worst fears about AI and ML do not come into effect, we propose various types of solutions which must be developed in collaboration between lawyers, policy makers, and computer scientists in order to ensure a fair and balanced society in the presence of algorithms.

3 Proposed Solutions

To begin, we draw a comparison between the regulation of algorithms and regulations ensuring food safety. Consumers must trust the food that producers and distributors provide on the market. The EU General Food Law Regulation establishes basic criteria for whether a food item is safe. If we instead think of data and algorithms instead of food, one could similarly build a system that is meant to guarantee safety to the functioning of algorithms, following the same reasoning as the EU General Food Law Regulation. Figure 1 draws this parallel between the food law regulation and our proposed regulation of algorithms.

Regulation (EC) No. 178/2002 of the European Parliament and of the Council of 28 January 2002 lays down the general principles and requirements of food law, establishing the European Food Safety Authority and laying down procedures in matters of food safety. On a similar basis, we propose that an EU Regulation dedicated to algorithms, accompanied with European Algorithms Safety Authority laying down procedures in matters of algorithms. This could involve establishing codes of conduct (such as the Food Law Practice guidance), developing third party quality control labels (such as organic certification) and establishing transparency by careful regulation and monitoring of data use as it propagates through various algorithms and tools (as is done when tracing food through the food chain).

Lastly, we call on algorithm designers to further push towards developing the technical tools required to detect, prevent, and correct algorithmic and data biases.

3.1 Codes of Conduct

On 27 June 2017, the European Commission fined Google a record-breaking €2.42 billion for antitrust violations pertaining to its shopping search comparison service. It ordered Google to comply with the simple principle of giving equal treatment to rival comparison shopping services and its own service. Competition commissioner Margrethe Vestager said that "Google has given its own comparison shopping service an illegal advantage by abusing its dominance in general internet search. It has promoted its own service, and demoted rival services. It has harmed competition and consumers. That's illegal under EU antitrust rules." In effect, Google systematically gave disproportionately prominent placement to its own shopping service in its search results. As a result, Google's comparison shopping service is much more visible to consumers in Google's search results, whilst rival comparison shopping services are much less visible. This appeared to be the result of an *explicit* code in Google's algorithm whose *intent* was to discriminate against other services.

Burrell identifies between three *barriers* to transparency [Burrell, 2016]: *1)* intentional concealment on the part of corporations or other institutions, *2)* gaps in technical literacy which, for most people, mean that having access to underlying code is insufficient, and *3)* a lack of interpretability of the decisions made by the algorithm even to experts. For barrier 1, clear codes of conduct that are enforceable, as demonstrated in the example with Google above, is a crucial first step.

3.2 Quality Labels and Audits

To increase transparency, one possibility could be to open the code to public scrutiny. The main drawback to this approach would be the harm it could cause to the valuable intellectual property exposed, and barriers 2 and 3, which state that, even if made public, the results would not be interpretable. As [Lisboa, 2013] notes, "machine learning approaches are alone in the spectrum in their lack of interpretability". Hence, we instead propose that quality labels – similar, e.g., to organic certification, Minergie label, quality management systems and insurance certification (9001 ISO norms), IT security certification (ISO 27 001 norms or Information Technology Infrastructure Library) be made available on a voluntary basis.

The GDPR allows the data controller or processor to draft approved codes of conduct or get a certification on data protection to demonstrate the fulfillment of its duties. The codes of conduct will be approved by the competent authority. The monitoring of compliance with a code of conduct pursuant to Article 40 of GDPR may be carried out by a body which has an appropriate level of expertise in relation to the subject-matter of the code, and is accredited for that purpose by the competent supervisory authority.

The certification can be done by a limited number of certification bodies (Art. 43 GDPR) or by the competent supervisory authority, on the basis of criteria approved by that competent supervisory authority pursuant to Art. 58, §3 GDPR or by the Board (Art. 63 GDPR). Where the criteria are approved by the Board, this may result in a common certification - the European Data Protection Seal. Certification may be issued for a maximal period of three years (renewable). The Board shall collate all certification mechanisms and data protection seals and marks in a register and shall make them publicly available by any appropriate means (Art. 42 GDPR).

The GDPR empowers the regulator to conduct audits and inspections of companies on demand. Strict new compliance requirements are imposed. For example, entities have to perform "Privacy Impact Assessments" and privacy audits as a matter of course. They have to implement "Privacy by Design" methodologies into their business, so that compliance is baked-in to everything they do. They also have to deliver on a new "Accountability" obligation, which means creating written compliance plans, which they will have to deliver to regulators on demand.

3.3 Transparency in the Data Chain

Algorithms must be designed so that a human can interpret the outcome [Goodman and Flaxman, 2016]. However, there is a trade-off between the representation and interpretation of algorithms. Simpler models are easier to explain, but also fail to capture complex interactions among many variables. This also happens to be one of the biggest issues with neural networks, because while they give excellent results in practice, we have very sparse theoretical understanding for them and therefore they are almost completely uninterpretable.

Making reference to the GDPR, [Goodman and Flaxman, 2016] highlighted that "while this law will pose large challenges for industry, it highlights opportunities for computer

scientists to take the lead in designing algorithms and evaluation frameworks which avoid discrimination and enable explanation".

The notion of a "right to explanation" [Goodman and Flaxman, 2016] for an automated decision is correlated to the right to obtain an "explanation of system's functionality". Meaningful information must be provided about the logic involved as well as the significance and the envisaged consequences of such a processing to the data subject (under Articles 15.1.h and 14.2.g). Appropriate safeguards should include the ability of data subjects "to obtain an explanation of the decision reached after such assessment" (recital 71).

Data controllers will have to provide satisfactory explanations for specific automated decisions, i.e., they will have to give the reason why the AI/ML model gives the outputs it does. This will be especially difficult for AI/ML systems, whose outcome may vary from one test to another even if the attributes remain the same. Providing transparency to machine learning systems and black boxes will be a significant technical challenge. Transparency about the personal attributes used by the organizations may allow the data subject to use the decision tree [Rivest, 1987] to follow its logic and gain meaningful information about its significance and the envisaged consequences of such a processing [Wachter *et al.*, 2017]. The data subject could work out what decisions the model would recommend based on a variety of different values for the attributes it considers. Transparency about the logic and likely effects of the automated decision-making system given the person's personal circumstances, transparency about the values used by the algorithm and how it was trained should be guaranteed. Log files may help bringing those guarantees.

We propose to create a data chain traceability, based on the same pattern as the food chain cycle (see Figure 1).

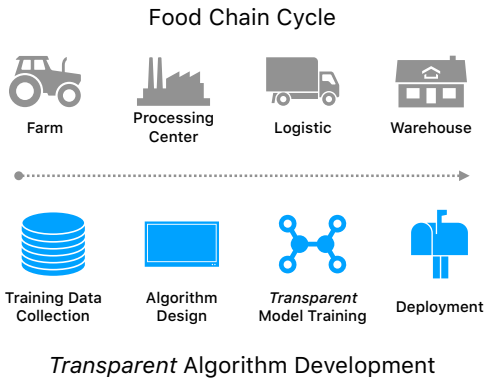


Figure 1: This figure illustrates the symmetries between the food chain cycle and the *transparent* algorithm development. Different regulations and codes of conducts can be devised for each of the steps in algorithm development to ensure overall transparency.

3.4 De-biasing Datasets and Algorithms

According to [Žliobaitė, 2017], "Discrimination-aware data mining studies how to make predictive models free from discrimination, when historical data, on which they are built, may be biased, incomplete, or even contain past discriminatory decisions". There are two main parts to discrimination-aware machine learning, namely discrimination detection and discrimination prevention. Discrimination detection involves finding discriminatory patterns in the training data. Discrimination prevention, on the other hand, entails the development of algorithms which are free from discrimination even on datasets on which standard AI models may discriminate.

The traditional approach to discrimination detection is to fit a regression model to the training data and look at the regression coefficients of the potentially discriminating features such as race, gender etc. The magnitude and the statistical significance of these coefficients can tell us about the possibility of discrimination in the dataset. Discrimination prevention on the other hand can be applied in one of the following three stages of the data processing pipeline according to [Žliobaitė, 2017]: *a)* data preprocessing, *b)* model post-processing, and *c)* model regularization. Data preprocessing is when the training data is preprocessed to remove the discrimination from it and then standard AI models are used for prediction on the cleaned data. Model post-processing starts with standard model and modifies it to incorporate the non-discrimination condition in it. And model regularization adds some constraints to the optimization problem to ensure non-discrimination.

Discrimination-aware machine learning is still in its nascent stage of research and much more needs to be done before it can be incorporated as part of the law.

4 Conclusion

As the new economic business models worldwide are based on data mining and algorithms, a balance has to be found between encouraging innovation with a flexible regulation while protecting the fundamental rights and freedom of people. In the EU, the Charter of Fundamental Rights became legally binding on the European Union in December of 2009, with the entry into force of the Treaty of Lisbon. The Charter contains rights and freedoms under six titles: Dignity, Freedoms, Equality, Solidarity, Citizens' Rights, and Justice.

Building AI Safeguards in order to ensure the respect of those fundamental rights as well as a proper, safe, and reliable functioning of algorithms must be a priority. These safeguards should consider designing accountable algorithms in a way that ensures that ethical principles are encoded in the algorithms. Transparency and trust of algorithms is of key importance to ensure the equal treatment among people and the adequate functioning of a true democratic system.

In this paper we surveyed recent formal advances, and consider what other mechanisms should be put in place. We consider this to be an important first step – enacting this vision will require a concerted effort by policy makers, lawyers and computer scientist alike.

References

[Angwin *et al.*, 2016] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks. *ProPublica, May*, 23, 2016.

[Baer, 2016] Drake Baer. The 'Filter Bubble' Explains Why Trump Won and You Didn't See It Coming, November 2016. NY Mag.

[Bakshy *et al.*, 2015] Eytan Bakshy, Solomon Messing, and Lada A Adamic. Exposure to ideologically diverse news and opinion on facebook. *Science*, 348(6239):1130–1132, 2015.

[Barocas and Selbst, 2014] Solon Barocas and Andrew D. Selbst. Big Data's Disparate Impact. *SSRN eLibrary*, 2014.

[Barocas and Selbst, 2015] S. Barocas and A.D. Selbst. *Big Data's Disparate Impact*. SSRN eLibrary, 2015.

[Burrell, 2016] Jenna Burrell. How the machine thinks: Understanding opacity in machine learning algorithms. *Big Data & Society*, 3(1):2053951715622512, 2016.

[Costello *et al.*, 2016] Matthew Costello, James Hawdon, Thomas Ratliff, and Tyler Grantham. Who views online extremism? individual attributes leading to exposure. *Computers in Human Behavior*, 63:311–320, 2016.

[Datta *et al.*, 2015] Amit Datta, Michael Carl Tschantz, and Anupam Datta. Automated experiments on ad privacy settings. *Proceedings on Privacy Enhancing Technologies*, 2015(1):92–112, 2015.

[Epstein and Robertson, 2015] Robert Epstein and Ronald E Robertson. The search engine manipulation effect (seme) and its possible impact on the outcomes of elections. *Proceedings of the National Academy of Sciences*, 112(33):E4512–E4521, 2015.

[Executive office of the President *et al.*, 2014] United States Executive office of the President, John Podesta, Penny Pritzker, Ernest J. Moniz, John Holdren, and Zients Jeffrey. *Big data: Seizing opportunities, preserving values*. White House, 2014.

[Goodman and Flaxman, 2016] Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a" right to explanation". *arXiv preprint arXiv:1606.08813*, 2016.

[Kirkpatrick, 2016] Keith Kirkpatrick. Battling algorithmic bias: how do we ensure algorithms treat us fairly? *Communications of the ACM*, 59(10):16–17, 2016.

[Lisboa, 2013] Paulo JG Lisboa. Interpretability in machine learning–principles and practice. In *International Workshop on Fuzzy Logic and Applications*, pages 15–21. Springer, 2013.

[Lowry and Macpherson, 1988] Stella Lowry and Gordon Macpherson. A blot on the profession. *British medical journal (Clinical research ed.)*, 296(6623):657, 1988.

[O'Neil, 2016] Cathy O'Neil. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown/Archetype, 2016.

[Pager and Shepherd, 2008] Devah Pager and Hana Shepherd. The sociology of discrimination: Racial discrimination in employment, housing, credit, and consumer markets. *Annual review of sociology*, 34:181, 2008.

[Purdy and Daugherty, 2016] Mike Purdy and Paul Daugherty. Why artificial intelligence is the future of growth. *Accenture, September*, 28, 2016.

[Rivest, 1987] Ronald L Rivest. Learning decision lists. *Machine learning*, 2(3):229–246, 1987.

[Sweeney, 2013] Latanya Sweeney. Discrimination in online ad delivery. *Queue*, 11(3):10, 2013.

[Wachter *et al.*, 2017] Sandra Wachter, Brent Mittelstadt, and Luciano Floridi. Why a right to explanation of automated decision-making does not exist in the general data protection regulation. *International Data Privacy Law*, 2017.

[Žliobaitė, 2017] Indrė Žliobaitė. Measuring discrimination in algorithmic decision making. *Data Mining and Knowledge Discovery*, 31(4):1060–1089, Jul 2017.

Unsupervised Neural-Symbolic Integration

Son N. Tran
The Australian E-Health Research Center, CSIRO
son.tran@csiro.au

Abstract

Symbolic has been long considered as a language of human intelligence while neural networks have advantages of robust computation and dealing with noisy data. Integration of neural-symbolic can offer better learning and reasoning while providing a means for interpretability through the representation of symbolic knowledge. Although previous works focus intensively on supervised feedforward neural networks, little has been done for the unsupervised counterparts. In this paper we show how to integrate symbolic knowledge into unsupervised neural networks. We exemplify our approach with knowledge in different forms, including propositional logic for DNA promoter prediction and first-order logic for understanding family relationship.

1 Introduction

An interesting topic in AI is integration of symbolic and neural networks, two different information processing paradigms. While the former is the key of higher level of intelligence the latter is well known for the capability of effective learning from data. In the last two decades, researchers have been working on the idea that combination of neural networks and symbolic representation of knowledge should offer joint benefits [Towell and Shavlik, 1994; Smolensky, 1995; Avila Garcez and Zaverucha, 1999; Valiant, 2006; Garcez *et al.*, 2008; Penning *et al.*, 2011; França *et al.*, 2014; Tran and Garcez, 2016].

In previous work, supervised neural networks have been used intensively for the integration based on the analogy of *modus ponens* inference with symbolic rules and forward passing in neural networks [Towell and Shavlik, 1994; Avila Garcez and Zaverucha, 1999]. In such networks, due to the discriminative structures only a subset of variables can be inferred, i.e. the variables in the left hand of *if-then* $\leftarrow$ formulas. This may limit their use in general reasoning. Unsupervised network, on the other hand, offers more flexible inference mechanism which seems more suitable for symbolic reasoning. Let us consider an XOR example $\mathrm{z} \leftrightarrow (\mathrm{x} \oplus \mathrm{y})$. Here, given the truth values of any two variables one can infer the rest. For supervised networks, a class variable must be discriminated from the others and only it can be inferred. An unsupervised network, in contrast, do not require such discrimination.

Encoding symbolic knowledge in an unsupervised neural network needs a mechanism to convert symbolic formulas to the network without loss of generality. In previous work, Penalty logic shows that any propositional formula can be represented in a symmetric connectionist network (SCN) where inference with rules is equivalent to minimising the network's energy [Pinkas, 1995]. However, SCN uses dense connections of hidden and visible units which make the inference very computational. Recent work shows that any propositional formula can be represented in restricted Boltzmann machines(RBMs) [Tran, 2017]. Different from Penalty logic, here the RBM is a simplified version of SCN where there is no visible-visible and hidden-hidden connections. This makes inference in RBMs is easier.

Several attempts have been made recently to integrate symbolic representation and RBMs [Penning *et al.*, 2011; Tran and Garcez, 2016]. Despite achieving good practical results they are still heuristic. In this paper, we show how to encode symbolic knowledge in both propositional and first-order forms into the RBM by extending the theory in [Tran, 2017].

The remainder of this paper is organized as follows. Section 2 reviews the idea of Confidence rule, a knowledge form to represent symbolic formulas in RBMs. In section 3 we show how to encode knowledge into RBMs. Section 4 presents the empirical verification of our encoding approach and Section 5 concludes the work.

2 Confidence Rules: Revisit

A confidence rule [Tran and d'Avila Garcez, 2013; Tran and Garcez, 2016] is a propositional formula in the form:

$$c : \mathrm{h} \leftrightarrow \bigwedge_t \mathrm{x}_t \wedge \bigwedge_k \neg \mathrm{x}_k \tag{1}$$

where h is called *hypothesis*, c is a non-negative real value called *confidence value*. Inference with a confidence rule is to find the model that makes the hypothesis h holds. If there exist a target variable y the inference of such variable will be similar to *modus ponens*, as shown in Table 1

An interesting feature of Confidence rules is that one can represent them in an RBM where Gibbs sampling can be seen

Confidence rule inference	Modus ponens
$\mathrm{h} \leftrightarrow \bigwedge_{t\in T} \mathrm{x}_t \wedge \bigwedge_{k\in K} \neg \mathrm{x}_k \wedge \mathrm{y}$	$\mathrm{y} \leftarrow \bigwedge_t \mathrm{x}_t \wedge \bigwedge_k \neg \mathrm{x}_k$
$\{\mathrm{x}_t, \neg \mathrm{x}_k \mid \text{for } \forall t \in T, \forall k \in K\}$	$\{\mathrm{x}_t, \neg \mathrm{x}_k \mid \text{for } \forall t \in T, \forall k \in K\}$
y	y

Table 1: Confidence rule and Modus ponens

equivalently as maximising the total (weighted) satisfiability [Tran, 2017]. If a knowledge base is converted into Confidence rules then we can take the advantage of the computation mechanism in such neural networks for efficient inference. The equivalence between confidence rules and an RBM is defined in that the satisfiability of a formula is inversely proportional to the energy of a network:

$$s_\varphi(\mathbf{x}) = -aE_{rank}(\mathbf{x}) + b$$

where s_φ is the truth value of the formula φ given an assignment $\mathbf{x}$; $E_{rank}(\mathbf{x}) = min_{\mathbf{h}} E(\mathbf{x}, \mathbf{h})$ is the energy function minimised over all hidden variables; $a > 0$,b are scalars.

By using disjunctive normal form (DNF) to present knowledge Confidence rules attract some criticism for practicality since it is more popular to convert a formula to a conjunctive normal form of polynomial size. However, we will show that Confidence rules are still very useful in practice. In fact, in such tasks as knowledge extraction, transfer, and integration Confidence rules have been already employed [Penning *et al.*, 2011; Tran and d'Avila Garcez, 2013; Tran and Garcez, 2016]. For knowledge integration previous work separates the *if-and-only-if* symbol in Confidence rules into two *if-then* rules to encode in a hierarchical network [Tran and Garcez, 2016]. In this work, we show that such separation is not necessary since any propositional *if-then* formulas can be efficiently converted to Confidence rules. The details are in the next section.

3 Knowledge Encoding

In many cases background knowledge presents a set of *if-then* formulas (or equivalent Horn clauses). This section shows how to convert them into Confidence rules for both propositional and first-order logic forms

3.1 Proposition Logic

A propositional *if-then* formula has the form

$$c : \mathrm{y} \leftarrow \bigwedge_t \mathrm{x}_t \wedge \bigwedge_k \neg \mathrm{x}_k$$

which can be transformed to a DNF as:

$$c : (\mathrm{y} \wedge \bigwedge_t \mathrm{x}_t \wedge \bigwedge_k \neg \mathrm{x}_k) \vee \bigvee_t (\neg \mathrm{x}_t) \vee \bigvee_k (\mathrm{x}_k)$$

and then to the confidence rules:

$$c : \mathrm{h}_y \leftrightarrow \mathrm{y} \wedge \bigwedge_t \mathrm{x}_t \wedge \bigwedge_k \neg \mathrm{x}_k$$
$$c : \mathrm{h}_t \leftrightarrow \neg \mathrm{x}_t \text{ for } \forall t$$
$$c : \mathrm{h}_k \leftrightarrow \mathrm{x}_k \text{ for } \forall k$$

Encoding these rules into an RBM does not guarantee the equivalence. This is because it violates the condition that the DNF of a formula should *have at most one conjunct is true given an assignment* [Tran, 2017]. Fortunately this can be solved by grouping $\neg \mathrm{x}_t$, x_k with a max-pooling hidden unit which results in an RBM with the energy function as[1][2]:

$$\begin{aligned} E = &- c \times h_y(y + \sum_t x_t - \sum_k x_k - |T| - 1 + \epsilon) \\ &- c \times h_p \max(\{-x_t + \epsilon, x_k - 1 + \epsilon | t \in T, k \in K\}) \end{aligned} \quad (2)$$

Here a max pooling hidden unit represents a hypothesis: $\mathrm{h}_p \leftrightarrow \bigvee_t \mathrm{h}_t \vee \bigvee_k \mathrm{h}_k$ which, in this case, can be written as: $c : \mathrm{h}_p \leftrightarrow \bigvee_t \neg \mathrm{x}_t \vee \bigvee_k \mathrm{x}_k$. The final set rules are:

$$c : \mathrm{h}_y \leftrightarrow \mathrm{y} \wedge \bigwedge_t \mathrm{x}_t \wedge \bigwedge_k \neg \mathrm{x}_k$$
$$c : \mathrm{h}_p \leftrightarrow \bigvee_t \neg \mathrm{x}_t \vee \bigvee_k \mathrm{x}_k$$

Example 1. *Let us consider the formula:* $5 : \mathrm{y} \leftarrow \mathrm{x}_1 \wedge \neg \mathrm{x}_2$ *which would be converted to DNF as:* $5 : (\mathrm{y} \wedge \mathrm{x}_1 \wedge \neg \mathrm{x}_2) \vee (\neg \mathrm{x}_1) \vee (\mathrm{x}_2)$*, and then to an RBM with the energy function:* $E = -5h_1(y + x_1 - x_2 - 1.5) - 5h_2 \max(-x_1 + 0.5, x_2 - 0.5)$. *Table 2 shows the equivalence between the RBM and the formula.*

x_1	x_2	y	s_φ	E_{rank}
0	0	0	1	-2.5
0	0	1	1	-2.5
0	1	0	1	-2.5
0	1	1	1	-2.5
1	0	0	0	0
1	0	1	1	-2.5
1	1	0	1	-2.5
1	1	1	1	-2.5

Table 2: Energy of the RBM and truth values of the formula $5 : \mathrm{y} \leftarrow \mathrm{x}_1 \wedge \neg \mathrm{x}_2$

where s_φ is (unweighted) truth values of the formula. This indicates the equivalence between the RBM and the formula as:

$$s_\varphi(\mathrm{x}_1, \mathrm{x}_2, \mathrm{y}) = -\frac{1}{2.5} E_{rank}(x_1, x_2, y) + 0$$

3.2 First-order Logic

A first order logic formula can also be converted into a set of Confidence rules. First, let us consider a predicate: $P(x, y)$ which one can present in a propositional DNF as:

$$\bigvee_{a,b|P(a,b)=true} \mathrm{p}_{x=a} \wedge \mathrm{p}_{y=b} \wedge \mathrm{p}_P$$

where (a, b) are the models of $P(x, y)$; $\mathrm{p}_{x=a}$ and $\mathrm{p}_{y=b}$ are the propositions that are $true$ if $x = a$ and $y = b$ respectively, otherwise they are $false$; p_P is the proposition indicating if

[1]The proof is similar as in [Tran, 2017]

[2]$0 < \epsilon < 1$

the value of $P(a, b)$. Each conjunct in this DNF then can be represented as a Confidence rule.

Now, let us consider a first-order formula which we are also able to present in a set of Confidence rules. For example, a clause φ as:

$$\forall_{x,y,z} \text{son}(x, z) \leftarrow \text{brother}(x, y) \wedge \text{has\_father}(y, z)$$

can be converted into:

$$\bigvee_{a,b,c|\varphi=true} (\text{p}_{x=a} \wedge \text{p}_{y=b} \wedge \text{p}_{z=c} \wedge \text{p}_{\text{son}} \wedge \text{p}_{\text{brother}} \wedge \text{p}_{\text{has\_father}}) \vee (\neg\text{p}_{x=a} \vee \neg\text{p}_{y=b} \vee \neg\text{p}_{z=c} \vee \neg\text{p}_{\text{brother}} \vee \neg\text{p}_{\text{has\_father}})$$

If one want to encode the background knowledge through its samples, for example:

$$\text{son}(James, Andrew) \leftarrow \text{brother}(James, Jen) \wedge \text{has\_father}(Jen, Andrew)$$

then we can convert it into confidence rules:

$c : \text{h}_1 \leftrightarrow \text{james} \wedge \text{jen} \wedge \text{andrew} \wedge$ **son** $\wedge$ **brother** $\wedge$ **has_father**

$c : \text{h}_p \leftrightarrow \neg\text{james} \vee \neg\text{jen} \vee \neg\text{andrew} \vee \neg$**brother** $\vee \neg$**has_father**

In practice, in many cases we are only interested in inferring the predicates therefore we can omit $\neg$james, $\neg$jen, $\neg$andrew from the second rule.

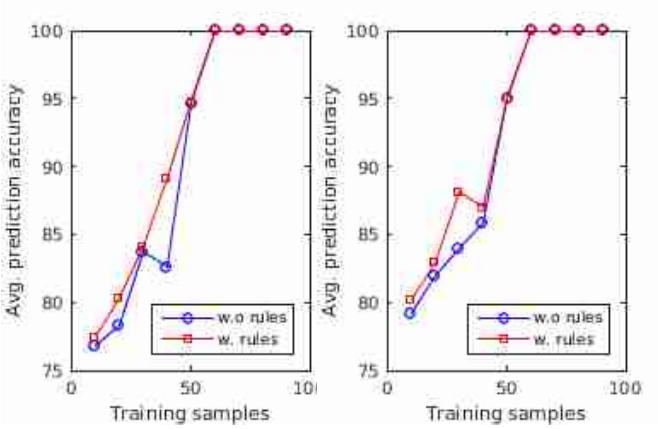
4 Empirical Evaluation

In this section we apply the encoding approaches discussed in the previous section to integrate knowledge into unsupervised networks.

4.1 DNA promoter

The DNA promoter dataset consist of a background theory with 14 logical *if-then* rules [Towell and Shavlik, 1994]. The rules includes four symbols $contact$, $minus_{10}$, $minus_{35}$, $conformation$ which are not observed in the data. This is suitable for hierarchical models as shown in previous works [Towell and Shavlik, 1994; Tran and Garcez, 2016]. In this experiment we group the rules using *hypothetical syllogism* to eliminate the unseen symbols. After that we encode the rules in an RBM following the theory in Section 3.1. The confidence values are selected empirically.

We test the normal RBMs and the RBMs with encoded rule using leave-one-out method, both achieve 100% accuracy. In order to evaluate the effectiveness of our approach we partition the data into nine different training-test sets with number of training samples are 10, 20, 30, 40, 50, 60, 70, 80, 90. All experiments are repeated 50 times and the average results are reported in Figure 1. We perform the prediction using both Gibbs sampling and conditional distribution $P(y|\mathbf{x})$. In particular, Figure 1a shows the prediction results using 1-step Gibbs sampling where the input is fixed to infer the hidden states and then to infer the label unit. In Figure 1b the results show the prediction accuracy achieved by inferring the label unit from the conditional distribution. As we can see, in both cases the integrated RBMs perform better than the normal RBMs on small training sets with number of training sample is less than 60. With larger training sets, the rules are no longer advantageous to the learning since the training samples are adequate to generalise the model to achieve 100% accuracy.



(a) Infer with Gibbs sampling (b) Infer with $P(y|\mathbf{x})$

Figure 1: RBMs without rules v.s RBMs with rules

4.2 Kinship

In this experiment, we use the approach discussed in Section 3.2 for relation discovery and reasoning tasks with Kinship dataset [Hinton, 1986; Sutskever and Hinton, 2008]. Here given a set of examples about relations we perform two type of reasoning: (1) what is relation between two people, i.e $?(x, y)$; and (2) a person has a relation R with whom, i.e. $R(x, ?)$. Previous approaches are using matrices/tensors to represents the relations making it difficult to explain [Sutskever and Hinton, 2008]. In this work, since only predicates are given, we encode the examples for the predicates in an unsupervised network as shown earlier in Section 3.2. This constructs the left part of the integrated model in Figure 2. In the right part, we model the unknown clauses by using a set of hidden units. The idea here is that by inferring the predicates using the encoded rules in the left part we can capture the relationship information, from which the desired relation is inferred by reconstruction of such relationship in the right part. In this experiment, we use auto-encoder [Bengio, 2009] for the right part for the purpose of efficient learning. The whole process is described in Algorithm 1.

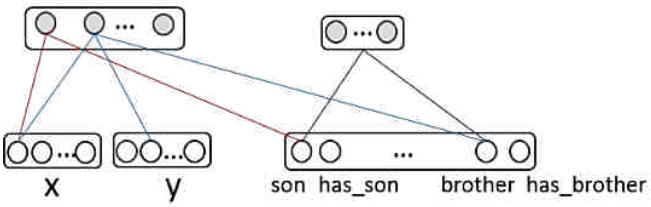


Figure 2: Encoding Kinship examples

Let us take an example where one wants to find the relation between two people $R(Marco, Pierro) =?$. First, we use the other examples to construct an integrated model as in Figure 2. After that, we train the auto-encoder in the right part using unsupervised learning algorithm, then we extract the relation features from Marco and Pierro as shown in Ta-

Marco, Pierro	1.000: has_father	Marco has father who is Pierro
Marco	0.191:has_wife	Marco has wife
	0.191:husband	Marco is a husband of someone
	0.191:has_mother	Marco has mother
	0.191:father	Marco is a father of someone
	0.191:has_daughter	Marco has daughter
	0.191:son	Marco has is a son of someone
	0.191:has_son	Marco has son
	0.191:has_sister	Marco has sister
	0.191:brother	Marco is a brother of someone
Pierro	0.191: has_wife	Pierro has wife
	0.191: husband	Pierro is a husband of someone
	0.191: father	Pierro is a father of someone
	0.191: has_daughter	Pierro has daughter
Reconstruct (possible relations)	0.008:wife 0.008:husband 0.001:mother 0.016:father 0.045:daughter **0.252:son** 0.005:sister 0.013:brother 0.001:aunt 0.002:uncle 0.003:niece 0.002:nephew	

Table 3: Relation features

ble 3. In that table we also show the reconstructed scores for all the relations where son is the correct one.

Algorithm 1

Data: Examples: E, Question: R(a,b)
Result: R
Encode all examples in an RBM: N
Initialise $\mathcal{D} = \emptyset$
for *each example* $R(x, y)$ *in E* **do**
  f = INFER(N,x,y)
  Add f to $\mathcal{D}$
end
Train an Auto-Encoder (AE) on $\mathcal{D}$
f = INFER(N,a,b)
Reconstruct $\hat{f}$ using AE
Return R = $\arg\max_R(\hat{f}_R)$ ▷ Return the unseen relation where the reconstruction feature have the highest value.

1: **function** INFER(N, a, b)
2: Infer direct relation between a,b
3: Infer possible relations of a: R(a,*)
4: Infer possible relations of b: R(*,b)
5: f = concatenation of all relations
6: Return f
7: **end function**

We test the model on answering the question $R(x, y) =?$ using leave-one-out validation which achieve 100% accuracy. We also use the integrated model to reason about whom one has a relation with. This question may have more than one answer, for example $\mathrm{son}(Athur, ?)$ can be either $Cristopher$ or $Penelope$. We randomly select 10 examples for testing and repeat it for 5 times. If the designate answers are in the top relations with highest reconstructed features then we consider this as correct, otherwise we set it as wrong. The average error of this test is 0%. However, when we increase the number of test samples to 20 and 30 the average errors grow to 2.8% and 6.8% respectively. For comparison, the matrices based approach such as [Sutskever and Hinton, 2008] achieves 0.4%,1.2%,2.0% average error rates for 10, 20, 30 test examples respectively. Note that, such approach and many others [Socher *et al.*, 2013] model each relation by a matrix/tensor while in this experiment we share the parameters across all relations. Also, the others use discriminative learning while we use unsupervised learning. The purpose of this is to exemplify the encoding technique we proposed earlier in this paper. Improvement can be achieved if similar methods are employed.

5 Conclusions

The paper shows how to integrate symbolic knowledge into unsupervised neural networks. This work bases on the theoretical finding that any propositional formula can be represented in RBMs [Tran, 2017]. We show that converting background knowledge in the form of *if-then* rules to Confidence rules for encoding is efficient. In the experiments, we evaluate our approaches for DNA promoter prediction and relationship reasoning to show the validity of the approach.

References

[Avila Garcez and Zaverucha, 1999] Artur S. Avila Garcez and Gerson Zaverucha. The connectionist inductive learning and logic programming system. *Applied Intelligence*, 11(1):5977, July 1999.

[Bengio, 2009] Yoshua Bengio. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127, January 2009.

[França *et al.*, 2014] Manoel V. M. França, Gerson Zaverucha, and Artur S. d'AvilaGarcez. Fast relational learning using bottom clause propositionalization with artificial neural networks. *Machine Learning*, 94(1):81–104, 2014.

[Garcez *et al.*, 2008] Artur S. d'Avila Garcez, Lus C. Lamb, and Dov M. Gabbay. *Neural-Symbolic Cognitive Reasoning*. Springer Publishing Company, Incorporated, 2008.

[Hinton, 1986] Geoffrey E. Hinton. Learning distributed representations of concepts. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, pages 1–12. Hillsdale, NJ: Erlbaum, 1986.

[Penning *et al.*, 2011] Leo de Penning, Artur S. d'Avila Garcez, Lus C. Lamb, and John-Jules Ch Meyer. A neural-symbolic cognitive agent for online learning and reasoning. In *IJCAI*, pages 1653–1658, 2011.

[Pinkas, 1995] Gadi Pinkas. Reasoning, nonmonotonicity and learning in connectionist networks that capture propositional knowledge. *Artificial Intelligence*, 77(2):203–247, September 1995.

[Smolensky, 1995] Paul Smolensky. Constituent structure and explanation in an integrated connectionist/symbolic cognitive architecture. In C. Mcdonald, editor, *Connectionism: Debates on Psychological Explanation*, pages 221–290. Blackwell, Cambridge, 1995.

[Socher *et al.*, 2013] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, pages 926–934, USA, 2013. Curran Associates Inc.

[Sutskever and Hinton, 2008] Ilya Sutskever and Geoffrey E Hinton. Using matrices to model symbolic relationship. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1593–1600. Curran Associates, Inc., 2008.

[Towell and Shavlik, 1994] Geoffrey G. Towell and Jude W. Shavlik. Knowledge-based artificial neural networks. *Artificial Intelligence*, 70(1-2):119–165, 1994.

[Tran and d'Avila Garcez, 2013] Son N. Tran and Artur d'Avila Garcez. Knowledge extraction from deep belief networks for images. In *IJCAI-2013 Workshop on Neural-Symbolic Learning and Reasoning*, 2013.

[Tran and Garcez, 2016] Son Tran and Artur Garcez. Deep logic networks: Inserting and extracting knowledge from deep belief networks. *IEEE Transactions on Neural Networks and Learning Systems*, PP(99):1–13, 2016.

[Tran, 2017] Son N. Tran. Propositional knowledge representation in restricted boltzmann machines. https://arxiv.org/abs/1705.10899, 2017.

[Valiant, 2006] Leslie G. Valiant. Knowledge infusion. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pages 1546–1551, 2006.

Towards Explainable Tool Creation by a Robot

Handy Wicaksono[1,2] **Claude Sammut**[1] **Raymond Sheh**[3]
[1] School of Computer Science and Engineering, University of New South Wales; Sydney, Australia
[2] Department of Electrical Engineering, Petra Christian University; Surabaya, Indonesia
[3] Department of Computing, Curtin University; Perth, Australia
handyw@cse.unsw.edu.au, claude@cse.unsw.edu.au, Raymond.Sheh@curtin.edu.au

Abstract

A robot may benefit from being able to use a tool to solve a complex task. When an appropriate tool is not available, a useful ability for a robot would be to create a novel one based on its experiences. With the advent of inexpensive 3D printing, it is now possible to give robots such an ability. We propose CREATIVE, a relational approach to enable a robot to learn how to use an object as a tool and, if needed, to design and construct a new tool. The advantage of a relational approach is its interpretable learning results. To get meaningful explanations, we store the relevant knowledge during experiments, reason about them, and present them in a meaningful way to a human. Furthermore, a human user should be able to take action based on explanations given by a robot. We outline our plan to add this feature in CREATIVE and perform preliminary experiments on it.

1 Introduction

Humans use tools to solves complex problems in their daily life. When an appropriate tool is not available, we can innovate and design a new tool, often based on prior experience, or even invent it from scratch. Such abilities are also beneficial for a robot. Although there is increasing interest in tool use learning by a robot, there has been little work on tool creation. For example, Wang [2014] developed a robot that can create tools on the fly, but the design of the tools are predefined, and no learning is performed. Most of the existing work uses feature-based representations which are not expressive enough to be extended to tool creation.

Creating a novel tool by modifying or extending a previous design is called tool innovation in cognitive science [Beck *et al.*, 2011]. We take this approach as a means of limiting the search space of possible tools that the robot may try to design. The system uses a specific-to-general search, starting with a design of a known tool, but which either is not present or whose properties only partially match the requirements of the task. These properties are then modified by a generalization of the tool description.

We introduce a system called CREATIVE (Cognitive Robot Equipped with Autonomous Tool Invention Expertise), which refers to a robot that can learn how to use a tool and, if needed, design and build a new one. We use a relational representation of tool models, which are learned by a form of Inductive Logic Programming (ILP)[Sammut, 1981; Muggleton, 1991].

Figure 1: Baxter robot have a discussion with human about a novel tool it plans to create

Having an autonomous agent seems ruling out human user from the loop. However, recently there is an increasing demand to put human back in the loop, especially in critical fields, such as rescue, health, or military. Human user needs to know why (or why not) an AI agent makes such decisions, so he can trust it. We give its illustration in Fig. 1.

Explainable learning results is hard to achieve in "black box" AI mechanisms. Fortunately, CREATIVE utilises relational representation, so its results are inherently explainable as it describes the relation between objects as Prolog facts. Furthermore, it can give deeper explanation about robot's decisions compared than the feature-based one.

To summarize, here are the features that we propose to add in CREATIVE:

- Ability to debug the learned robot's plan so, if needed, errors can be corrected by human
- Ability to explain to human about why (or why not) the robot makes a decision
- Facility for human user to take action based on that explanations

In the following sections, we review relevant related work, describe our representation of states and actions, explain our learning mechanism, and present the preliminary experimental results.

2 Related Work

Previous work on tool use learning by a robot includes Stoytchev [2005], who demonstrated learning the tool affordances that are grounded in the robot's behaviors. Recent work by Tikhanoff et al. [2013] integrates exploratory behaviors and geometrical feature extraction to learn affordances and tool use. Mar et al. [2015] extend their work by learning the grasp configurations that affect the outcome of a tool use action. All of these systems use feature vector representations, and therefore, are limited in their ability to describe relations between components of a tool and their relations to other objects. A relational approach overcomes this limitation [Brown and Sammut, 2012].

There has been little prior work on tool creation. Wang et. al. [2014] developed a manipulator robot that can manufacture tools on the fly. However, the designs of tools are predefined, not learned. Brodbeck et al. [2015] performed evolutionary stochastic optimization of mechanical designs to construct a complex agent. This evolutionary approach is also limited to feature-based representations and learning takes a long time as it can not incorporate any background knowledge. ILP is capable of learning more expressive relational representation, and it can easily make use of background knowledge.

A relational representation also produces easy-to-interpret learning results, which are desirable in AI application. Recently there are increasing interests in this field, namely eXplainable Artificial Intelligence or XAI [DARPA, 2016], as many popular and accurate AI algorithms behave like black boxes. To tackle this issue in robotics, a verbalization is done so a mobile robot can "tell" its experience in a way that understandable by humans [Rosenthal *et al.*, 2016]. Other work enables a rescue robot to explain its actions to humans by converting a decision tree into a human-friendly information [Sheh, 2017]. Our representation is more general than previous work so that a deeper explanation can be expected.

3 States and Actions Representation

We maintain two levels of state representation: abstract and primitive. Primitive states contain quantitative values, such as the pose of objects in the world, while abstract states capture qualitative relationships between objects. As we operate in an ILP framework, state representations are expressed as a set of Prolog facts.

We define a tool as an object that is deliberately employed by an agent to help it achieve a goal that would otherwise be difficult or impossible to achieve. A tool possesses spatial and structural properties. We build a simple ontology of tools, where a general tool can be specified into a hook, a wedge, a hammer or other kind of tools. Each tool has unique structural and material properties. A hierarchy example of hook's properties is shown in Fig. 2.

We use a STRIPS [Fikes and Nilsson, 1971] action model to describe tool actions:

PRE: condition that must hold so that the action can be performed

EFFECTS: conditions that become true or false as a result of performing the action

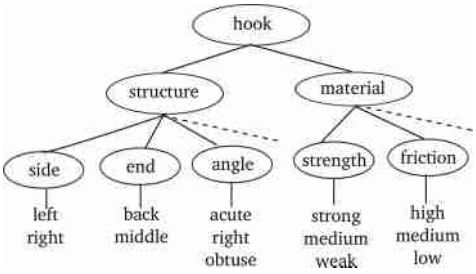


Figure 2: Properties of a hook

The action model does not provide any quantitative information needed to execute a motor command. For example, an action model might state that a hook must be placed behind a target object to be pulled, but it does not give a precise position. However, spatial literals in the action model's effects (e.g. behind(X, Y)) can be treated as constraints to a constraint solver [Apt *et al.*, 2007], which later produces a set of quantitive parameter values, any of which can be used to achieve the effects.

4 CREATIVE

We illustrate our learning framework for CREATIVE in Fig. 3. Initially, a robot does not have a complete action models, so it can not construct a plan. The robot can **learn by observing** a single correct example given by a tutor and build an initial novel model to complete the previous ones. A problem solver must achieve a goal and may use a tool to do it. Results of its attempt to solve the problem are sent to a critic that determines if they correspond to the expectations of a planner. Depending on that assessment, **learning by trial and error** is performed, via an ILP learner, to update a relevant action model. A problem generator selects a new experiment to test the updated model. If there are no suitable tool to accomplish the task, **tool invention** is performed. A label from a critic is passed to the tool generalizer and manufacturer. A simple **user interface** is added so a human user can get explanations from a robot.

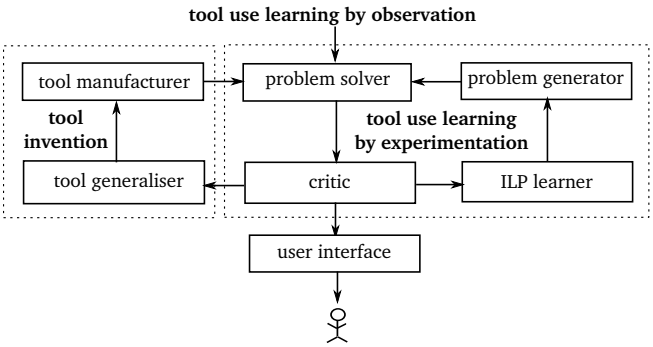


Figure 3: CREATIVE learning framework with a user interface

4.1 Tool Use Learning

In learning by observation stage, the robot is shown a correct example by a trainer, which is used to construct the initial action model. Primitive states are acquired by a vision sensor and grouped as segments. They are matched with existing action models. If there are segments that have no match, a new action models is constructed. Preconditions of this new action model contain a hypothesis which describes tool's required properties.

Our hypothesis formation adapts Mitchell's [Mitchell, 1977] version space approach in that the initial example represents the system's most specific hypothesis (h_s) for the tool's structure and pose and the most general hypothesis (h_g), initially, covers everything. The system refines its hypothesis by conducting experiments. It may generalize the most specific hypothesis, then construct an instance that is consistent with its generalization, and vice versa.

To select the tool, the object that matches the most structural properties of the tool in the trainer's example is chosen. To select the pose, firstly, we collect the spatial literals in h_g then append them to the shuffled spatial predicates in h_s. Those predicates are treated as constraints and solved by a constraint solver to get numerical goal.

The learning algorithm is borrowed from Golem [Muggleton and Feng, 1990] which performs a Relative Least General Generalisation (RLGG) [Plotkin, 1971]. h_s is refined by finding the constrained Least General Generalisation (LGG) of a pair of positive examples, and h_g is refined by performing a negative-based reduction. Our tool use learning algorithm is a modification from Haber's [2015], and can be seen in our recent work [2016].

4.2 Tool Creation

The robot is provided with a simple ontology to describe the properties of simple tools. In Fig.2, this is limited to hook-like tools. Note that in the present work, this ontology is handcrafted, but it is possible to learn ontologies with ILP. The main function of our ontology is to limit the search space when a generalization is conducted. It is similar to Shapiro's "refinement graph" in MIS [Shapiro, 1983].

In tool creation, the refinement graph can be used to suggest generalizations by climbing the generalization hierarchy. For example, if the current model requires a hook to be on the right-hand side of the tool, a generalization may suggest that the hook can be on either side. In this case, the new hypothesis is tested by generating a new instance of the hypothesis that does not match previous instances seen or constructed.

Even though the refinement graph limits the search, a reasonably large graph may give rise to many possible configurations. The system could attempt to find a suitable tool by manufacturing and testing all possible tools exhaustively. To avoid this, the system prioritizes new tools that are more similar to the old one. The similarity is computed by the Levenshtein distance [Levenshtein, 1966], the number of edit operations needed to transform one representation to another.

The numerical sizes of tools are acquired by treating the structural properties as constraints and solving them. In a simulation, a tool is "manufactured" by converting its description into a URDF (Universal Robotic Description Format) file, which can be fed to Gazebo simulator. Because of a limited camera's field of view, we can only test five new tools in one batch. Tools with smaller edit distances are prioritized. If a tool use fails, the system moves on to the next one, otherwise, it checks which of the new tool's properties are different from the old one and generalize them. Our mechanism is described in Algorithm 1.

Algorithm 1 Tool creation in simulation

Require: h_s, h_g, background knowledge (BK), experiment status,learning status
1: Apply RLGG to structural properties of h_s and BK to get "generalised properties"
2: Instantiate "generalised properties" to get a collection of grounded properties of new tools
3: **for** each new tool **do**
4: Find the edit distance
5: Do constraints solving to get its numerical size
6: Create a tool in URDF
7: Select 5 novel tools with minimum edit distances.
8: Test the tool one by one.
9: **if** The testing is successful **then**
10: In old and new tools, find the literals that have same functor names, but different grounded arguments
11: Generalise the relevant properties
12: Update h_s and h_g
return Updated h_s and h_g

4.3 Explainable Tool Creation

We have an advantage in having relational representation, as its learning results are inherently interpretable by a human. Hence, to make our tool creation explainable for a human is a matter of **storing** all relevant information, including the learned hypotheses in Prolog, the primitive objects poses and the camera snapshots, **reasoning** on that knowledge base, and **presenting** them in a meaningful way to a human.

Essentially, we want the robot to have "conversation" with a human so failures can be prevented, errors can be resolved, and learning time can be reduced. There are two things that a human user can do here:

- Fully debugging the learned robot's plan in Prolog to find and correct the errors. This is following the bug-correction algorithm [Shapiro, 1983].
- Asking the robot about why it makes such a decision, and take action based on its explanations.

We will clarify these ideas on section 5.3.

5 Results and Discussions

We evaluate the CREATIVE algorithm by conducting experiments in tool use learning and tool creation. The preliminary work to demonstrate the explainability feature is also given here.

5.1 Tool use learning experiment

We perform experiments in the Gazebo simulator, which incorporates realistic physics engine [Wicaksono and Sammut,

2016]. The simulation environment includes the Baxter robot, a cube that the robot is required to retrieve from inside a tube, and five different objects that may be used as tools. We prepare two sets of five different tools based on the width of their hooks. We then randomly select the width of those tools in each trial. A downward-facing web camera is set up to capture the whole scene. A camera is also mounted in Baxter's gripper. These are required to grab the tool and pull the cube accurately.

As our primary goal is to demonstrate tool creation, we speed up our experiments by using the action models learned by Brown (2009) as background knowledge. These are then refined by experimentation. 12 learning episodes are required for the robot to learn these properties of a reliable tool [Wicaksono and Sammut, 2016]:

- The hook is attached on the same side as the cube's position inside the tube.
- The handle touches the cube on the same side as its position inside the tube.
- The hook touches the cube on its back.
- The hook is located on the same end as the cube's location inside the tube.

5.2 Tool creation experiment

Here, we change the environment to trigger the creation of novel tools, by creating a longer tube compare to the previous one. The old tools will fail as it is impossible to reach far enough into the tube, and the robot will begin to create new tools. It starts by by applying RLGG on the structural properties of the tool and the background knowledge. Only properties with grounded value that will be generalized.

The instantiation can be performed later on, so 18 potential tools are generated. We only care for the novel tools (15 tools) and ignore the old ones (3 tools). All generated tools and their edit distances are shown in Fig. 4. The black tool is old and useful, the gray tools are old but not useful, while the white ones are novel and will be tested later.

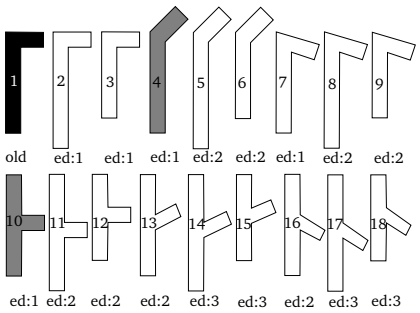


Figure 4: All generated tools and their edit distances

To select the novel tools we use two strategies: random selection and ranked selection. As the limitation of our vision sensor, only 5 tools can be tested at a time. As the result, the second strategy finds the useful novel tool, a tool with similar shape with the old one but has a longer handle (number 2), only in 3 episodes, while the first one finds it in 8 episodes.

After a useful novel tool is found, we compare its properties with the ones belonging to the old tool. To be more specific, we look for a literal with the same functor name but has different ground value. In this case, we find that `handle_length(Handle, medium)` is not valid any more and should be generalised into `handle_length(Handle, Length)`.

5.3 Experiment in explainable tool creation

Fully debugging on a learned robot's plan in Prolog is straightforward by following Shapiro's bug-correction algorithm [Shapiro, 1983]. Here, human acts as an infallible oracle, to whom the robot asks whether a Prolog predicate is correct or not. If it is incorrect, the robot will suggest the correction, and the debugging loop will be repeated until the human oracle think everything is correct.

The human-robot "conversation" on robot's decisions to create a tool is more abstract. Because of the pages limitation, we only illustrate it in a simplified and easy-to-understand form. Currently, we do not do any natural language processing, so the complete dialogue in Prolog is just using a set of simple questions and answers. The italic words are based on robot's knowledge.

ROBOT : I will print a *right_acute_hook* that has *acute_angle*, *right_side*.
HUMAN : Why you choose *right_acute_hook* ?
ROBOT : This is the successful tool after *16 experiments*. Here are *the learned hypotheses* and *the snapshots of tools and environments*.
HUMAN : Why don't you choose hook with *obtuse_angle* ?
ROBOT : I have tried *right_obtuse_hook* in *experiment 15*, but it *fails (60%)*. Do you want me to try it?
HUMAN : No, I've just checked. How about tool with *right_angle* ?
ROBOT : I have not tried *right_right_hook* yet. Do you want me to try it?
HUMAN : Yes. I think it is more likely to success compared than the one that you suggest.
ROBOT : Okay, I will try it. I will get back to you later.

Currently, this feature is only tested locally. In the future, it will be integrated into our system.

6 Conclusions and future work

We have equipped the simulated Baxter robot with an autonomous tools invention expertise. A tool ontology is developed to represent the tools and reduce the search space. We extend tool use learning, so the robot can generate novel tools and find the useful one based on the past experiences. A feature to enable robot explains its decision to human is developed, so better decision and reduced learning time can be expected.

In the future, we will build a system that combines a physical and simulated robot. Further evaluations that include another kind of tools also need to be done. Lastly, we plan to fully integrate the explainability feature into our system and improve it by providing deeper knowledge and more intuitive user interface.

Acknowledgments

Handy Wicaksono is supported by The Directorate General of Resources for Science, Technology and Higher Education (DG-RSTHE), Ministry of Research, Technology, and Higher Education of the Republic of Indonesia.

References

[Apt *et al.*, 2007] Krzysztof R Apt, Mark Wallace, et al. *Constraint logic programming using ECLiPSe*. Cambridge University Press New York, 2007.

[Beck *et al.*, 2011] Sarah R. Beck, Ian A. Apperly, Jackie Chappell, Carlie Guthrie, and Nicola Cutting. Making tools isn't child's play. *Cognition*, 119(2):301 – 306, 2011.

[Brodbeck *et al.*, 2015] Luzius Brodbeck, Simon Hauser, and Fumiya Iida. Morphological evolution of physical robots through model-free phenotype development. *PLOS ONE*, 10(6):1–17, 06 2015.

[Brown and Sammut, 2012] Solly Brown and Claude Sammut. A relational approach to tool-use learning in robots. In *Inductive Logic Programming*, pages 1–15. Springer, 2012.

[DARPA, 2016] DARPA. *Broad Agency Announcement: Explainable Artificial Intelligence (XAI)*, 2016.

[Fikes and Nilsson, 1971] Richard Fikes and Nils J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artif. Intell.*, 2(3/4):189–208, 1971.

[Haber, 2015] Adam Haber. *A system architecture for learning robots*. PhD thesis, School of Computer Science and Engineering, UNSW Australia, 2015.

[Levenshtein, 1966] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.

[Mar *et al.*, 2015] Tanis Mar, Vadim Tikhanoff, Giorgio Metta, and Lorenzo Natale. Self-supervised learning of grasp dependent tool affordances on the icub humanoid robot. In *Proceedings of ICRA*, May 2015.

[Mitchell, 1977] Tom M Mitchell. Version spaces: A candidate elimination approach to rule learning. In *Proceedings of the 5th international joint conference on Artificial intelligence-Volume 1*, pages 305–310. Morgan Kaufmann Publishers Inc., 1977.

[Muggleton and Feng, 1990] Stephen Muggleton and Cao Feng. Efficient induction of logic programs. In *New Generation Computing*. Academic Press, 1990.

[Muggleton, 1991] Stephen Muggleton. Inductive logic programming. *New Generation Computing*, 8(4):295–318, 1991.

[Plotkin, 1971] G. D. Plotkin. A further note on inductive generalization. In B. Meltzer and D. Michie, editors, *Machine Intelligence 6*. Elsevier, New York, 1971.

[Rosenthal *et al.*, 2016] Stephanie Rosenthal, Sai P Selvaraj, and Manuela Veloso. Verbalization: Narration of autonomous robot experience. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 862–868. AAAI Press, 2016.

[Sammut, 1981] C. A. Sammut. Concept learning by experiment. In *Seventh International Joint Conference on Artificial Intelligence*, pages 104–105, Vancouver, 1981.

[Shapiro, 1983] Ehud Y Shapiro. *Algorithmic program debugging*. MIT press, 1983.

[Sheh, 2017] Raymond Sheh. "why did you do that?" explainable intelligent robots. In *AAAI Workshop on Human-Aware Artificial Intelligence*, 2017.

[Stoytchev, 2005] A. Stoytchev. Behavior-grounded representation of tool affordances. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 3060–3065, April 2005.

[Tikhanoff *et al.*, 2013] V. Tikhanoff, U. Pattacini, L. Natale, and G. Metta. Exploring affordances and tool use on the icub. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 130–137, Oct 2013.

[Wang *et al.*, 2014] Liyu Wang, Luzius Brodbeck, and Fumiya Iida. Mechanics and energetics in tool manufacture and use: a synthetic approach. *Journal of The Royal Society Interface*, 11(100), 2014.

[Wicaksono and Sammut, 2016] Handy Wicaksono and Claude Sammut. Relational tool use learning by a robot in a real and simulated world. In *Proceedings of ACRA*, December 2016.