The Springer Series on Challenges in Machine Learning

Hugo Jair Escalante · Sergio Escalera
Isabelle Guyon · Xavier Baró
Yağmur Güçlütürk · Umut Güçlü
Marcel van Gerven *Editors*

Explainable and Interpretable Models in Computer Vision and Machine Learning

Springer

The Springer Series on Challenges in Machine Learning

The books of this innovative series collect papers written by successful competitions in machine learning. They also include analyses of the challenges, tutorial material, dataset descriptions, and pointers to data and software. Together with the websites of the challenge competitions, they offer a complete teaching toolkit and a valuable resource for engineers and scientists.

More information about this series at http://www.springer.com/series/15602

Hugo Jair Escalante • Sergio Escalera
Isabelle Guyon • Xavier Baró • Yağmur Güçlütürk
Umut Güçlü • Marcel van Gerven
Editors

Explainable and Interpretable Models in Computer Vision and Machine Learning

Springer

Editors

Hugo Jair Escalante
INAOE
Puebla, Mexico

Isabelle Guyon
INRIA, Université Paris Sud, Université
Paris Saclay, Paris, France

ChaLearn
Berkeley, CA, USA

Yağmur Güçlütürk
Radboud University Nijmegen
Nijmegen, The Netherlands

Marcel van Gerven
Radboud University Nijmegen
Nijmegen, The Netherlands

Sergio Escalera
University of Barcelona
Barcelona, Spain

Xavier Baró
Open University of Catalonia
Barcelona, Spain

Umut Güçlü
Radboud University Nijmegen
Nijmegen, The Netherlands

ISSN 2520-131X ISSN 2520-1328 (electronic)
The Springer Series on Challenges in Machine Learning
ISBN 978-3-319-98130-7 ISBN 978-3-319-98131-4 (eBook)
https://doi.org/10.1007/978-3-319-98131-4

Library of Congress Control Number: 2018962179

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

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

Foreword

“Too much of a black box to me”. That is an often heard and long-standing criticism of data-driven machine learning methods, in general, and (deep) neural networks, in particular. Nevertheless, astounding results have been obtained with these black boxes.

Interestingly, one could argue that this is, to a large extent, not in spite of but rather thanks to their black box nature: researchers no longer aim at full control over the model intrinsics, a common practice in the hand-crafted features era. Instead, the data leads the way, optimising the whole system in an end-to-end manner for the task at hand, yielding superior results.

The flip side of the coin is that, given the complexity of the models used these days, with millions of parameters, it is hard to understand the processes inside the box. As a consequence, the question rises whether such systems can be trusted at all – especially when it comes to safety-critical applications such as self-driving cars or medical image interpretation.

Three more observations further add to this scepticism. First, networks often struggle to generalise beyond the circumstances seen at training time. Yet, they keep making (often wrong) predictions with high confidence for out-of-distribution samples. Second, there is the issue with adversarial examples, where it has been shown that adding relatively low amounts of noise suffices to change the output of a neural network in an arbitrary predefined direction. Finally, with artificial systems reaching or even surpassing human performance, the long-standing criticism of the black box approach now becomes more relevant than ever.

After the initial enthusiasm at the start of the third summer of AI about the good performance obtained with deep learning, more and more concerns are raised along the lines sketched above. As a countermeasure, we need more research towards model explainability and interpretability. Let us build a new generation of machine learning models that are capable not only of predicting the output with high accuracy but also of explaining the produced result and enabling researchers to interpret the learned models. This is a challenging endeavour, with several open research questions: How to visualise or communicate model explanations and interpretations with the user? How to avoid a misguided feeling of trust? How to evaluate model explanations and interpretations? How to avoid or deal with subjectivity in this matter? Within this book, a fine collection of the current state of the art in this direction is brought together, highlighting different approaches to tackle the problem.

KU Leuven, Flanders, Belgium Tinne Tuytelaars
June 2018

Preface

Research progress in computer vision and pattern recognition has led to a variety of modelling techniques with (almost) human-like performance in a variety of tasks. A clear example of this type of models is neural networks, whose deep variants dominate the arenas of computer vision among other fields. Although this type of models has obtained astounding results in a variety of tasks (e.g. face recognition), they are limited in their explainability and interpretability. That is, in general, users cannot say too much about:

- What is the rationale behind the decision made? (explainability)
- What in the model structure explains its functioning? (interpretability)

Hence, while good performance is a critical required characteristic for learning machines, explainability/interpretability capabilities are highly needed if one wants to take learning machines to the next step and, in particular, include them into decision support systems involving human supervision (for instance, in medicine or in security). Because of their critical importance, there is a research trend within the computer vision and machine learning communities in studying both aspects. In fact, in recent years, much work has been devoted to defining what is explainability and interpretability in the context of models and how to evaluate these aspects, proposing and analysing mechanisms for explaining recommendations of models and interpreting their structure.

All this progress puts us in perfect time to compile in a single book the latest research advances on explainable and interpretable models in the context of computer vision and machine learning. The book is divided into four parts that cover complimentary and relevant topics around this subject.

Part I focuses on general notions and concepts around explainability and interpretability. F. Doshi-Velez and Kim elaborate on considerations for the evaluation of interpretable machine learning models. They provide a definition of interpretability, principles for evaluation and a taxonomy of evaluation approaches. They conclude with recommendations for researchers in the field. In the same line, Ras et al. elaborate on issues regarding deep learning and explainability, trying to bridge a gap between expert users and lay/average users. They discuss the relation between users laws and regulations, explanations and methods in the context of explainability.

The second part of the book is devoted to chapters that focus on explainability and interpretability from the machine learning point of view. Goudet et al. describe Causal Generative Neural Networks, a methodology to infer causal relations from observational data. More specifically, they provide a means to estimate a generative model of the joint distribution of observed variables. Since causality is the ultimate explanatory mechanism desired for most modelling techniques, this methodology can have a great impact into the field. Loza Mencia et al. contribute to the book with a chapter on rule-based methods for multi-label classification. The chapter emphasises the interpretability characteristics of rule-based approaches to multi-label classification, and two approaches for learning predictive rule-based models are reviewed in detail. Rieger et al. study the relationship between performance and quality of explainability in the context of deep neural networks. They aim to determine whether explanations exhibit a systematic bias and how the structure of the neural network can be adapted to reduce such bias.

The third part of the book focuses on explainability and interpretability in computer vision. Akata el al. describe a novel methodology for generating explanations in image-object classification. The key features of the proposed method are a relevance loss that conditions sentence generation on the image category and, on the other hand, a discriminative loss inspired on reinforcement learning that relies on a sentence classifier. N. Fatema and R. Mooney present a novel methodology to generate explanations for ensembles of visual question answering (VQA) systems. In addition, two evaluation protocols are described and used to evaluate explanations generated by their ensemble. This is among the first works dealing with explanation of ensembles of VQA systems.J. Kim and J. Canny describe a methodology for generating visually interpretable images in the context of autonomous vehicle driving. The methodology comprises two steps: a CNN with attention model that highlights potentially salient regions in images and a filtering step that aims at removing spurious salient regions. The methodology is extensively evaluated, comprising qualitative and quantitative assessments.

Last but not least, Part IV covers methodologies related to explainability and interpretability in the context of first impressions and job candidate screening. Liem et al. elaborate on the gap between machine learning (and computer science in general) and psychology in the context of job candidate screening. Through a detailed review, the authors try to fill an understanding gap between both areas. Liem at al. describe their solution to the job candidate screening competition. H. Kaya and A. Salah describe the winning methodology of the job candidate screening competition. The authors focus on the explanatory characteristics of their solution and discuss the potential bias of their model. Similarly, Aakur et al. describe their winning methodology for the explainable job candidate screening challenge. The authors provide an detailed description of their method and an in depth analysis of their results.

To the best of our knowledge, this is the first compilation of research on this topic. We were fortunate to gather 11 chapters of extraordinary quality that, together, capture a snapshot of the state of the art in this pretty much important topic.

Puebla, Mexico	Hugo Jair Escalante
Barcelona, Spain	Sergio Escalera
Paris, France	Isabelle Guyon
Barcelona, Spain	Xavier Baró
Nijmegen, The Netherlands	Yağmur Güçlütürk
Nijmegen, The Netherlands	Umut Güçlü
Nijmegen, The Netherlands	Marcel van Gerven
June 2018	

Acknowledgements

The editors are grateful with the attendees and participants of the First Impressions Challenges @ECCV16 and @ICPR2916 and the Job Candidate Screening Coopetition @CVPR2017 @IJCNN2017. The editors gratefully acknowledge a grant from Azure for Research, which allowed running the challenges on the CodaLab platform and the technical support of Université Paris-Saclay. ChaLearn provided prizes and travel awards to the winners of the aforementioned competitions. This work was partially supported by CONACyT under grant 241306, Spanish Ministry projects TIN2016-74946-P and TIN2015-66951-C2-2-R (MINECO/FEDER, UE), and CERCA Programme/Generalitat de Catalunya. H. J. Escalante was supported by *Red Temáticas CONACyT en Tecnologías del Lenguaje (RedTTL) e Inteligencia Computacional Aplicada (RedICA)*. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

Contents

Part I Notions and Concepts on Explainability and Interpretability

Part II Explainability and Interpretability in Machine Learning

Part III Explainability and Interpretability in Computer Vision

Part IV Explainability and Interpretability in First Impressions Analysis

Contributors

Sathyanarayanan N. Aakur University of South Florida, Department of Computer Science and Engineering, Tampa, FL, USA

Zeynep Akata AMLAB, University of Amsterdam, Amsterdam, The Netherlands

Stephan Alaniz AMLAB, University of Amsterdam, Amsterdam, The Netherlands

Marise Ph. Born Erasmus School of Social and Behavioral Sciences, Erasmus University, Rotterdam, The Netherlands

Philippe Caillou Team TAU - CNRS, INRIA, Université Paris Sud, Université Paris Saclay, Paris, France

John Canny Department of Electrical Engineering and Computer Sciences, UC Berkeley, Berkeley, CA, USA

Pattarawat Chormai Department of Electrical Engineering and Computer Science, Technische Universität Berlin, Berlin, Germany

Trevor Darrell EECS, University of California Berkeley, Berkeley, CA, USA

Andrew Demetriou Multimedia Computing Group, Delft University of Technology, Delft, The Netherlands

Fillipe D. M. de Souza University of South Florida, Department of Computer Science and Engineering, Tampa, FL, USA

Finale Doshi-Velez Harvard University, Cambridge, MA, USA

Johannes Fürnkranz Knowledge Engineering Group, Technische Universität Darmstadt, Darmstadt, Germany

Olivier Goudet Team TAU - CNRS, INRIA, Université Paris Sud, Université Paris Saclay, Paris, France

Isabelle Guyon INRIA, Université Paris Sud, Université Paris Saclay, Paris, France

ChaLearn, Berkeley, CA, USA

Lars Kai Hansen DTU Compute, Technical University of Denmark, Kongens Lyngby, Denmark,

Pim Haselager Radboud University, Nijmegen, The Netherlands

Lisa Anne Hendricks EECS, University of California Berkeley, Berkeley, CA, USA

Annemarie M. F. Hiemstra Erasmus School of Social and Behavioral Sciences, Erasmus University, Rotterdam, The Netherlands

Eyke Hüllermeier Intelligent Systems, Universität Paderborn, Paderborn, Germany

Diviyan Kalainathan Team TAU - CNRS, INRIA, Université Paris Sud, Université Paris Saclay, Paris, France

Heysem Kaya Department of Computer Engineering, Namik Kemal University, Corlu, Tekirdag, Turkey

Been Kim Google Brain, Mountain View, CA, USA

Jinkyu Kim Department of Electrical Engineering and Computer Sciences, UC Berkeley, Berkeley, CA, USA

Cornelius J. König Universität des Saarlandes, Saarbrücken, Germany

Markus Langer Universität des Saarlandes, Saarbrücken, Germany

Cynthia C. S. Liem Multimedia Computing Group, Delft University of Technology, Delft, The Netherlands

David Lopez-Paz Facebook AI Research, Menlo Park, CA, USA

Eneldo Loza Mencía Knowledge Engineering Group, Technische Universität Darmstadt, Darmstadt, Germany

Grégoire Montavon Department of Electrical Engineering and Computer Science, Technische Universität Berlin, Berlin, Germany

Raymond J. Mooney Department of Computer Science, The University of Texas at Austin, Austin, TX, USA

Klaus-Robert Müller Department of Electrical Engineering and Computer Science, Technische Universität Berlin, Berlin, Germany

Department of Brain and Cognitive Engineering, Korea University, Seongbuk-gu, Seoul, South Korea

Max Planck Institute for Informatics, Saarbrücken, Germany

Nazneen Fatema Rajani Department of Computer Science, The University of Texas at Austin, Austin, TX, USA

Michael Rapp Knowledge Engineering Group, Technische Universität Darmstadt, Darmstadt, Germany

Gabriëlle Ras Radboud University, Nijmegen, The Netherlands

Laura Rieger DTU Compute, Technical University of Denmark, Kongens Lyngby, Denmark

Albert Ali Salah Department of Computer Engineering, Bogazici University, Istanbul, Turkey

Future Value Creation Research Center, Nagoya University, Nagoya, Japan

Sudeep Sarkar University of South Florida, Department of Computer Science and Engineering, Tampa, FL, USA

Michèle Sebag Team TAU - CNRS, INRIA, Université Paris Sud, Université Paris Saclay, Paris, France

Marcel van Gerven Radboud University, Nijmegen, The Netherlands

Achmadnoer Sukma Wicaksana Datasintesa Teknologi Nusantara, Jakarta, Indonesia

Part I
Notions and Concepts on Explainability and Interpretability

Considerations for Evaluation and Generalization in Interpretable Machine Learning

Finale Doshi-Velez and Been Kim

Abstract As machine learning systems become ubiquitous, there has been a surge of interest in interpretable machine learning: systems that provide explanation for their outputs. These explanations are often used to qualitatively assess other criteria such as safety or non-discrimination. However, despite the interest in interpretability, there is little consensus on what interpretable machine learning is and how it should be measured and evaluated. In this paper, we discuss a definitions of interpretability and describe when interpretability is needed (and when it is not). Finally, we talk about a taxonomy for rigorous evaluation, and recommendations for researchers. We will end with discussing open questions and concrete problems for new researchers.

Keywords Interpretability · Machine learning · Accountability · Transparency

1 Introduction

From autonomous cars and adaptive email-filters to predictive policing systems, machine learning (ML) systems are increasingly commonplace; they outperform humans on specific tasks (Mnih et al. 2013; Silver et al. 2016; Hamill 2017) and often guide processes of human understanding and decisions (Carton et al. 2016; Doshi-Velez et al. 2014). The deployment of ML systems in complex, realworld

Authors "Finale Doshi-Velez and Been Kim" contributed equally.

F. Doshi-Velez (✉)
Harvard University, Cambridge, MA, USA
e-mail: finale@seas.harvard.edu

B. Kim
Google Brain, Mountain View, CA, USA
e-mail: beenkim@google.com

H. J. Escalante et al. (eds.), *Explainable and Interpretable Models in Computer Vision and Machine Learning*, The Springer Series on Challenges in Machine Learning, https://doi.org/10.1007/978-3-319-98131-4_1

settings has led to increasing interest in systems optimized not only for expected task performance but also other important criteria such as safety (Otte 2013; Amodei et al. 2016; Varshney and Alemzadeh 2016), nondiscrimination (Bostrom and Yudkowsky 2014; Ruggieri et al. 2010; Hardt et al. 2016), avoiding technical debt (Sculley et al. 2015), or satisfying the right to explanation (Goodman and Flaxman 2016). For ML systems to be used robustly in realworld situations, satisfying these auxiliary criteria is critical. However, unlike measures of performance such as accuracy, these criteria often cannot be completely quantified. For example, we might not be able to enumerate all unit tests required for the safe operation of a semi-autonomous car or all confounds that might cause a credit scoring system to be discriminatory. In such cases, a popular fallback is the criterion of *interpretability*: if the system can *explain* its reasoning, we then can verify whether that reasoning is sound with respect to these auxiliary criteria.

Unfortunately, there is little consensus on what interpretability in machine learning *is*—let alone how to *evaluate* it for benchmarking or reason about how it may *generalize* to other contexts. Current interpretability evaluation typically falls into two categories. The first evaluates interpretability in the context of an application: if the interpretable system provides human-understandable explanation in either a practical application or a simplified version of it, then it must be interpretable (e.g. Ribeiro et al. 2016; Lei et al. 2016; Kim et al. 2015a; Doshi-Velez et al. 2015; Kim et al. 2015b). The second evaluates interpretability via a quantifiable proxy: a researcher might first claim that some model class—e.g. sparse linear models, rule lists, gradient boosted trees—are interpretable and then present algorithms to optimize within that class (e.g. Buciluǎ et al. 2006; Wang et al. 2017; Wang and Rudin 2015; Lou et al. 2012).

To large extent, both evaluation approaches rely on some notion of "you'll know it when you see it." Should we be concerned about a lack of rigor? Yes and no: the notions of interpretability above appear reasonable because they *are* reasonable: they pass the first test of having face-validity on the correct test set of subjects: human beings. However, this basic notion leaves many kinds of questions unanswerable: Are all models in all defined-to-be-interpretable model classes equally interpretable? Quantifiable proxies such as sparsity may seem to allow for comparison, but how does one think about comparing a model sparse in features to a model sparse in prototypes? Moreover, if one builds and evaluates an interpretable machine learning model from a particular dataset for a particular application, does that provide insights on whether the model will be similarly interpretable with a different dataset or different application? If we are to move this field forward—to compare methods and understand when methods may generalize—we need to formalize these notions and make them evidence-based.

The objective of this chapter is to describe a set of principles for the evaluation of interpretability. The need is urgent: European Union regulation may *require* algorithms that make decisions based on user-level predictors and "significantly affect" users to provide explanation ("right to explanation") (Parliament and of the European Union 2016). Meanwhile, interpretable machine learning is an increasingly popular area of research, with forms of interpretability ranging from regressions with simplified functions (e.g. Caruana et al. 2015; Kim et al. 2015a; Rüping 2006; Buciluǎ et al. 2006; Ustun and Rudin 2016; Doshi-Velez et al. 2015; Kim et al. 2015b; Krakovna and Doshi-Velez 2016; Hughes et al. 2016), various kinds of logic-based methods (e.g. Wang and Rudin 2015; Lakkaraju et al. 2016; Singh et al. 2016; Liu and Tsang 2016; Safavian and Landgrebe 1991; Wang et al. 2017), methods of probing black box models (e.g. Ribeiro et al. 2016; Lei et al. 2016; Adler et al. 2016; Selvaraju et al. 2016; Smilkov et al. 2017; Shrikumar et al. 2016; Kindermans et al. 2017; Ross et al. 2017; Singh et al. 2016). International conferences regularly have workshops on interpretable machine learning, and Google Scholar finds more than 20,000 publications related to interpretability in ML in the last 5 years. How do we know which methods work best when? While there have been reviews of interpretable machine learning more broadly (e.g. Lipton 2016), the lack of consensus on how to evaluate interpretability limits both research progress and the effectiveness of interpretability-related regulation.

In this chapter, we start with a short discussion of what interpretability is Sect. 2. Next we describe when interpretability is needed, including a taxonomy of use-cases (Sect. 3). In Sect. 4, we review current approaches to evaluation and propose a taxonomy for the evaluation of interpretability—application-grounded, human-grounded and functionally-grounded. Finally, we discuss considerations for generalization in Sect. 5. We review suggestions for researchers doing work in interpretability in Sect. 6.

2 Defining Interpretability

According to the Merriam-Webster dictionary, the verb *interpret* means *to explain or to present in understandable terms*.[1] In the context of ML systems, we add an emphasis on providing explanation to humans, that is, *to explain or to present in understandable terms to a human*.

While explanation may be a more intuitive term than interpretability, we still must answer what then is an explanation? A formal definition of explanation remains elusive; we turn to the field of psychology for insights. Lombrozo (2006) argue that “explanations are more than a human preoccupation—they are central to our senses of understanding, and the currency in which we exchanged beliefs” and notes that questions such as what constitutes an explanation, what makes some explanations better than others, how explanations are generated and when explanations are sought are just beginning to be addressed. Indeed, the definition of explanation in the psychology literature ranges from the “deductive-nomological” view (Hempel and Oppenheim 1948), where explanations are thought of as logical proofs to providing some more general sense of mechanism (Bechtel and Abrahamsen 2005; Chater and Oaksford 2006; Glennan 2002). More recently (Keil 2006)

[1] Merriam-Webster dictionary, accessed 2017-02-07.

considered a broader definition of explanations—implicit explanatory understanding. All the activities in the processes of providing and receiving explanations are considered as a part of what explanation means.

In this chapter, we propose data-driven ways to derive operational definitions and evaluations of explanations. We emphasize that the explanation needs within the context of an application may not require knowing the flow of bits through a complex neural architecture—it may be much simpler, such as being able to identify to which input the model was most sensitive, or whether a protected category was used when making a decision.

3 Defining the Interpretability Need

Interpretable Machine Learning as a Verification Tool

In Sect. 1, we mentioned that interpretability is often used as a proxy for some other criteria. There exist many desiderata that we might want of our ML systems. Notions of *fairness* or *unbiasedness* imply that protected groups (explicit or implicit) are not somehow discriminated against. *Privacy* means the method protects sensitive information in the data. Properties such as *safety*, *reliability* and *robustness* ascertain whether algorithms reach certain levels of performance in the face of parameter or input variation. *Causality* implies that the predicted change in output due to a perturbation will occur in the real system. *Usable* methods provide information that assist users to accomplish a task—e.g. a knob to tweak image lighting—while *trusted* systems have the confidence of human users—e.g. aircraft collision avoidance systems.

There exist many ways of verifying whether an ML system meets such desiderata. In some cases, properties can be proven. For example, formalizations of fairness (Hardt et al. 2016) and privacy (Toubiana et al. 2010; Dwork et al. 2012; Hardt and Talwar 2010) have resulted in algorithms that are guaranteed to meet those criteria. In other cases, we can track the performance of a system and validate the criteria empirically. For example, pilots trust aircraft collision avoidance systems because they knew they are based on millions of simulations (Kochenderfer et al. 2012) and these systems have an excellent track record.

However, both of these cases require us to be able to formalize our desiderata in advance, and, in the case of empirical validation, accept the cost of testing the ML system to collect data on its performance with respect to our desiderata. Unfortunately, formal definitions of auxiliary desiderata are often elusive. In such cases, explanation can be valuable to qualitatively ascertain whether desiderata such as fairness, privacy, reliability, robustness, causality, usability and trust are met. For example, one can provide a feasible explanation that fails to correspond to a causal structure, exposing a potential concern.

This observation, of interpretability as a *verification tool*, suggests that carefully thought-out work in interpretable machine learning should be able to specify *What are the downstream goals of this interpretable machine learning system?* and *Why is interpretability the right tool for achieving those goals?*

When Is Interpretability the Right Tool?

As noted above, there are many tools for verification. Not all ML systems require interpretability. Ad servers, postal code sorting, air craft collision avoidance systems—all can be evaluated without interpretable machine learning and perform their tasks without human intervention. In these cases, we have a formal guarantee of performance or evidence that the problem is sufficiently well-studied and validated in real applications that we trust the system's decision, even if the system is not perfect. In other cases, explanation is not necessary because there are no significant consequences for unacceptable results (e.g. an occasional poor book recommendation).

We argue that the need for interpretability stems from an *incompleteness* in the problem formalization, creating a fundamental barrier to optimization and evaluation. Indeed, in the psychology literature, (Keil et al. 2004) notes "explanations may highlight an incompleteness," that is, explanations can be one of ways to ensure that effects of gaps in problem formalization are visible to us.

Before continuing, we note that incompleteness is distinct from uncertainty: the fused estimate of a missile location may be uncertain, but such uncertainty can be rigorously quantified and formally reasoned about. In machine learning terms, we distinguish between cases where unknowns result in quantified variance—e.g. trying to learn from small data set or with limited sensors—and incompleteness that produces some kind of unquantified bias—e.g. the effect of including domain knowledge in a model selection process.

Below we provide some illustrative scenarios in which incomplete problem specifications are common:

- Scientific Understanding: The human's goal is to gain knowledge. We do not have a complete way of stating what knowledge is; thus the best we can do is ask for explanations we can convert into knowledge.
- Safety: For complex tasks, the end-to-end system is almost never completely testable; one cannot create a complete list of scenarios in which the system may fail. Enumerating all possible outputs given all possible inputs be computationally or logistically infeasible, and we may be unable to flag all undesirable outputs.
- Ethics: The human may want to guard against certain kinds of discrimination, and their notion of fairness may be too abstract to be completely encoded into the system (e.g., one might desire a 'fair' classifier for loan approval). Even if we can encode protections for specific protected classes into the system, there might be biases that we did not consider a priori (e.g., one may not build gender-biased word embeddings on purpose, but it was a pattern in data that became apparent only after the fact).
- Mismatched objectives: The agent's algorithm may be optimizing an incomplete objective—that is, a proxy function for the ultimate goal. For example, a clinical system may be optimized for cholesterol control, without considering the likelihood of adherence; an automotive engineer may be interested in engine data not to make predictions about engine failures but to more broadly build a better car.

- Multi-objective trade-offs: Two well-defined desiderata in ML systems may compete with each other, such as privacy and prediction quality (Hardt et al. 2016) or privacy and non-discrimination (Strahilevitz 2008). Even if each objectives are fully-specified, the exact dynamics of the trade-off may not be fully known, and the decision may have to be case-by-case.

Additional taxonomies for situations in which explanation is needed, as well as a survey of interpretable models, are reviewed in Lipton (2016). In this work, we focus on making clear that interpretability is just one tool for the verification, suited for situations in which problems are incompletely specified, and focus most of efforts on its evaluation. To expand upon our suggestion above, we suggest that research in interpretable machine learning should specify *How is the problem formulation incomplete?*

4 Evaluation

Once we know that we need an interpretable machine learning approach from Sect. 3, the next logical question is to determine how to evaluate it. Even in standard ML settings, there exists a taxonomy of evaluation that is considered appropriate. In particular, the evaluation should match the claimed contribution. Evaluation of applied work should demonstrate success in the application: a game-playing agent might beat a human player, a classifier may correctly identify star types relevant to astronomers. In contrast, core methods work should demonstrate generalizability via careful evaluation on a variety of synthetic and standard benchmarks.

In this section we lay out an analogous taxonomy of evaluation approaches for interpretability: application-grounded, human-grounded, and functionally-grounded (see Fig. 1). These range from task-relevant to general, also acknowledge that while human evaluation is essential to assessing interpretability, human-subject evaluation

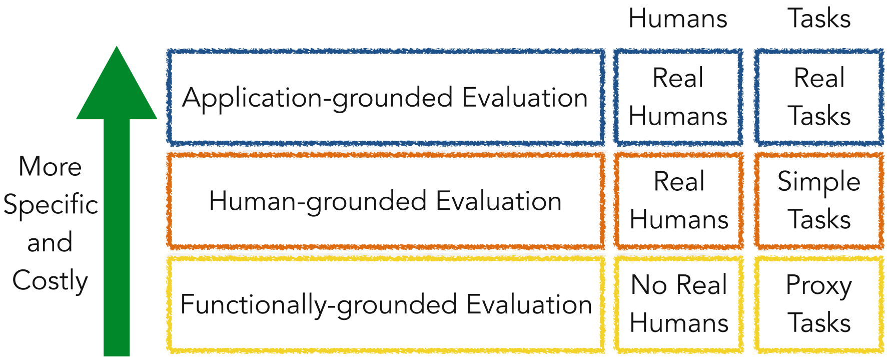


Fig. 1 Taxonomy of evaluation approaches for interpretability

is not an easy task. A human experiment needs to be well-designed to minimize confounding factors, consumed time, and other resources. We discuss the trade-offs between each type of evaluation and when each would be appropriate.

Application-Grounded Evaluation: Real Humans, Real Tasks
As mentioned in Sect. 3, interpretability is most often used a tool to verify some other objective, such as safety or nondiscrimination. Application-grounded evaluation involves conducting human experiments within a real application. If the researcher has a concrete application in mind—such as working with doctors on diagnosing patients with a particular disease—the best way to show that the model works is to evaluate it with respect to the task: doctors performing diagnoses. This reasoning aligns with the methods of evaluation common in the human-computer interaction and visualization communities, where there exists a strong ethos around making sure that the system delivers on its intended task (Antunes et al. 2012; Lazar et al. 2010). For example, a visualization for correcting segmentations from microscopy data would be evaluated via user studies on segmentation on the target image task (Suissa-Peleg et al. 2016); a homework-hint system is evaluated on whether the student achieves better post-test performance (Williams et al. 2016).

Specifically, we evaluate the quality of an explanation in the context of its end-task, such as whether it results in better identification of errors, new facts, or less discrimination. Examples of experiments include:

- Domain expert experiment with the exact application task.
- Domain expert experiment with a simpler or partial task to shorten experiment time and increase the pool of potentially-willing subjects.

In both cases, an important baseline is how well *human-produced* explanations assist in other humans trying to complete the task.

Finally, to make high impact in real world applications, it is essential that we as a community respect the time and effort involved to do such evaluations, and also demand high standards of experimental design when such evaluations are performed. As HCI community recognizes (Antunes et al. 2012), this is not an easy evaluation metric. Nonetheless, it directly tests the objective that the system is built for, and thus performance with respect to that objective gives strong evidence of success.

Human-Grounded Metrics: Real Humans, Simplified Tasks
Human-grounded evaluation is about conducting simpler human-subject experiments that maintain the essence of the target application. Such an evaluation is appealing when experiments with the target community is challenging. These evaluations can be completed with lay humans, allowing for both a bigger subject pool and less expenses, since we do not have to compensate highly trained domain experts. Human-grounded evaluation is most appropriate when one wishes to test more general notions of the quality of an explanation. For example, to study what kinds of explanations are best understood under severe time constraints, one might create abstract tasks in which other factors—such as the overall task complexity—can be controlled (Kim et al. 2013, 2014; Lakkaraju et al. 2016).

The key question, of course, is how we can evaluate the quality of an explanation without a specific end-goal (such as identifying errors in a safety-oriented task or identifying relevant patterns in a science-oriented task). Ideally, our evaluation approach will depend only on the quality of the explanation, regardless of whether the explanation is the model itself or a post-hoc interpretation of a black-box model, and regardless of the correctness of the associated prediction. Examples of potential experiments include:

- Binary forced choice: humans are presented with pairs of explanations, and must choose the one that they find of higher quality (basic face-validity test made quantitative).
- Forward simulation/prediction: humans are presented with an explanation and an input, and must correctly simulate the model's output (regardless of the true output).
- Counterfactual simulation: humans are presented with an explanation, an input, and an output, and are asked what must be changed to change the method's prediction to a desired output (and related variants).

As an example, the common intrusion-detection test (Chang et al. 2009) in topic models is a concrete form of the forward simulation/prediction task: we ask the human to find the difference between the model's true output and some corrupted output as a way to determine whether the human has correctly understood what the model's true output is.

Functionally-Grounded Evaluation: No Humans, Proxy Tasks
Functionally-grounded evaluation requires no human experiments; instead, it uses some formal definition of interpretability as a proxy for explanation quality. Such experiments are appealing because even general human-subject experiments require time and costs both to perform and to get necessary approvals (e.g., IRBs), which may be beyond the resources of a machine learning researcher. Functionally-grounded evaluations are most appropriate once we have a class of models or regularizers that have already been validated, e.g. via human-grounded experiments. They may also be appropriate when a method is not yet mature or when human subject experiments are unethical.

The challenge, of course, is to determine what proxies to use. For example, decision trees have been considered interpretable in many situations (Freitas 2014). In Sect. 5, we describe open problems in determining what proxies are reasonable. Once a proxy has been formalized, the challenge is squarely an optimization problem, as the model class or regularizer is likely to be discrete, non-convex and often non-differentiable. Examples of experiments include

- Show the improvement of prediction performance of a model that is already proven to be interpretable (assumes that someone has run human experiments to show that the model class is interpretable).
- Show that one's method performs better with respect to certain regularizers—for example, is more sparse—compared to other baselines (assumes someone has run human experiments to show that the regularizer is appropriate).

5 Considerations for Generalization

Identifying a need (Sect. 3) and being able to perform quantitative comparisons (Sect. 4) allows us to know that we are justified in our use of an interpretable machine learning approach and determine whether our approach is more interpretable than our baselines. However, we are often interested in more than just a comparison; we want insights on how our method might perform on other tasks.

For example, when it comes to the form of the explanation, Subramanian et al. (1992) found that users prefer decision trees to tables in games, whereas Huysmans et al. (2011) found users prefer, and are more accurate, with decision tables rather than other classifiers in a credit scoring domain. Hayete and Bienkowska (2004) found a preference for non-oblique splits in decision trees. When it comes to the amount of explanation, a number of human-subject studies have found that longer or more complex explanations can result in higher human accuracy and trust (Kulesza et al. 2013; Bussone et al. 2015; Allahyari and Lavesson 2011; Elomaa 2017), yet sparsity remains closely tied with interpretablity in the machine learning community (Mehmood et al. 2012; Chandrashekar and Sahin 2014) (often citing the famous seven plus or minus two rule (Miller 1956)). From this collection of results, are there ways to infer what method might perform well on a new task?

In this section, we describe a taxonomy of factors to describe contexts within interpretability is needed. These features can be used to link across experiments and the three types of evaluations, and thus being able to generalize to new problems where interpretability is needed. We also argue that a shared set of key terms for describing different interpretability contexts is essential to other researchers being able to find other methods that they should be including in their comparisons.

Task-Related Factors of Interpretability

Disparate-seeming applications may share common categories: an application involving preventing medical error at the bedside and an application involving support for identifying inappropriate language on social media might be similar in that they involve making a decision about a specific case—a patient, a post—in a relatively short period of time. However, when it comes to time constraints, the needs in those scenarios might be different from an application involving the understanding of the main characteristics of a large omics data set, where the goal—science—is much more abstract and the scientist may have hours or days to inspect the model outputs.

Below, we list a set of factors that might make tasks similar in their explanation needs:

- *Global vs. Local.* Global interpretability implies knowing what patterns are present in general (such as key features governing galaxy formation), while local interpretability implies knowing the reasons for a specific decision (such as why a particular loan application was rejected). The former may be important for when scientific understanding or bias detection is the goal; the latter when one needs a justification for a specific decision.

- *Characterization of Incompleteness.* What part of the problem formulation is incomplete, and how incomplete is it? We hypothesize that the types of explanations needed may vary depending on whether the source of concern is due to incompletely specified inputs, constraints, domains, internal model structure, costs, or even in the need to understand the training algorithm. The severity of the incompleteness may also affect explanation needs. For example, one can imagine a spectrum of questions about the safety of self-driving cars. On one end, one may have general curiosity about how autonomous cars make decisions. At the other, one may wish to check a specific list of scenarios (e.g., sets of sensor inputs that causes the car to drive off of the road by 10 cm). In between, one might want to check a general property—safe urban driving—without an exhaustive list of scenarios and safety criteria.
- *Time Constraints.* How long can the user afford to spend to understand the explanation? A decision that needs to be made at the bedside or during the operation of a plant must be understood quickly, while in scientific or anti-discrimination applications, the end-user may be willing to spend hours trying to fully understand an explanation.
- *Nature of User Expertise.* How experienced is the user in the task? The user's experience will affect what kind of *cognitive chunks* they have, that is, how they organize individual elements of information into collections (Neath and Surprenant 2003). For example, a clinician may have a notion that autism and ADHD are both developmental diseases. The nature of the user's expertise will also influence what level of sophistication they expect in their explanations. For example, domain experts may expect or prefer a somewhat larger and sophisticated model—which confirms facts they know—over a smaller, more opaque one. These preferences may be quite different from hospital ethicist who may be more narrowly concerned about whether decisions are being made in an ethical manner. More broadly, decision-makers, scientists, compliance and safety engineers, data scientists, and machine learning researchers all come with different background knowledge and communication styles.

Each of these factors an be isolated in human-grounded experiments in simulated tasks to determine which methods work best when they are present; more factors can be added if it turns out generalization within applications sharing these factors is poor. As mentioned above, these factors can also be used as key terms when searching for methods that might be relevant for a new problem.

Explanation-Related Factors of Interpretability
Just as disparate applications may share common categories, disparate explanations may share common qualities that correlate to their utility. As before, we provide a set of factors that may correspond to different explanation needs. Here, we define *cognitive chunks* to be the basic units of explanation.

- *Form of cognitive chunks.* What are the basic units of the explanation? Are they raw features? Derived features that have some semantic meaning to the expert (e.g. "neurological disorder" for a collection of diseases or "chair" for a collection of pixels)? Prototypes?

- *Number of cognitive chunks.* How many cognitive chunks does the explanation contain? How does the quantity interact with the type: for example, a prototype can contain a lot more information than a feature; can we handle them in similar quantities?
- *Level of compositionality.* Are the cognitive chunks organized in a structured way? Rules, hierarchies, and other abstractions can limit what a human needs to process at one time. For example, part of an explanation may involve *defining* a new unit (a chunk) that is a function of raw units, and then providing an explanation in terms of that new unit.
- *Monotonicity and other interactions between cognitive chunks.* Does it matter if the cognitive chunks are combined in linear or nonlinear ways? In monotone ways (Gupta et al. 2016)? Are some functions more natural to humans than others (Wilson et al. 2015; Schulz et al. 2016)?
- *Uncertainty and stochasticity.* How well do people understand uncertainty measures? To what extent is stochasticity understood by humans?

Identifying methods by their characteristics will also make it easier to search for general properties of high-quality explanation that span across multiple methods, and facilitate meta-analyses that study whether these factors are associated with deeper interpretability-related universals. Ultimately, we would hope to discover that certain task-related properties benefit from explanations with certain explanation-specific properties.

6 Conclusion: Recommendations for Researchers

In this work, we have laid the groundwork for a process performing rigorous science in interpretability: defining the need; careful evaluation; and defining factors for generalization. While there are many open questions, this framework can help ensure that our research outputs in this field are evidence-based and generalizable. Below, we summarize our recommendations.

The claim of the research should match the type of the evaluation. Just as one would be critical of a reliability-oriented paper that only cites accuracy statistics, the choice of evaluation should match the specificity of the claim being made. A contribution that is focused on a particular application should be expected to be evaluated in the context of that application (application-grounded evaluation), or on a human experiment with a closely-related task (human-grounded evaluation). A contribution that is focused on better optimizing a model class for some definition of interpretability should be expected to be evaluated with functionally-grounded metrics. As a community, we must be careful in the work on interpretability, both recognizing the need for and the costs of human-subject experiments. We should also make sure that these evaluations are on problems where there is a need for interpretability.

We should categorize our applications and methods with a common taxonomy. In Sect. 5, we hypothesized factors that may be the factors of interpretability. Creating a shared language around such factors is essential not only to evaluation, but also for the citation and comparison of related work. For example, work on creating a safe healthcare agent might be framed as focused on the need for explanation due to unknown inputs at the local scale, evaluated at the level of an application. In contrast, work on learning sparse linear models might also be framed as focused on the need for explanation due to unknown inputs, but this time evaluated at global scale. As we share each of our work with the community, we can do each other a service by describing factors such as

1. What is the ultimate verification (or other) goal? How is the problem formulation incomplete? (Sect. 3)
2. At what level is the evaluation being performed? (Sect. 4)
3. What are the task-related and explanation-related factors in the experiments? (Sect. 5)

These considerations should move us away from vague claims about the interpretability of a particular model and toward classifying applications by a common set of generalizable terms.

Acknowledgements This piece would not have been possible without the dozens of deep conversations about interpretability with machine learning researchers and domain experts. Our friends and colleagues, we appreciate your support. We want to particularity thank Ian Goodfellow, Kush Varshney, Hanna Wallach, Solon Barocas, Stefan Rüping and Jesse Johnson for their feedback.

References

Adler P, Falk C, Friedler SA, Rybeck G, Scheidegger C, Smith B, Venkatasubramanian S (2016) Auditing black-box models for indirect influence. In: Data Mining (ICDM), 2016 IEEE 16th International Conference on, IEEE, pp 1–10

Allahyari H, Lavesson N (2011) User-oriented assessment of classification model understandability. In: 11th scandinavian conference on Artificial intelligence, IOS Press

Amodei D, Olah C, Steinhardt J, Christiano P, Schulman J, Mané D (2016) Concrete problems in AI safety. arXiv preprint arXiv:160606565

Antunes P, Herskovic V, Ochoa SF, Pino JA (2012) Structuring dimensions for collaborative systems evaluation. In: ACM Computing Surveys, ACM

Bechtel W, Abrahamsen A (2005) Explanation: A mechanist alternative. Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences

Bostrom N, Yudkowsky E (2014) The ethics of artificial intelligence. The Cambridge Handbook of Artificial Intelligence

Buciluă C, Caruana R, Niculescu-Mizil A (2006) Model compression. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM

Bussone A, Stumpf S, O'Sullivan D (2015) The role of explanations on trust and reliance in clinical decision support systems. In: Healthcare Informatics (ICHI), 2015 International Conference on, IEEE, pp 160–169

Carton S, Helsby J, Joseph K, Mahmud A, Park Y, Walsh J, Cody C, Patterson CE, Haynes L, Ghani R (2016) Identifying police officers at risk of adverse events. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM

Caruana R, Lou Y, Gehrke J, Koch P, Sturm M, Elhadad N (2015) Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp 1721–1730

Chandrashekar G, Sahin F (2014) A survey on feature selection methods. Computers & Electrical Engineering 40(1):16–28

Chang J, Boyd-Graber JL, Gerrish S, Wang C, Blei DM (2009) Reading tea leaves: How humans interpret topic models. In: NIPS

Chater N, Oaksford M (2006) Speculations on human causal learning and reasoning. Information sampling and adaptive cognition

Doshi-Velez F, Ge Y, Kohane I (2014) Comorbidity clusters in autism spectrum disorders: an electronic health record time-series analysis. In: Pediatrics, Am Acad Pediatrics, vol 133:1, pp e54–e63

Doshi-Velez F, Wallace B, Adams R (2015) Graph-sparse lda: a topic model with structured sparsity. In: Association for the Advancement of Artificial Intelligence

Dwork C, Hardt M, Pitassi T, Reingold O, Zemel R (2012) Fairness through awareness. In: Innovations in Theoretical Computer Science Conference, ACM

Elomaa T (2017) In defense of c4. 5: Notes on learning one-level decision trees. ML-94 254:62

Freitas A (2014) Comprehensible classification models: a position paper. In: ACM SIGKDD Explorations

Glennan S (2002) Rethinking mechanistic explanation. Philosophy of science

Goodman B, Flaxman S (2016) European union regulations on algorithmic decision-making and a "right to explanation". arXiv preprint arXiv:160608813

Gupta M, Cotter A, Pfeifer J, Voevodski K, Canini K, Mangylov A, Moczydlowski W, Van Esbroeck A (2016) Monotonic calibrated interpolated look-up tables. In: Journal of Machine Learning Research

Hamill S (2017) CMU computer won poker battle over humans by statistically significant margin. http://www.post-gazette.com/business/tech-news/2017/01/31/CMU-computer-won-poker-battle-over-humans-by-statistically-significant-margin/stories/201701310250, accessed: 2017-02-07

Hardt M, Talwar K (2010) On the geometry of differential privacy. In: ACM Symposium on Theory of Computing, ACM

Hardt M, Price E, Srebro N (2016) Equality of opportunity in supervised learning. In: Advances in Neural Information Processing Systems

Hayete B, Bienkowska JR (2004) Gotrees: Predicting go associations from proteins. Biocomputing 2005 p 127

Hempel C, Oppenheim P (1948) Studies in the logic of explanation. Philosophy of science

Hughes MC, Elibol HM, McCoy T, Perlis R, Doshi-Velez F (2016) Supervised topic models for clinical interpretability. In: arXiv preprint arXiv:1612.01678

Huysmans J, Dejaeger K, Mues C, Vanthienen J, Baesens B (2011) An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. In: DSS, Elsevier

Keil F (2006) Explanation and understanding. Annu Rev Psychol

Keil F, Rozenblit L, Mills C (2004) What lies beneath? understanding the limits of understanding. Thinking and seeing: Visual metacognition in adults and children

Kim B, Chacha C, Shah J (2013) Inferring robot task plans from human team meetings: A generative modeling approach with logic-based prior. Association for the Advancement of Artificial Intelligence

Kim B, Rudin C, Shah J (2014) The Bayesian Case Model: A generative approach for case-based reasoning and prototype classification. In: NIPS

Kim B, Glassman E, Johnson B, Shah J (2015a) iBCM: Interactive bayesian case model empowering humans via intuitive interaction. In: MIT-CSAIL-TR-2015-010

Kim B, Shah J, Doshi-Velez F (2015b) Mind the gap: A generative approach to interpretable feature selection and extraction. In: Advances in Neural Information Processing Systems

Kindermans PJ, Schütt KT, Alber M, Müller KR, Dähne S (2017) Patternnet and patternlrp–improving the interpretability of neural networks. arXiv preprint arXiv:170505598

Kochenderfer MJ, Holland JE, Chryssanthacopoulos JP (2012) Next-generation airborne collision avoidance system. Tech. rep., Massachusetts Institute of Technology-Lincoln Laboratory Lexington United States

Krakovna V, Doshi-Velez F (2016) Increasing the interpretability of recurrent neural networks using hidden markov models. In: arXiv preprint arXiv:1606.05320

Kulesza T, Stumpf S, Burnett M, Yang S, Kwan I, Wong WK (2013) Too much, too little, or just right? ways explanations impact end users' mental models. In: Visual Languages and Human-Centric Computing (VL/HCC), 2013 IEEE Symposium on, IEEE, pp 3–10

Lakkaraju H, Bach SH, Leskovec J (2016) Interpretable decision sets: A joint framework for description and prediction. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp 1675–1684

Lazar J, Feng JH, Hochheiser H (2010) Research methods in human-computer interaction. John Wiley & Sons

Lei T, Barzilay R, Jaakkola T (2016) Rationalizing neural predictions. arXiv preprint arXiv:160604155

Lipton ZC (2016) The mythos of model interpretability. arXiv preprint arXiv:160603490

Liu W, Tsang IW (2016) Sparse perceptron decision tree for millions of dimensions. In: AAAI, pp 1881–1887

Lombrozo T (2006) The structure and function of explanations. Trends in cognitive sciences 10(10):464–470

Lou Y, Caruana R, Gehrke J (2012) Intelligible models for classification and regression. In: ACM SIGKDD international conference on Knowledge discovery and data mining, ACM

Mehmood T, Liland KH, Snipen L, Sæbø S (2012) A review of variable selection methods in partial least squares regression. Chemometrics and Intelligent Laboratory Systems 118:62–69

Miller GA (1956) The magical number seven, plus or minus two: Some limits on our capacity for processing information. The Psychological Review (2):81–97

Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, Riedmiller M (2013) Playing atari with deep reinforcement learning. In: arXiv preprint arXiv:1312.5602

Neath I, Surprenant A (2003) Human Memory. Wadsworth Cengage Learning

Otte C (2013) Safe and interpretable machine learning: A methodological review. In: Computational Intelligence in Intelligent Data Analysis, Springer

Parliament, of the European Union C (2016) General data protection regulation

Ribeiro MT, Singh S, Guestrin C (2016) "why should i trust you?": Explaining the predictions of any classifier. In: arXiv preprint arXiv:1602.04938

Ross A, Hughes MC, Doshi-Velez F (2017) Right for the right reasons: Training differentiable models by constraining their explanations. In: International Joint Conference on Artificial Intelligence

Ruggieri S, Pedreschi D, Turini F (2010) Data mining for discrimination discovery. ACM Transactions on Knowledge Discovery from Data

Rüping S (2006) Thesis: Learning interpretable models. PhD thesis, Universitat Dortmund

Safavian SR, Landgrebe D (1991) A survey of decision tree classifier methodology. IEEE transactions on systems, man, and cybernetics 21(3):660–674

Schulz E, Tenenbaum J, Duvenaud D, Speekenbrink M, Gershman S (2016) Compositional inductive biases in function learning. In: bioRxiv, Cold Spring Harbor Labs Journals

Sculley D, Holt G, Golovin D, Davydov E, Phillips T, Ebner D, Chaudhary V, Young M, Crespo JF, Dennison D (2015) Hidden technical debt in machine learning systems. In: Advances in Neural Information Processing Systems

Selvaraju RR, Das A, Vedantam R, Cogswell M, Parikh D, Batra D (2016) Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. arXiv preprint arXiv:161002391

Shrikumar A, Greenside P, Shcherbina A, Kundaje A (2016) Not just a black box: Interpretable deep learning by propagating activation differences. ICML

Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, et al (2016) Mastering the game of go with deep neural networks and tree search. In: Nature, Nature Publishing Group

Singh S, Ribeiro MT, Guestrin C (2016) Programs as black-box explanations. arXiv preprint arXiv:161107579

Smilkov D, Thorat N, Kim B, Viégas F, Wattenberg M (2017) Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:170603825

Strahilevitz LJ (2008) Privacy versus antidiscrimination. University of Chicago Law School Working Paper

Subramanian GH, Nosek J, Raghunathan SP, Kanitkar SS (1992) A comparison of the decision table and tree. Communications of the ACM 35(1):89–94

Suissa-Peleg A, Haehn D, Knowles-Barley S, Kaynig V, Jones TR, Wilson A, Schalek R, Lichtman JW, Pfister H (2016) Automatic neural reconstruction from petavoxel of electron microscopy data. In: Microscopy and Microanalysis, Cambridge Univ Press

Toubiana V, Narayanan A, Boneh D, Nissenbaum H, Barocas S (2010) Adnostic: Privacy preserving targeted advertising

Ustun B, Rudin C (2016) Supersparse linear integer models for optimized medical scoring systems. Machine Learning 102(3):349–391

Varshney K, Alemzadeh H (2016) On the safety of machine learning: Cyber-physical systems, decision sciences, and data products. In: CoRR

Wang F, Rudin C (2015) Falling rule lists. In: Artificial Intelligence and Statistics, pp 1013–1022

Wang T, Rudin C, Doshi-Velez F, Liu Y, Klampfl E, MacNeille P (2017) Bayesian rule sets for interpretable classification. In: International Conference on Data Mining

Williams JJ, Kim J, Rafferty A, Maldonado S, Gajos KZ, Lasecki WS, Heffernan N (2016) Axis: Generating explanations at scale with learnersourcing and machine learning. In: ACM Conference on Learning@ Scale, ACM

Wilson A, Dann C, Lucas C, Xing E (2015) The human kernel. In: Advances in Neural Information Processing Systems

Explanation Methods in Deep Learning: Users, Values, Concerns and Challenges

Gabriëlle Ras, Marcel van Gerven, and Pim Haselager

Abstract Issues regarding explainable AI involve four components: users, laws and regulations, explanations and algorithms. Together these components provide a context in which explanation methods can be evaluated regarding their adequacy. The goal of this chapter is to bridge the gap between expert users and lay users. Different kinds of users are identified and their concerns revealed, relevant statements from the General Data Protection Regulation are analyzed in the context of Deep Neural Networks (DNNs), a taxonomy for the classification of existing explanation methods is introduced, and finally, the various classes of explanation methods are analyzed to verify if user concerns are justified. Overall, it is clear that (visual) explanations can be given about various aspects of the influence of the input on the output. However, it is noted that explanation methods or interfaces for lay users are missing and we speculate which criteria these methods/interfaces should satisfy. Finally it is noted that two important concerns are difficult to address with explanation methods: the concern about bias in datasets that leads to biased DNNs, as well as the suspicion about unfair outcomes.

Keywords Explanation methods · Explainable AI · Interpretability · Deep neural networks · Artificial intelligence

1 Introduction

Increasingly, Artificial Intelligence (AI) is used in order to derive actionable outcomes from data (e.g. categorizations, predictions, decisions). The overall goal of this chapter is to bridge the gap between expert users and lay users, highlighting the explanation needs of both sides and analyzing the current state of explainability.

G. Ras (✉) · M. van Gerven · P. Haselager
Radboud University, Nijmegen, The Netherlands
e-mail: g.ras@donders.ru.nl; marcel.vangerven@donders.ru.nl; w.haselager@donders.ru.nl

H. J. Escalante et al. (eds.), *Explainable and Interpretable Models in Computer Vision and Machine Learning*, The Springer Series on Challenges in Machine Learning, https://doi.org/10.1007/978-3-319-98131-4_2

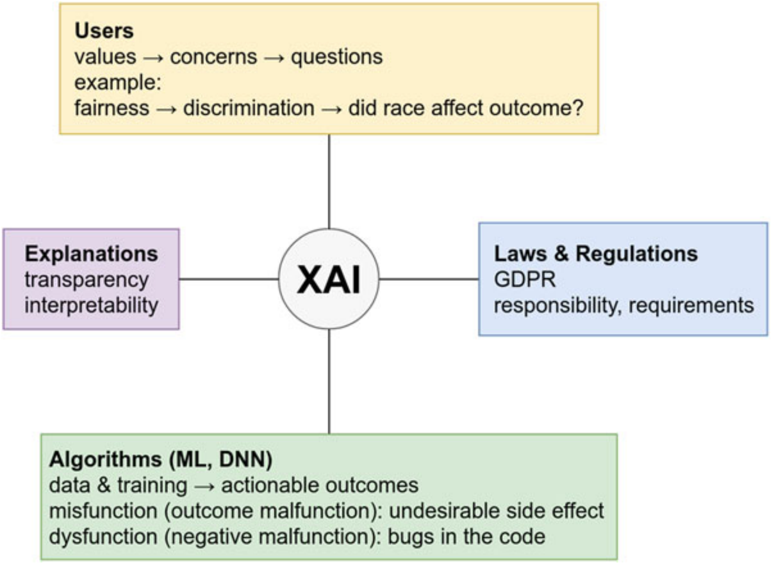

Users
values → concerns → questions
example:
fairness → discrimination → did race affect outcome?
Explanations
transparency
interpretability
XAI
Laws & Regulations
GDPR
responsibility, requirements
Algorithms (ML, DNN)
data & training → actionable outcomes
misfunction (outcome malfunction): undesirable side effect
dysfunction (negative malfunction): bugs in the code

Fig. 1 Issues regarding explainable DNNs involve (at least) four components: users, algorithms, laws and explanations. Together these components provide a context in which explanations can be evaluated regarding their adequacy

We do this by taking a more detailed look at each component mentioned above and in Fig. 1. Finally we address some concerns in the context of DNNs.

1.1 The Components of Explainability

Issues regarding explainable AI (XAI) involve (at least) four components: users, laws and regulations, explanations and algorithms. Together these components provide a context in which explanation methods can be evaluated regarding their adequacy. We briefly discuss these components in Fig. 1.

1.2 Users and Laws

AI has a serious impact on society, due to the large scale adoption of digital automation techniques that involve information processing and prediction. Deep Neural Networks (DNNs) belong to this set of automation techniques and are used increasingly because of their capability to extract meaningful patterns from raw input. DNNs are fed large quantities of digital information that are easily collected from users. Currently there is much debate regarding the safety of and trust in data processes in general, leading to investigations regarding the explainability of AI-supported decision making. The level of concern about these topics is reflected by official regulations such as the General Data Protection Regulation[1] (GDPR), also mentioned in Doshi-Velez and Kim (2017), Holzinger et al. (2017a), incentives to promote the field of explainability (Gunning 2017) and institutional initiatives to ensure the safe development of AI such as OpenAI. As the technology becomes more widespread, DNNs in particular, the dependency on said technology increases and ensuring trust in DNN technology becomes a necessity. Current DNNs are achieving unparalleled performance in areas of Computer Vision (CV) and Natural Language Processing (NLP). They are also being used in real-world applications in e.g. medical imaging (Lee et al. 2017), autonomous driving (Bojarski et al. 2017) and legislation (Lockett et al. 2018).

1.3 Explanation and DNNs

The challenge with DNNs in particular lies in providing insight into the processes leading to their outcomes, and thereby helping to clarify under which circumstances they can be trusted to perform as intended and when they cannot. Unlike other methods in Machine Learning (ML), such as decision trees or Bayesian networks, an explanation for a certain decision made by a DNN cannot be retrieved by simply scrutinizing the inference process. The learned internal representations and the flow of information through the network are hard to analyze: As architectures get deeper, the number of learnable parameters increases. It is not uncommon to have networks with millions of parameters. Furthermore, network architecture is determined by various components (unit type, activation function, connectivity pattern, gating mechanisms) and the result of a complex learning procedure, which itself depends on various properties (regularization, adaptive mechanisms, employed cost function). The net result of the interaction between these components cannot be predicted in advance. Because of these complications, DNNs are often called *black box models*, as opposed to glass-box models (Holzinger et al. 2017b). Fortunately, these problems have not escaped the attention of the ML/Deep Learning (DL) community (Zeng 2016; Samek et al. 2017; Seifert et al. 2017; Olah et al. 2017; Hall et al. 2017; Montavon et al. 2018; Marcus 2018; Doshi-Velez and Kim 2017). Research on how to interpret and explain the decision process of Artificial Neural Networks (ANNs) has been going on since the late 1980s (Elman 1989; Andrews et al. 1995). The objective of explanation methods is to make specific aspects of a DNN's internal representations and information flow interpretable by humans.

[1] https://www.eugdpr.org

2 Users and Their Concerns

Various kinds of DNN users can be distinguished. Users entertain certain values; these include ethical values such as fairness, neutrality, lawfulness, autonomy, privacy or safety, or functional values such as accuracy, usability, speed or predictability. Out of these values certain concerns regarding DNNs may arise, e.g. apprehensions about discrimination or accuracy. These concerns get translated into questions about the system, e.g. "did the factor "race" influence the outcome of the system" or "how reliable was the data used?" In this section we identify at least two general types of users: the expert users and the lay users, that can be further categorized into six specific kinds of users. Note that there could be (and there regularly is) overlap between the users described below, such that a particular user can be classified as belonging to more than one of the categories.

1. **Expert users** are the system builders and/or modifiers that have direct influence on the implementation of the network. Two kinds of experts can be identified:

   (a) **DNN engineers** are generally researchers involved in extending the field and have detailed knowledge about the mathematical theories and principles of DNNs. DNN engineers are interested in explanations of a functional nature, e.g. the effects of various hyperparameters on the performance of the network or methods that can be used for model debugging.

   (b) **DNN developers** are generally application builders who make software solutions that can be used by lay people. DNN developers often make use of off-the-shelf DNNs, often re-training the DNN along with tuning certain hyperparameters and integrating them with various software components, resulting in a functional application. The DNN developer is concerned with the goals of the overall application and assesses whether they have been met by the DNN solution. DNN developers are interested in explanation methods that allow them to understand the behavior of the DNN in the various use cases of the integrated software application.

2. **Lay users** do not and need not have knowledge of how the DNN was implemented and the underlying mathematical principles, nor do they require knowledge of how the DNN was integrated with other software components resulting in a final functional application. At least four lay users are identified:

   (a) **The owner** of the software application in which the DNN is embedded. The owner is usually an entity that acquires the application for possible commercial, practical or personal use. For example, an owner can be an organization (e.g. a hospital or a car manufacturer) that purchases the application for end users (e.g. employees (doctors) or clients (car buyers)), but the owner can also be a consumer that purchases the application for personal use. In the latter case the categorization of owner fully overlaps with the next category of users which are the end users. The owner is concerned with explainability questions about the capabilities of the application, e.g. justification of a prediction or a prediction given the input data, and aspects of accountability, e.g. to what extent can application malfunction be attributed to the DNN component?

(b) **The end user** for whom the application was intended to be used by. The end user uses the application as part of their profession or for personal use. The end user is concerned with explainability about the capabilities of the application, e.g. justification of a prediction given the input data, and explainability regarding the behavior of the application, e.g. why does the application not do what it was advertised to do?

(c) **The data subject** is the entity whose information is being processed by the application or the entity which is directly affected by the application outcome. An outcome is the output of the application in the context of the use case. Sometimes the data subject is the same entity as the end user, for example in the case that the application is meant for personal use. The data subject is mostly concerned with the ethical and moral aspects that result from the actionable outcomes. An actionable outcome is an outcome that has consequences or an outcome on which important decisions are based.

(d) **Stakeholders** are people or organizations without a direct connection to either the development, use or outcome of the application and who can reasonably claim an interest in the process, for instance when its use runs counter to particular values they protect. Governmental and non-governmental organizations may put forward legitimate information requests regarding the operations and consequences of DNNs. Stakeholders are often interested in the ethical and legal concerns raised in any phase of the process.

2.1 Case Study: Autonomous Driving

In this section the different users are presented in the context of a self-driving car.

1. The DNN engineer creates a DL solution to the problem of object segmentation and object classification by experimenting with various types of networks. Given raw video input the DL solution gives the output of the type of object and the location of the object in the video.
2. The DNN developer creates a planning system which integrates the output of the DL solution with other components in the system. The planning system decides which actions the car will take.
3. The owner acquires the planning system and produces a car in which the planning system is operational.
4. The end user purchases the car and uses the car to travel from point A to point B.
5. The data subjects are all the entities from which information is captured along the route from point A to point B: pedestrians, private property such as houses, other cars.

6. The stakeholders are governmental institutions which formulate laws regulating the use of autonomous vehicles, or insurance companies that have to assess risk levels and their consequences.

3 Laws and Regulations

An important initiative within the European Union is the GDPR that was approved on April 14, 2016, and became enforceable on May 25, 2018. The GDPR distinguishes between personal data, data subjects, data processors and data controllers (Article 4, Definitions, Paragraphs 1, 7 and 8). *Personal data* is defined as "any information relating to an identified or identifiable natural person (data subject)". A *data processor* is the natural or legal person, public authority, agency or other body which processes data on behalf of the *data controller*, who determines the purposes, conditions and means of the processing. Hence, the DNN can function as a tool to be used by the data processor, whereas owners or end users can fill the role of data controllers.

The GDPR focuses in part on profiling: "any form of automated processing of personal data consisting of the use of personal data to evaluate certain personal aspects relating to a natural person, in particular to analyse or predict aspects concerning that natural person's performance at work, economic situation, health, personal preferences, interests, reliability, behaviour, location or movements" (Article 4, Definitions, Paragraph 4). According to articles 13, 14 and 15, when personal data is collected from a data subject for automated decision-making, the data subject has the right to access, and the data controller is obliged to provide, "meaningful information about the logic involved." Article 12 stipulates that the provision of information to data subjects should be in "concise, transparent, intelligible and easily accessible form, using clear and plain language."

4 Explanation

The right to meaningful information translates into the demand that actionable outcomes of DNNs need to be explained, i.e. be made transparent, interpretable or comprehensible to humans. Transparency refers to the extent to which an explanation makes a specific outcome understandable to a particular (group of) users. Understanding, in this context, amounts to a person grasping how a particular outcome was reached by the DNN. Note that this need not imply agreeing with the conclusion, i.e. accepting the outcome as valid or justified. In general, transparency may be considered as recommendable, leading to e.g. a greater (societal) sense of control and acceptance of ML applications. Transparency is normally also a precondition for accountability: i.e. the extent to which the responsibility for the actionable outcome can be attributed to legally (or morally) relevant agents (governments, companies, expert or lay users, etc.). However, transparency may also have negative consequences, e.g. regarding privacy or by creating possibilities for manipulation (of data, processing or training).

In relation to the (perceived) need for explanation, two reasons for investigation stand out in particular. First, a DNN may appear to dysfunction, i.e. fail to operate as intended, e.g. through bugs in the code (process malfunction). Second, it may misfunction, e.g. by producing unintended or undesired (side-)effects (Floridi et al. 2015; Mittelstadt et al. 2016) that are deemed to be societally or ethically unacceptable (outcome malfunction). Related to dysfunction is a first category of explanations. This category is based on the information necessary in order to understand the system's basic processes, e.g. to assess whether it is functioning properly, as intended, or whether it dysfunctions (e.g. suboptimal or erroneous results). This type of explanation is normally required by DNN developers and expert users. The information is used to interpret, predict, monitor, diagnose, improve, debug or repair the functioning of a system (Weller 2017).

Once an application is made available to non-expert users, normally certain guarantees regarding the system's proper functioning are in place. Generally speaking, owners, end users, data subjects and stakeholders are more interested in a second category of explanations, where suspicions about a DNN's misfunctioning (undesired outcomes) leads to requests for "local explanations". Users may request information about how a particular outcome was reached by the DNN, which aspects of input data, which learning factors or other parameters of the system influenced its decision or prediction. This information is then used to assess the appropriateness of the outcome in relation to the concerns and values of users (Doran et al. 2017; Wachter et al. 2017; Doshi-Velez et al. 2017; Weller 2017). The aim of local explanations is to strengthen the confidence and trust of users that the system is not (or will not be) conflicting with their values, i.e. that it does not violate fairness or neutrality. Note that this implies that the offered explanations should match (within certain limits) the particular user's capacity for understanding (Doshi-Velez and Kim 2017), as indicated by the GDPR.

5 Explanation Methods

So far the users, the GDPR, and the role of explanations have been discussed. To bridge the gap from that area to the more technical area of explanation methods, we need to be able to evaluate the capabilities of existing methods, in the context of the users and their needs. We bridge the gap in two ways. First, we identify, on a high level, desirable properties of explanation methods. Second, we introduce a taxonomy to categorize all types of explanation methods and third, assess the presence of the desirable properties in the categories in our taxonomy.

5.1 Desirable Properties of Explainers

Based on a survey of the literature, we arrive at the following properties which any *explainer* should have:

1. **High Fidelity** The degree to which the interpretation method agrees with the input-output mapping of the DNN. This term appears in Arbatli and Akin (1997), Markowska-Kaczmar and Wnuk-Lipiński (2004), Zilke et al. (2016), Ribeiro et al. (2016a), Ribeiro et al. (2016b), Andrews et al. (1995), and Lakkaraju et al. (2017). Fidelity is arguably the most important property that an explanation model should possess. If an explanation method is not faithful to the original model then it cannot give valid explanations because the input-output mapping is incorrect.
2. **High Interpretabiliy** To what extent a user is able to obtain true insight into how actionable outcomes are obtained. We distinguish interpretability into the following two subproperties:

   (a) **High Clarity** The degree to which the resulting explanation is unambiguous. This property is extremely important in safety-critical applications (Andrews et al. 1995) where ambiguity is to be avoided. Lakkaraju et al. (2017) introduces a quantifiable measure of clarity (unambiguity) for their method.

   (b) **High Parsimony** This refers to the complexity of the resulting explanation. An explanation that is parsimonious is a simple explanation. This concept is generally related to Occam's razor and in the case of explaining DNNs the principle is also of importance. The optimal degree of parsimony can in part be dependent on the user's capabilities.

3. **High Generalizability** The range of architectures to which the explanation method can be applied. This increases the usefulness of the explanation method. Methods that are model-agnostic (Ribeiro et al. 2016b) are the highest in generalizability.
4. **High Explanatory Power** In this context this means how many phenomena the method can explain. This roughly translates to how many different kinds of questions the method can answer. Previously in Sect. 2 we have identified a number of questions that users may have. It is also linked to the notion that the explainer should be able to take a global perspective (Ribeiro et al. 2016b), in the sense that it can explain the behaviour of the model rather than only accounting for individual predictions.

5.2 A Taxonomy for Explanation Methods

Over a relatively short period of time a plethora of explanation methods and strategies have come into existence, driven by the need of expert users to analyze and debug their DNNs. However, apart from a non-exhaustive overview of existing methods (Montavon et al. 2018) and classification schemes for purely visual methods (Grün et al. 2016; Seifert et al. 2017; Zeng 2016; Kindermans et al. 2017), little is known about efforts to rigorously map the landscape of explanation methods and isolate the underlying patterns that guide explanation methods. In this section a taxonomy for explanation methods is proposed. Three main classes of explanation methods are identified and their features described. The taxonomy was derived by analyzing the historical and contemporary trends surrounding the topic of interpretation of DNNs and explainable AI. We realize that we cannot foresee the future developments of DNNs and their explainability methods. As such it is possible that in the future the taxonomy needs to be modified. We propose the following taxonomy:

Rule-extraction methods

Extract rules that approximate the decision-making process in a DNN by utilizing the input and output of the DNN.

Attribution methods

Measure the importance of a component by changing to the input or internal components and recording how much the changes affect model performance. Methods known by other names that fall in this category are occlusion, perturbation, erasure, ablation and influence. Attribution methods are often visualized and sometimes referred to as visualization methods.

Intrinsic methods

Aim to improve the interpretability of internal representations with methods that are part of the DNN architecture. Intrinsic methods increase fidelity, clarity and parsimony in attribution methods.

In the following subsections we will describe the main features of each class and give examples from current research.

5.2.1 Rule-Extraction Methods

Rule-extraction methods extract human interpretable rules that approximate the decision-making process in a DNN. Older genetic algorithm based rule extraction methods for ANNs can be found in Andrews et al. (1995), Arbatli and Akin (1997), and Lu et al. (2006). Andrews et al. (1995) specify three categories of rule extraction methods:

Decompositional approach

Decomposition refers to breaking down the network into smaller individual parts. For the decompositional approach, the architecture of the network and/or its outputs are used in the process. Zilke et al. (2016) uses a decompositional algorithm that extracts rules for each layer in the DNN. These rules are merged together in a final merging step to produce a set of rules that describe the network behaviour by means of its inputs. Murdoch and Szlam (2017) succeeded in extracting rules from an LSTM by applying a decompositional approach.

Pedagogical approach

Introduced by Craven and Shavlik (1994) and named by Andrews et al. (1995) the pedagogical approach involves "viewing rule extraction as a learning task where the target concept is the function computed by the network and the input features are simply the network's input features" (Craven and Shavlik 1994). The pedagogical approach has the advantage that it is inherently model-agnostic. Recent examples are found in Ribeiro et al. (2016a) and Lakkaraju et al. (2017).

Eclectic approach

According to Andrews et al. (1995) "membership in this category is assigned to techniques which utilize knowledge about the internal architecture and/or weight vectors in the trained artificial neural network to complement a symbolic learning algorithm."

In terms of fidelity, local explanations are more faithful than global explanations. For rule-extraction this means that rules that govern the result of a specific input, or a neighborhood of inputs are more faithful than rules that govern all possible inputs. Rule extraction is arguably the most interpretable category of methods in our taxonomy considering that the resulting set of rules can be unambiguously be interpreted by a human being as a kind of formal language. Therefore we can say that it has a high degree of clarity. In terms of parsimony we can say that if the ruleset is "small enough" the parsimony is higher than when the ruleset is "too large". What determines "small enough" and "too large" is difficult to quantify formally and is also dependent on the user (expert vs. lay). In terms of generalizability it can go both ways: if a decompositional approach is used it is likely that the method is not generalizable, while if a pedagogical approach is used the method is highly generalizable. In terms of explanatory power, rule-extraction methods can (1) validate whether the network is working as expected in terms of overall logic flow, and (2) explain which aspects of the input data had an effect that lead to the specific output.

5.2.2 Attribution Methods

Attribution, a term introduced by Ancona et al. (2018), also referred to as relevance (Bach et al. 2015; Binder et al. 2016; Zintgraf et al. 2017; Robnik-Šikonja and Kononenko 2008), contribution (Shrikumar et al. 2017), class saliency (Simonyan et al. 2013) or influence (Kindermans et al. 2016; Adler et al. 2016; Koh and Liang 2017), aims to reveal components of high importance in the input to the DNN and their effect as the input is propagated through the network. Because of this property we can categorize the following methods to the attribution category: occlusion (Guçlütürk et al. 2017), erasure (Li et al. 2016), perturbation (Fong and Vedaldi 2017), adversarial examples (Papernot et al. 2017) and prediction difference analysis (Zintgraf et al. 2017). Other methods that belong to this category are found in Baehrens et al. (2010), Murdoch et al. (2018), and Ribeiro et al. (2016b). It is worth mentioning that attribution methods do not only apply to image input but also to other forms of input, such as text processing by LSTMs (Murdoch et al. 2018). The definition of attribution methods in this chapter is similar to that of saliency methods (Kindermans et al. 2017), but more general than the definition of attribution methods in Kindermans et al. (2017) akin to the definition in Ancona et al. (2018).

The majority of explanation methods for DNNs visualize the information obtained by attribution methods. Visualization methods were popularized by Erhan et al. (2009), Simonyan et al. (2013), Zeiler and Fergus (2014) in recent years and are concerned with how the important features are visualized. Zeng (2016) identifies that current methods focus on three aspects of visualization: feature visualization, relationship visualization and process visualization. Overall visualization methods are very intuitive methods to gain a variety of insight about a DNN decision process on many levels including architecture assessment, model quality assessment and even user feedback integration, e.g. Olah et al. (2018) create intuitive visualization interfaces for image processing DNNs.

Kindermans et al. (2017) has shown recently that attribution methods "lack reliability when the explanation is sensitive to factors that do not contribute to the model prediction." Furthermore they introduce the notion of *input invariance* as a prerequisite for accurate attribution. In other words, if the attribution method does not satisfy input invariance, we can consider it to have low fidelity. In terms of clarity, there is a degree of ambiguity that is inherent with these methods because visual explanations can be interpreted in multiple ways by different users, even by users in the same user category. In contrast to the precise results of rule-extraction methods, the information that results from attribution methods has less structure. In addition, the degree of clarity is dependent on the degree of fidelity of the method: low fidelity can cause incorrect attribution, resulting in noisy output with distracting attributions that increase ambiguity. The degree of parsimony depends on the method of visualization itself. Methods that visualize only the significant attributions exhibit a higher degree of parsimony. The degree of generalizability depends on which components are used to determine attribution. Methods that only use the input and output are inherently model agnostic, resulting in the highest degree of generalizability. Following this logic, methods that make use of internal components are generalizable to the degree that other models share these components. For example, deconvolutional networks (Zeiler et al. 2010) can be applied to models that make use of convolutions to extract features from input images. In terms of explanatory power, this class of methods can reflect intuitively with visual explanations which factors in the input dimension had a significant impact on the output of the DNN. However these methods do not explain the reason for the importance of the particular factor attribution.

5.2.3 Intrinsic Methods

The previous categories are designed to make explainable some aspects of a DNN in a process separate from training the DNN. In contrast, this category aims to improve the interpretability of internal representations with methods that are part of the DNN architecture, e.g. as part of the loss function (Dong et al. 2017b,a), modules that add additional capabilities (Santoro et al. 2017; Palm et al. 2017), or as part of the architecture structure, in terms of operations between layers (Li et al. 2017; Wu et al. 2017; Louizos et al. 2017; Goudet et al. 2017). Dong et al. (2017b) provide an interpretive loss function to increase the visual fidelity of the learned features. More importantly Dong et al. (2017a) show that by training DNNs with adversarial data and a consistent loss, we can trace back errors made by the DNN to individual neurons and identify whether the data was adversarial. Santoro et al. (2017) give a DNN the ability to answer relational reasoning questions about a specific environment, by introducing a relational reasoning module that learns a relational function, which can be applied to any DNN. Palm et al. (2017) build on work by Santoro et al. (2017) and introduces a recurrent relational network which can take the temporal component into account. Li et al. (2017) introduce an explicit structure to DNNs for visual recognition by building in an AND-OR grammar directly in the network structure. This leads to better interpretation of the information flow in the network, hence increased parsimony in attribution methods. Louizos et al. (2017) make use of generative neural networks perform causal inference and Goudet et al. (2017) use generative neural networks to learn functional causal models. Intrinsic methods do not explicitly explain anything by themselves. Instead they increase fidelity, clarity and parsimony in attribution methods. This class of methods is different from attribution methods because it tries to make the DNN inherently more interpretable by changing the architecture of the DNN, where attribution methods use what is there already and only transform aspects of the representation to something meaningful after the network is trained.

6 Addressing General Concerns

As indicated in Fig. 1, users have certain values, that in relation to a particular technology may lead to concerns, that in relation to particular applications can lead to specific questions. Mittelstadt et al. (2016) and Danks and London (2017) distinguish various concerns that users may have. The kinds of concerns they discuss focus to a large extent on the inconclusiveness, inscrutability or misguidedness of used evidence. That is, they concern to a significant extent the reliability and accessibility of used data (data mining, generally speaking). In addition to apprehensions about data, there are concerns that involve aspects of the processing itself, e.g. the inferential validity of an algorithm. Also, questions may be raised about the validity of a training process (e.g. requiring information about how exactly a DNN is trained). In the following, we provide a list of general concerns that should be addressed when developing predictive models such as DNNs:

Flawed data collection

Data collection may be flawed in several ways. Large labeled datasets that are used to train DNNs are either acquired by researchers (often via crowdsourcing) or by companies that 'own' the data. However, data quality may depend on multiple factors such as noise or censoring and there is no strict control on whether data is annotated correctly. Furthermore, the characteristics of the workers who annotated the data may introduce unwanted biases (Barocas and Selbst 2016). These biases may be due to preferences that do not generalize across cultures or due to stereotyping, where sensitivity to irrelevant attributes such as race or gender may induce unfair actionable outcomes. The same holds for the selection of the data that is used for annotation in the first place. Used data may reflect the status quo, which is not necessarily devoid of biases (Caliskan et al. 2017). Furthermore, selection bias may have as a result that data collected in one setting need not generalize to other settings. For example, video data used to train autonomous driving systems may not generalize to other locations or conditions.

Inscrutable data use

The exact use of the data to train DNNs may also be opaque. Users may worry about what (part of the) data exactly has led to the outcome. Often it is not even known to the data subject which personal data is being used for what purposes. A case in point is the use of person data for risk profiling by governmental institutions. For example, criticisms have been raised about the way the Dutch SyRI system uses data to detect fraud.[2] Furthermore, the involvement of expert users who may be prone to biases as well may have an implicit influence on DNN training.

Suboptimal inferences

The inferences made by DNNs are of a correlational rather than a causal nature. This implies that subtle correlations between input features may influence network output, which themselves may be driven by various biases. Work is in progress to mitigate or remove the influence of sensitive variables that should not affect decision outcomes by embracing causal inference procedures (Chiappa and Gillam 2018). Note further that the impact of suboptimal inferences is domain dependent. For example, in medicine and the social sciences, suboptimal inferences may directly affect the lives of individuals or whole populations whereas in the exact sciences, suboptimal inferences may affect evidence for or against a specific scientific theory.

Undesirable outcomes

End users or data subjects may feel that the outcome of the DNN is somehow undesirable in relation to the particular values they hold, e.g. violating fairness or privacy. Importantly, actionable outcomes should take into account preferences of the stakeholder, which can be an individual (e.g. when deciding on further medical investigation) as well as the community as a whole (e.g. in case of policies about autonomous driving or predictive policing). These considerations demand the involvement of domain experts and ethicists already in the earliest stages of model development. Finally, model predictions may be of a statistical

[2] https://bijvoorbaatverdacht.nl

rather than deterministic nature. This speaks for the inclusion of decision-theoretic constructs in deciding on optimal actionable outcomes (von Neumann and Morgenstern 1953).

Adversarial attacks

Images (Szegedy et al. 2013; Cubuk et al. 2017) and audio (Carlini and Wagner 2018) can easily be distorted with modifications that are imperceptible to humans. Such distortions cause DNNs to make incorrect inferences and can be done with the purpose of intentionally misleading DNNs (e.g. yielding predictions in favor of the perpetrator). Work in progress shows that there are methods to detect adversarial instances (Rawat et al. 2017) and to mitigate the attacks (Lin et al. 2017). However further research is needed to increase the robustness of DNNs against adversarial attacks as there are no methods in existence that fully diminish the effects of adversarial attacks.

As stated by Doran et al. (2017), explanation methods may make predictive models such as DNNs more comprehensible. However, explanation methods alone not completely resolve the raised concerns.

7 Discussion

In this chapter we set out to analyze the question of "What can be explained?" given the users and their needs, laws and regulations, and existing explanation methods. Specifically, we looked at the capabilities of explanation methods and analyzed which questions/concerns about explainability these methods address in the context of DNNs.

Overall, it is clear that (visual) explanations can be given about various aspects of the influence of the input on the output (e.g. given the input data, which aspects of the data lead to the output?), by making use of both rule-extraction and attribution methods. Also, when used in combination with attribution methods, intrinsic methods lead to more explainable DNNs. It is likely that in the future we will see the rise of a new category of explanation methods that combine aspects of rule-extraction, attribution and intrinsic methods, to answer specific questions in a simple human interpretable language.

Furthermore, it is obvious that current explanation methods are tailored to expert users, since the interpretation of the results require knowledge of the DNN process. As far as we are aware, explanation methods, e.g. intuitive explanation interfaces, for lay users do not exist. Ideally, if such explanation methods would exist, they should be able to answer, in a simple human language, questions about every operation that the application performs. This is not an easy task since the number of conceivable questions one could ask about the working of an application is substantial.

Two particular concerns, which are difficult to address with explanation methods, is the concern about bias in datasets that leads to biased DNNs, as well as the suspicion about unfair outcomes: Can we indicate that the DNN is biased, and if so, can we remove the bias? Has the DNN been applied responsibly? These are not problems that are directly solvable with explanation methods. However, explanation methods alleviate the first problem to the extent that learned features can be visualized (using attribution methods) and further analyzed for bias using other methods that are not explanation methods. For the second problem, more general measures, such as regulations and laws, will need to be developed.

References

Adler, P., Falk, C., Friedler, S. A., Rybeck, G., Scheidegger, C., Smith, B., and Venkatasubramanian, S. (2016). Auditing black-box models for indirect influence. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE.

Ancona, M., Ceolini, E., Oztireli, C., and Gross, M. (2018). Towards better understanding of gradient-based attribution methods for deep neural networks. In *6th International Conference on Learning Representations (ICLR 2018)*.

Andrews, R., Diederich, J., and Tickle, A. B. (1995). Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems*, 8(6):373–389.

Arbatli, A. D. and Akin, H. L. (1997). Rule extraction from trained neural networks using genetic algorithms. *Nonlinear Analysis: Theory, Methods & Applications*, 30(3):1639–1648.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7).

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

Barocas, S. and Selbst, A. D. (2016). Big data's disparate impact. *Cal. L. Rev.*, 104:671.

Binder, A., Bach, S., Montavon, G., Müller, K.-R., and Samek, W. (2016). Layer-wise relevance propagation for deep neural network architectures. In *Information Science and Applications (ICISA) 2016*, pages 913–922. Springer.

Bojarski, M., Yeres, P., Choromanska, A., Choromanski, K., Firner, B., Jackel, L., and Muller, U. (2017). Explaining how a deep neural network trained with end-to-end learning steers a car. *arXiv preprint arXiv:1704.07911*.

Caliskan, A., Bryson, J. J., and Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Carlini, N. and Wagner, D. (2018). Audio adversarial examples: Targeted attacks on speech-to-text. *arXiv preprint arXiv:1801.01944*.

Chiappa, S. and Gillam, T. P. (2018). Path-specific counterfactual fairness. *arXiv preprint arXiv:1802.08139*.

Craven, M. W. and Shavlik, J. W. (1994). Using sampling and queries to extract rules from trained neural networks. In *Machine Learning Proceedings 1994*, pages 37–45. Elsevier.

Cubuk, E. D., Zoph, B., Schoenholz, S. S., and Le, Q. V. (2017). Intriguing properties of adversarial examples. *arXiv preprint arXiv:1711.02846*.

Danks, D. and London, A. J. (2017). Algorithmic bias in autonomous systems. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4691–4697. AAAI Press.

Dong, Y., Su, H., Zhu, J., and Bao, F. (2017a). Towards interpretable deep neural networks by leveraging adversarial examples. *arXiv preprint arXiv:1708.05493*.

Dong, Y., Su, H., Zhu, J., and Zhang, B. (2017b). Improving interpretability of deep neural networks with semantic information. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Doran, D., Schulz, S., and Besold, T. R. (2017). What does explainable AI really mean? a new conceptualization of perspectives. *arXiv preprint arXiv:1710.00794*.

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Doshi-Velez, F., Kortz, M., Budish, R., Bavitz, C., Gershman, S. J., O'Brien, D., Shieber, S., Waldo, J., Weinberger, D., and Wood, A. (2017). Accountability of AI under the law: The role of explanation. *SSRN Electronic Journal*.

Elman, J. L. (1989). Representation and structure in connectionist models. Technical report.

Erhan, D., Bengio, Y., Courville, A., and Vincent, P. (2009). Visualizing higher-layer features of a deep network. *University of Montreal*, 1341:3.

Floridi, L., Fresco, N., and Primiero, G. (2015). On malfunctioning software. *Synthese*, 192(4):1199–1220.

Fong, R. C. and Vedaldi, A. (2017). Interpretable explanations of black boxes by meaningful perturbation. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE.

Goudet, O., Kalainathan, D., Caillou, P., Lopez-Paz, D., Guyon, I., Sebag, M., Tritas, A., and Tubaro, P. (2017). Learning functional causal models with generative neural networks. *arXiv preprint arXiv:1709.05321*.

Grün, F., Rupprecht, C., Navab, N., and Tombari, F. (2016). A taxonomy and library for visualizing learned features in convolutional neural networks. *arXiv preprint arXiv:1606.07757*.

Guçlütürk, Y., Güçlü, U., Perez, M., Jair Escalante, H., Baro, X., Guyon, I., Andujar, C., Jacques Junior, J., Madadi, M., Escalera, S., van Gerven, M. A. J., and van Lier, R. (2017). Visualizing apparent personality analysis with deep residual networks. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 3101–3109.

Gunning, D. (2017). Explainable artificial intelligence (XAI). *Defense Advanced Research Projects Agency (DARPA)*.

Hall, P., Phan, W., and Ambati, S. (2017). Ideas on interpreting machine learning. Available online at: https://www.oreilly.com/ideas/ideas-on-interpreting-machine-learning.

Holzinger, A., Biemann, C., Pattichis, C. S., and Kell, D. B. (2017a). What do we need to build explainable AI systems for the medical domain? *arXiv preprint arXiv:1712.09923*.

Holzinger, A., Plass, M., Holzinger, K., Crişan, G. C., Pintea, C.-M., and Palade, V. (2017b). A glass-box interactive machine learning approach for solving NP-hard problems with the human-in-the-loop. *arXiv preprint arXiv:1708.01104*.

Kindermans, P.-J., Hooker, S., Adebayo, J., Alber, M., Schütt, K. T., Dähne, S., Erhan, D., and Kim, B. (2017). The (un) reliability of saliency methods. *arXiv preprint arXiv:1711.00867*.

Kindermans, P.-J., Schütt, K. T., Müller, K.-R., and Dähne, S. (2016). Investigating the influence of noise and distractors on the interpretation of neural networks. *arXiv preprint arXiv:1611.07270*.

Koh, P. W. and Liang, P. (2017). Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research (PMLR)*, pages 1885–1894.

Lakkaraju, H., Kamar, E., Caruana, R., and Leskovec, J. (2017). Interpretable & explorable approximations of black box models. *arXiv preprint arXiv:1707.01154*.

Lee, H., Tajmir, S., Lee, J., Zissen, M., Yeshiwas, B. A., Alkasab, T. K., Choy, G., and Do, S. (2017). Fully automated deep learning system for bone age assessment. *Journal of Digital Imaging (JDI)*, 30(4):427–441.

Li, J., Monroe, W., and Jurafsky, D. (2016). Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

Li, X., Wu, T., Song, X., and Krim, H. (2017). AOGNets: Deep AND-OR grammar networks for visual recognition. *arXiv preprint arXiv:1711.05847*.

Lin, Y.-C., Liu, M.-Y., Sun, M., and Huang, J.-B. (2017). Detecting adversarial attacks on neural network policies with visual foresight. *arXiv preprint arXiv:1710.00814*.

Lockett, A., Jefferies, T., Etheridge, N., and Brewer, A. White paper tag predictions: How DISCO AI is bringing deep learning to legal technology. Available online at: https://www.csdisco.com/disco-ai.

Louizos, C., Shalit, U., Mooij, J. M., Sontag, D., Zemel, R., and Welling, M. (2017). Causal effect inference with deep latent-variable models. In *Advances in Neural Information Processing Systems 30 (NIPS)*, pages 6446–6456.

Lu, J., Tokinaga, S., and Ikeda, Y. (2006). Explanatory rule extraction based on the trained neural network and the genetic programming. *Journal of the Operations Research Society of Japan (JORSJ)*, 49(1):66–82.

Marcus, G. (2018). Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*.

Markowska-Kaczmar, U. and Wnuk-Lipiński, P. (2004). Rule extraction from neural network by genetic algorithm with pareto optimization. *Artificial Intelligence and Soft Computing-ICAISC 2004*, pages 450–455.

Mittelstadt, B. D., Allo, P., Taddeo, M., Wachter, S., and Floridi, L. (2016). The ethics of algorithms: Mapping the debate. *Big Data & Society*, 3(2).

Montavon, G., Samek, W., and Müller, K.-R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15.

Murdoch, W. J., Liu, P. J., and Yu, B. (2018). Beyond word importance: Contextual decomposition to extract interactions from LSTMs. In *International Conference on Learning Representations (ICLR)*.

Murdoch, W. J. and Szlam, A. (2017). Automatic rule extraction from long short term memory networks. In *International Conference on Learning Representations (ICLR)*.

Olah, C., Mordvintsev, A., and Schubert, L. (2017). Feature visualization. *Distill*. Available online at: https://distill.pub/2017/feature-visualization.

Olah, C., Satyanarayan, A., Johnson, I., Carter, S., Schubert, L., Ye, K., and Mordvintsev, A. (2018). The building blocks of interpretability. *Distill*. Available online at: https://distill.pub/2018/building-blocks.

Palm, R. B., Paquet, U., and Winther, O. (2017). Recurrent relational networks for complex relational reasoning. *arXiv preprint arXiv:1711.08028*.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. (2017). Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (ASIA CCS '17)*, pages 506–519.

Rawat, A., Wistuba, M., and Nicolae, M.-I. (2017). Adversarial phenomenon in the eyes of Bayesian deep learning. *arXiv preprint arXiv:1711.08244*.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016a). Nothing else matters: Model-agnostic explanations by identifying prediction invariance. In *NIPS Workshop on Interpretable Machine Learning in Complex Systems*.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016b). "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 (KDD '16)*, pages 1135–1144.

Robnik-Šikonja, M. and Kononenko, I. (2008). Explaining classifications for individual instances. *IEEE Transactions on Knowledge and Data Engineering*, 20(5):589–600.

Samek, W., Wiegand, T., and Müller, K.-R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*.

Santoro, A., Raposo, D., Barrett, D. G., Malinowski, M., Pascanu, R., Battaglia, P., and Lillicrap, T. (2017). A simple neural network module for relational reasoning. *arXiv preprint arXiv:1706.01427*.

Seifert, C., Aamir, A., Balagopalan, A., Jain, D., Sharma, A., Grottel, S., and Gumhold, S. (2017). Visualizations of deep neural networks in computer vision: A survey. In *Transparent Data Mining for Big and Small Data*, pages 123–144. Springer.

Shrikumar, A., Greenside, P., and Kundaje, A. (2017). Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research (PMLR)*.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

von Neumann, J. and Morgenstern, O. (1953). *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, NJ, 3rd edition.

Wachter, S., Mittelstadt, B., and Floridi, L. (2017). Transparent, explainable, and accountable AI for robotics. *Science Robotics*, 2(6).

Weller, A. (2017). Challenges for transparency. *Workshop on Human Interpretability in Machine Learning – ICML 2017*.

Wu, T., Li, X., Song, X., Sun, W., Dong, L., and Li, B. (2017). Interpretable R-CNN. *arXiv preprint arXiv:1711.05226*.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*, pages 818–833. Springer.

Zeiler, M. D., Krishnan, D., Taylor, G. W., and Fergus, R. (2010). Deconvolutional networks. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2528–2535. IEEE.

Zeng, H. (2016). Towards better understanding of deep learning with visualization.

Zilke, J. R., Mencía, E. L., and Janssen, F. (2016). DeepRED – Rule extraction from deep neural networks. In *International Conference on Discovery Science (ICDS)*, pages 457–473. Springer.

Zintgraf, L. M., Cohen, T. S., Adel, T., and Welling, M. (2017). Visualizing deep neural network decisions: Prediction difference analysis. In *International Conference on Learning Representations (ICLR)*.

Part II
Explainability and Interpretability in Machine Learning

Learning Functional Causal Models with Generative Neural Networks

Olivier Goudet, Diviyan Kalainathan, Philippe Caillou, Isabelle Guyon, David Lopez-Paz, and Michèle Sebag

Abstract We introduce a new approach to functional causal modeling from observational data, called *Causal Generative Neural Networks* (CGNN). CGNN leverages the power of neural networks to learn a generative model of the joint distribution of the observed variables, by minimizing the Maximum Mean Discrepancy between generated and observed data. An approximate learning criterion is proposed to scale the computational cost of the approach to linear complexity in the number of observations. The performance of CGNN is studied throughout three experiments. Firstly, CGNN is applied to cause-effect inference, where the task is to identify the best causal hypothesis out of "$X \rightarrow Y$" and "$Y \rightarrow X$". Secondly, CGNN is applied to the problem of identifying v-structures and conditional independences. Thirdly, CGNN is applied to multivariate functional causal modeling: given a skeleton describing the direct dependences in a set of random variables $\mathbf{X} = [X_1, \ldots, X_d]$, CGNN orients the edges in the skeleton to uncover the directed acyclic causal graph describing the causal structure of the random variables. On all three tasks, CGNN is extensively assessed on both artificial and real-world data, comparing favorably to the state-of-the-art. Finally, CGNN is extended to handle the case of confounders, where latent variables are involved in the overall causal model.

O. Goudet (✉) · D. Kalainathan · P. Caillou · M. Sebag
Team TAU - CNRS, INRIA, Université Paris Sud, Université Paris Saclay, Paris, France
e-mail: olivier.goudet@inria.fr; Diviyan.kalainathan@lri.fr; Caillou@lri.fr; sebag@lri.fr

I. Guyon
INRIA, Université Paris Sud, Université Paris Saclay, Paris, France

ChaLearn, Berkeley, CA, USA
e-mail: guyon@chalearn.org

D. Lopez-Paz
Facebook AI Research, Menlo Park, CA, USA
e-mail: david@lopezpaz.org

H. J. Escalante et al. (eds.), *Explainable and Interpretable Models in Computer Vision and Machine Learning*, The Springer Series on Challenges in Machine Learning, https://doi.org/10.1007/978-3-319-98131-4_3

Keywords Generative neural networks · Causal structure discovery · Cause-effect pair problem · Functional causal models · Structural equation models

1 Introduction

Deep learning models have shown extraordinary predictive abilities, breaking records in image classification (Krizhevsky et al. 2012), speech recognition (Hinton et al. 2012), language translation (Cho et al. 2014), and reinforcement learning (Silver et al. 2016). However, the predictive focus of black-box deep learning models leaves little room for explanatory power. More generally, current machine learning paradigms offer no protection to avoid mistaking correlation by causation. For example, consider the prediction of target variable Y given features X and Z, assuming that the underlying generative process is described by the equations:

$$\begin{aligned} X, E_Y, E_Z &\sim \text{Uniform}(0, 1), \\ Y &\leftarrow 0.5X + E_Y, \\ Z &\leftarrow Y + E_Z, \end{aligned}$$

with (E_Y, E_Z) additive noise variables. The above model states that the values of Y are computed as a function of the values of X (we say that X causes Y), and that the values of Z are computed as a function of the values of Y (Y causes Z). The "assignment arrows" emphasize the asymmetric relations between all three random variables. However, as Z provides a stronger signal-to-noise ratio than X for the prediction of Y, the best regression solution in terms of least-square error is

$$\hat{Y} = 0.25X + 0.5Z$$

The above regression model, a typical case of inverse regression after Goldberger (1984), would wrongly explain some changes in Y as a function of changes in Z, although Z does not cause Y. In this simple case, there exists approaches overcoming the inverse regression mistake and uncovering all true cause-effect relations (Hoyer et al. 2009). In the general case however, mainstream machine learning approaches fail to understand the relationships between all three distributions, and might attribute some effects on Y to changes in Z.

Mistaking correlation for causation can be catastrophic for agents who must plan, reason, and decide based on observations. Thus, discovering causal structures is of crucial importance.

The gold standard to discover causal relations is to perform experiments (Pearl 2003). However, experiments are in many cases expensive, unethical, or impossible to realize. In these situations, there is a need for *observational causal discovery*, that is, the estimation of causal relations from observations alone (Spirtes et al. 2000; Peters et al. 2017).

In the considered setting, *observational* empirical data (drawn independent and identically distributed from an unknown distribution) is given as a set of n samples of real valued feature vectors of dimension d. We denote the corresponding random vector as $\mathbf{X} = [X_1, \ldots, X_d]$. We seek a Functional Causal Model (FCM), also known as Structural Equation Model (SEM), that best matches the underlying data-generating mechanism(s) in the following sense: under relevant manipulations/interventions/experiments the FCM would produce data distributed similarly to the real data obtained in similar conditions.

Let intervention *do(X=x)* be defined as the operation on distribution obtained by clamping variable X to value x, while the rest of the system remains unchanged (Pearl 2009). It is said that variable X_i is a **direct cause** of X_j with respect to $X_1, \ldots, X_d$ iff different interventions on variable X result in different marginal distributions on X_j, everything else being equal:

$$P_{X_j|\mathrm{do}(X_i=x,\mathbf{X}_{\backslash ij}=\mathbf{c})} \neq P_{X_j|\mathrm{do}(X_i=x',\mathbf{X}_{\backslash ij}=\mathbf{c})} \tag{1}$$

with $\mathbf{X}_{\backslash ij} := X_{\{1,\ldots,d\}\backslash i,j}$ the set of all variables except X_i and X_j, scalar values $x \neq x'$, and vector value $\mathbf{c}$. Distribution $P_{X_j|\mathrm{do}(X_i=x,\mathbf{X}_{\backslash ij}=c)}$ is the resulting interventional distribution of the variable X_j when the variable X_i is clamped to value x, while keeping all other variables at a fixed value (Mooij et al. 2016).

As said, conducting such interventions to determine direct causes and effects raises some limitations. For this reason, this paper focuses on learning the causal structure from observational data only, where the goal and validation of the proposed approach is to match the known "ground truth" model structure.

A contribution of the paper is to unify several state-of-art methods into one single consistent and more powerful approach. On the one hand, leading researchers at UCLA, Carnegie Mellon, University of Crete and elsewhere have developed powerful algorithms exploiting Markov properties of directed acyclic graphs (DAGs) (Spirtes et al. 2000; Tsamardinos et al. 2006; Pearl 2009). On the other hand, the Tübingen School has proposed new and powerful functional causal models (FCM) algorithms exploiting the asymmetries in the joint distribution of cause-effect pairs (Hoyer et al. 2009; Stegle et al. 2010; Daniusis et al. 2012; Mooij et al. 2016).

In this paper, the learning of functional causal models is tackled in the search space of generative neural networks (Kingma and Welling 2013; Goodfellow et al. 2014), and aims at the functional causal model (structure and parameters), best fitting the underlying data generative process. The merits of the proposed approach, called Causal Generative Neural Network (CGNN) are extensively and empirically demonstrated compared to the state of the art on artificial and real-world benchmarks.

This paper is organized as follows: Sect. 2 introduces the problem of learning an FCM and the underlying assumptions. Section 3 briefly reviews and discusses the state of the art in causal modeling. The FCM modeling framework within the search space of generative neural networks is presented in Sect. 4. Section 5 reports on an extensive experimental validation of the approach comparatively to the state of the art for pairwise cause-effect inference and graph recovery. An extension of the proposed framework to deal with potential confounding variables is presented in Sect. 6. The paper concludes in Sect. 7 with some perspectives for future works.

2 Problem Setting

A Functional Causal Model (FCM) upon a random variable vector $\mathbf{X} = [X_1, \ldots, X_d]$ is a triplet $(\mathscr{G}, f, \mathscr{E})$, representing a set of equations:

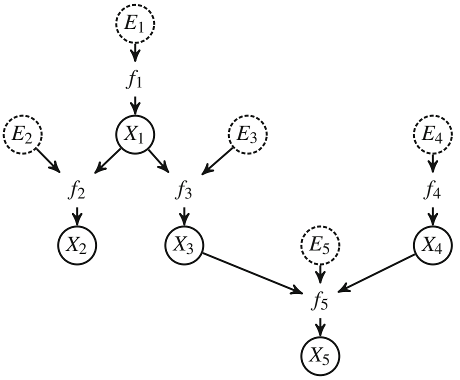
$$X_i \leftarrow f_i(X_{\mathrm{Pa}(i;\mathscr{G})}, E_i), E_i \sim \mathscr{E}, \text{ for } i = 1, \ldots, d \tag{2}$$

Each equation characterizes the direct causal relation explaining variable X_i from the set of its causes $X_{\mathrm{Pa}(i;\mathscr{G})} \subset \{X_1, \ldots, X_d\}$, based on some *causal mechanism* f_i involving besides $X_{\mathrm{Pa}(i;\mathscr{G})}$ some random variable E_i drawn after distribution $\mathscr{E}$, meant to account for all unobserved variables.

Letting $\mathscr{G}$ denote the causal graph obtained by drawing arrows from causes $X_{\mathrm{Pa}(i;\mathscr{G})}$ towards their effects X_i, we restrict ourselves to directed acyclic graphs (DAG), where the propagation of interventions to end nodes is assumed to be instantaneous. This assumption suitably represents causal phenomena in cross-sectional studies. An example of functional causal model with five variables is illustrated on Fig. 1.

2.1 Notations

By abuse of notation and for simplicity, a variable X and the associated node in the causal graph, in one-to-one correspondence, are noted in the same way. Variables X and Y are adjacent iff there exists an edge between both nodes in the graph. This edge can model (1) a direct causal relationship ($X \rightarrow Y$ or $Y \rightarrow X$); (2) a causal relationship in either direction ($X - Y$); (3) a non-causal association ($X \leftrightarrow Y$) due to external common causes (Richardson and Spirtes 2002).



$$\begin{cases} X_1 = f_1(E_1) \\ X_2 = f_2(X_1{}^{c}E_2) \\ X_3 = f_3(X_1{}^{c}E_3) \\ X_4 = f_4(E_4) \\ X_5 = f_5(X_3{}^{c}X_4{}^{c}E_5) \end{cases}$$

Fig. 1 Example of a functional causal model (FCM) on $\mathbf{X} = [X_1, \ldots, X_5]$. Left: causal graph $\mathscr{G}$; right: causal mechanisms

Conditional independence: $(X \perp\!\!\!\perp Y|Z)$ is meant as variables X and Y are independent conditionally to Z, i.e. $P(X, Y|Z) = P(X|Z)P(Y|Z)$.

V-structure, a.k.a. unshielded collider: Three variables $\{X, Y, Z\}$ form a v-structure iff their causal structure is: $X \rightarrow Z \leftarrow Y$.

Skeleton of the DAG: the skeleton of the DAG is the undirected graph obtained by replacing all edges by undirected edges.

Markov equivalent DAG: two DAGs with same skeleton and same v-structures are said to be *Markov equivalent* (Pearl and Verma 1991). A *Markov equivalence class* is represented by a *Completed Partially Directed Acyclic Graph* (CPDAG) having both directed and undirected edges.

2.2 Assumptions and Properties

The state of the art in causal modeling most commonly involves four assumptions:

Causal sufficiency assumption (CSA): $\mathbf{X}$ is said to be *causally sufficient* if no pair of variables $\{X_i, X_j\}$ in $\mathbf{X}$ has a common cause external to $\mathbf{X}_{\backslash i,j}$.

Causal Markov assumption (CMA): all variables are independent of their non-effects (non descendants in the causal graph) conditionally to their direct causes (parents) (Spirtes et al. 2000). For an FCM, this assumption holds if the graph is a DAG and error terms E_i in the FCM are independent on each other (Pearl 2009).

Conditional independence relations in an FCM: if CMA applies, the data generated by the FCM satisfy all conditional independence (CI) relations among variables in $\mathbf{X}$ via the notion of d-separation (Pearl 2009). CIs are called Markov properties. Note that there may be more CIs in data than present in the graph (see the Faithfulness assumption below). The joint distribution of the variables is expressed as the product of the distribution of each variable conditionally on its parents in the graph.

Causal Faithfulness Assumption (CFA): the joint distribution $P(\mathbf{X})$ is *faithful* to the graph $\mathscr{G}$ of an FCM iff every conditional independence relation that holds true in P is entailed by $\mathscr{G}$ (Spirtes and Zhang 2016). Therefore, if there exists an independence relation in $\mathbf{X}$ that is not a consequence of the Causal Markov assumption, then $\mathbf{X}$ is *unfaithful* (Scheines 1997). It follows from CMA and CFA that every causal path in the graph corresponds to a dependency between variables, and vice versa.

V-structure property: Under CSA, CMA and CFA, if variables $\{X, Y, Z\}$ satisfy: (1) $\{X, Y\}$ and $\{Y, Z\}$ are adjacent; (2) $\{X, Z\}$ are NOT adjacent; (3) $X \not\perp\!\!\!\perp Z|Y$, then their causal structure is a v-structure $(X \rightarrow Y \leftarrow Z)$.

3 State of the Art

This section reviews methods to infer causal relationships, based on either the Markov properties of a DAG such as v-structures or asymmetries in the joint distributions of pairs of variables.

3.1 *Learning the CPDAG*

Structure learning methods classically use conditional independence (CI) relations in order to identify the Markov equivalence class of the sought Directed Acyclic Graph, referred to as CPDAG, under CSA, CMA and CFA.

Considering the functional model on $\mathbf{X} = [X_1, \ldots, X_5]$ on Fig. 1, the associated DAG $\mathscr{G}$ and graph skeleton are respectively depicted on Fig. 2a, b. Causal modeling exploits observational data to recover the $\mathscr{G}$ structure from all CI (Markov proper-

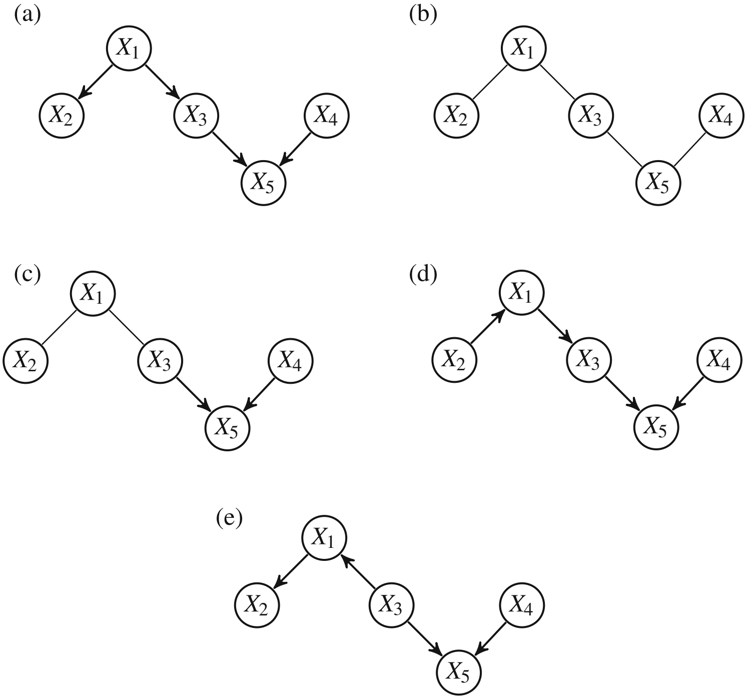


Fig. 2 Example of a Markov equivalent class. There exists three graphs (**a**, **d**, **e**) consistent with a given graph skeleton (**b**); the set of these consistent graphs defines the Markov equivalent class (**c**). (**a**) The exact DAG of $\mathscr{G}$. (**b**) The skeleton of $\mathscr{G}$. (**c**) The CPDAG of $\mathscr{G}$. (**d**) A Markov equivalent DAG of $\mathscr{G}$. (**e**) Another Markov equivalent DAG of $\mathscr{G}$

ties) between variables.[1] Under CSA, CMA and CFA, as $(X_3 \perp\!\!\!\perp X_4 | X_5)$ does not hold, a v-structure $X_3 \rightarrow X_5 \leftarrow X_4$ is identified (Fig. 2c). However, one also has $(X_1 \perp\!\!\!\perp X_5 | X_3)$ and $(X_2 \perp\!\!\!\perp X_3 | X_1)$. Thus the DAGs on Figs. 2d, e encode the same conditional independences as the true DAG (Fig. 2a). Therefore the true DAG cannot be fully identified based only on independence tests, and the edges between the pairs of nodes $\{X_1, X_2\}$ and $\{X_1, X_3\}$ are left undirected. The identification process thus yields the partially undirected graph depicted on Fig. 2c, called *Completed Partially Directed Acyclic Graph* (CPDAG).

The main three families of methods used to recover the CPDAG of an FCM with continuous data are constraint-based methods, score-based methods, and hybrid methods (Drton and Maathuis 2016).

3.1.1 Constraint-Based Methods

Constraint-based methods exploit conditional independences between variables to identify all v-structures. One of the most well-known constraint-based algorithms is the PC algorithm (Spirtes et al. 1993). PC first builds the DAG skeleton based on conditional independences among variables and subsets of variables. Secondly, it identifies v-structures (Fig. 2c). Finally, it uses propagation rules to orient remaining edges, avoiding the creation of directed cycles or new v-structures. Under CSA, CMA and CFA, and assuming an oracle indicating all conditional independences, PC returns the CPDAG of the functional causal model. In practice, PC uses statistical tests to accept or reject conditional independence at a given confidence level. Besides mainstream tests (e.g., s Z-test or T-Test for continuous Gaussian variables, and χ-squared or G-test for categorical variables), non-parametric independence tests based on machine learning are becoming increasingly popular, such as kernel-based conditional independence tests (Zhang et al. 2012). The FCI algorithm (Spirtes et al. 1999) extends PC; it relaxes the *causal sufficiency* assumption and deals with latent variables. The RFCI algorithm (Colombo et al. 2012) is faster than FCI and handles high-dimensional DAGs with latent variables. Achilles' heel of constraint-based algorithms is their reliance on conditional independence tests. The CI accuracy depends on the amount of available data, with exponentially increasing size with the number of variables. Additionally, the use of propagation rules to direct edges is prone to error propagation.

3.1.2 Score-Based Methods

Score-based methods explore the space of CPDAGs and minimize a global score. For example, the space of graph structures is explored using operators (*add edge*,

[1] The so-called constraint-based methods base the recovery of graph structure on conditional independence tests. In general, proofs of model identifiability assume the existence of an “oracle" providing perfect knowledge of the CIs, i.e. *de facto* assuming an infinite amount of training data.

remove edge, and *reverse edge*) by the Greedy Equivalent Search (GES) algorithm (Chickering 2002), returning the optimal structure in the sense of the Bayesian Information Criterion.[2]

In order to find the optimal CPDAG corresponding to the minimum score, the GES algorithm starts with an empty graph. A first forward phase is performed, iteratively adding edges to the model in order to improve the global score. A second backward phase iteratively removes edges to improve the score. Under CSA, CMA and CFA, GES identifies the true CPDAG in the large sample limit, if the score used is decomposable, score-equivalent and consistent (Chickering 2002). More recently, Ramsey (2015) proposed a GES extension called Fast Greedy Equivalence Search (FGES) algorithm. FGES uses the same scores and search algorithm with different data structures; it greatly speeds up GES by caching information about scores during each phase of the process.

3.1.3 Hybrid Algorithms

Hybrid algorithms combine ideas from constraint-based and score-based algorithms. According to Nandy et al. (2015), such methods often use a greedy search like the GES method on a restricted search space for the sake of computational efficiency. This restricted space is defined using conditional independence tests. For instance the Max-Min Hill climbing (MMHC) algorithm (Tsamardinos et al. 2006) firstly builds the skeleton of a Bayesian network using conditional independence tests and then performs a Bayesian-scoring greedy hill-climbing search to orient the edges. The Greedy Fast Causal Inference (GFCI) algorithm proceeds in the other way around, using FGES to get rapidly a first sketch of the graph (shown to be more accurate than those obtained with constraint-based methods), then using the FCI constraint-based rules to orient the edges in presence of potential confounders (Ogarrio et al. 2016).

3.2 Exploiting Asymmetry Between Cause and Effect

The abovementioned score-based and constraint-based methods do not take into account the full information from the observational data (Spirtes and Zhang 2016), such as data asymmetries induced by the causal directions.

[2] After Ramsey (2015), in the linear model with Gaussian variable case the individual BIC score to minimize for a variable X given its parents is up to a constant $n\ \ln(s) + c\ k\ \ln(n)$, where $n\ \ln(s)$ is the likelihood term, with s the residual variance after regressing X onto its parents, and n the number of data samples. $c\ k\ \ln(n)$ is a penalty term for the complexity of the graph (here the number of edges). $k = 2p + 1$, with p the total number of parents of the variable X in the graph. $c = 2$ by default, chosen empirically. The global score minimized by the algorithm is the sum over all variables of the individual BIC score given the parent variables in the graph.

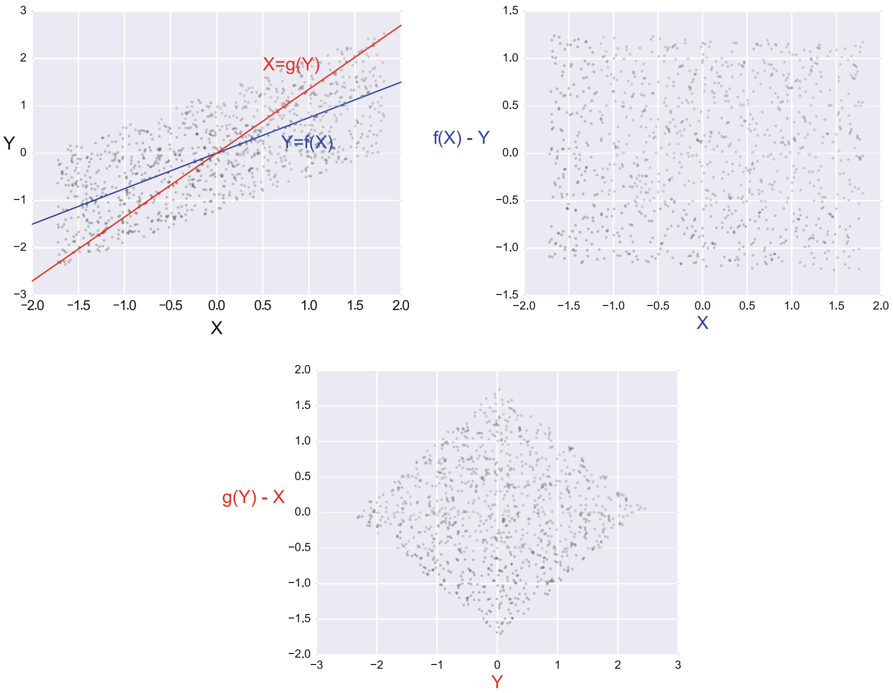


Fig. 3 Left: Joint distribution $P(X, Y)$ generated from DAG $X \rightarrow Y + E$, with E a uniform noise variable. The linear regression of Y on X (respectively of X on Y) is depicted as a blue (resp. red) curve. Middle: Error $f(X) - Y$ is independent of X. Right: Error $g(Y) - X$ is not independent of Y. The asymmetry establishes that the true causal model is $X \rightarrow Y$. Better seen in color

3.2.1 The Intuition

Let us consider FCM $Y = X + E$, with E a random noise independent of X by construction. Graph constraints cannot orient the $X - Y$ edge as both graphs $X \rightarrow Y$ and $Y \rightarrow X$ are Markov equivalent. However, the implicit v-structure $X \rightarrow Y \leftarrow E$ can be exploited provided that either X or E does not follow a **Gaussian distribution**. Consider the linear regression $Y = aX + b$ (blue curve in Fig. 3); the residual is independent of X. Quite the contrary, the residual of the linear regression $X = a'Y + b'$ (red curve in Fig. 3) is *not* independent of Y as far as the independence of the error term holds true (Shimizu et al. 2006). In this toy example, the asymmetries in the joint distribution of X and Y can be exploited to recover the causal direction $X \rightarrow Y$ (Spirtes and Zhang 2016).

3.2.2 Restriction on the Class of Causal Mechanisms Considered

Causal inference is bound to rely on assumptions such as non-Gaussianity or additive noise. In the absence of any such assumption, Zhang et al. (2016) show that, even in the bivariate case, for any function f and noise variable E independent of X such that $Y = f(X, E)$, it is always feasible to construct some $\tilde{f}$ and $\tilde{E}$, with $\tilde{E}$ independent of Y, such that $X = \tilde{f}(Y, \tilde{E})$. An alternative, supporting asymmetry detection and hinting at a causal direction, is based on restricting the class of functions f (e.g. only considering regular functions). According to Quinn et al. (2011), the first approach in this direction is LiNGAM (Shimizu et al. 2006). LiNGAM handles linear structural equation models, where each variable is continuous and modeled as:

$$X_i = \sum_k \alpha_k P_a^k(X_i) + E_i, i \in [\![1, n]\!] \tag{3}$$

with $P_a^k(X_i)$ the kth parent of X_i and α_k a real value. Assuming further that all probability distributions of source nodes in the causal graph are non-Gaussian, Shimizu et al. (2006) show that the causal structure is fully identifiable (all edges can be oriented).

3.2.3 Pairwise Methods

In the continuous, non-linear bivariate case, specific methods have been developed to orient the variable edge.[3] A well known example of bivariate model is the additive noise model (ANM) (Hoyer et al. 2009), with data generative model $Y = f(X)+E$, f a (possibly non-linear) function and E a noise independent of X. The authors prove the identifiability of the ANM in the following sense: if $P(X, Y)$ is consistent with ANM $Y = f(X)+E$, then (1) there exists no AMN $X = g(Y)+E'$ consistent with $P(X, Y)$; (2) the true causal direction is $X \rightarrow Y$. Under the independence assumption between E and X, the ANM admits a single non-identifiable case, the linear model with Gaussian input and Gaussian noise (Mooij et al. 2016).

A more general model is the post-nonlinear model (PNL) (Zhang and Hyvärinen 2009), involving an additional nonlinear function on the top of an additive noise: $Y = g(f(X) + E)$, with g an invertible function. The price to pay for this higher generality is an increase in the number of non identifiable cases.

The Gaussian Process Inference model (GPI) (Stegle et al. 2010) infers the causal direction without explicitly restricting the class of possible causal mechanisms. The authors build two Bayesian generative models, one for $X \rightarrow Y$ and one for $Y \rightarrow X$, where the distribution of the cause is modeled with a Gaussian mixture model, and the causal mechanism f is a Gaussian process. The causal direction is determined from the generative model best fitting the data (maximizing the data likelihood). Identifiability here follows from restricting the underlying class of functions and

[3] These methods can be extended to the multivariate case and used for causal graph identification by orienting each edge in turn.

enforcing their smoothness (regularity). Other causal inference methods (Sgouritsa et al. 2015) are based on the idea that if $X \rightarrow Y$, the marginal probability distribution of the cause $P(X)$ is independent of the causal mechanism $P(Y|X)$, hence estimating $P(Y|X)$ from $P(X)$ should hardly be possible, while estimating $P(X|Y)$ based on $P(Y)$ may be possible. The reader is referred to Statnikov et al. (2012) and Mooij et al. (2016) for a thorough review and benchmark of the pairwise methods in the bivariate case.

A new ML-based approach tackles causal inference as a pattern recognition problem. This setting was introduced in the Causality challenges (Guyon 2013, 2014), which released 16,200 pairs of variables $\{X_i, Y_i\}$, each pair being described by a sample of their joint distribution, and labeled with the true ℓ_i value of their causal relationship, with ℓ_i ranging in $\{X_i \rightarrow Y_i, Y_i \rightarrow X_i, X_i \perp\!\!\!\perp Y_i, X_i \leftrightarrow Y_i$ (presence of a confounder)$\}$. The causality classifiers trained from the challenge pairs yield encouraging results on test pairs. The limitation of this ML-based causal modeling approach is that causality classifiers intrinsically depend on the representativity of the training pairs, assumed to be drawn from a same "Mother distribution" (Lopez-Paz et al. 2015).

Note that bivariate methods can be used to uncover the full DAG, and independently orient each edge, with the advantage that an error on one edge does not propagate to the rest of the graph (as opposed to constraint and score-based methods). However, bivariate methods do not leverage the full information available in the dependence relations. For example in the linear Gaussian case (linear model and Gaussian distributed inputs and noises), if a triplet of variables $\{A, B, C\}$ is such that A, B (respectively B, C) are dependent on each other but $A \perp\!\!\!\perp C$), a constraint-based method would identify the v-structure $A \rightarrow B \leftarrow C$ (unshielded collider); still, a bivariate model based on cause-effect asymmetry would neither identify $A \rightarrow B$ nor $B \leftarrow C$.

3.3 Discussion

This brief survey has shown the complementarity of CPDAG and pairwise methods. The former ones can at best return partially directed graphs; the latter ones do not optimally exploit the interactions between all variables.

To overcome these limitations, an extension of the bivariate post-nonlinear model (PNL) has been proposed (Zhang and Hyvärinen 2009), where an FCM is trained for any plausible causal structure, and each model is tested *a posteriori* for the required independence between errors and causes. The main PNL limitation is its super-exponential cost with the number of variables (Zhang and Hyvärinen 2009). Another hybrid approach uses a constraint based algorithm to identify a Markov equivalence class, and thereafter uses bivariate modelling to orient the remaining edges (Zhang and Hyvärinen 2009). For example, the constraint-based PC algorithm can identify the v-structure $X_3 \rightarrow X_5 \leftarrow X_4$ in an FCM (Fig. 2), enabling the bivariate PNL method to further infer the remaining arrows $X_1 \rightarrow X_2$ and $X_1 \rightarrow X_3$. Note that an effective combination of constraint-based and bivariate approaches requires a final verification phase to test the consistency between the v-structures and the edge orientations.

This paper aims to propose a unified framework getting the best out of both worlds of CPDAG and bivariate approaches.

An inspiration of the approach is the CAM algorithm (Bühlmann et al. 2014), which is an extension to the graph setting of the pairwise additive model (ANM) (Hoyer et al. 2009). In CAM the FCM is modeled as:

$$X_i = \sum_{k \in \text{Pa}(i;\mathscr{G})} f_k(X_k) + E_i, \text{ for } i = 1, \ldots, d \tag{4}$$

Our method can be seen an extension of CAM, as it allows non-additive noise terms and non-additive contributions of causes, in order to model flexible conditional distributions, and addresses the problem of learning FCMs (Sect. 2):

$$X_i = f_i(X_{\text{Pa}(i;\mathscr{G})}, E_i), \text{ for } i = 1, \ldots, d \tag{5}$$

An other inspiration of our framework is the recent method of Lopez-Paz and Oquab (2016), where a conditional generative adversarial network is trained to model $X \rightarrow Y$ and $Y \rightarrow X$ in order to infer the causal direction based on the Occam's razor principle.

This approach, called **Causal Generative Neural Network (CGNN)**, features two original contributions. Firstly, multivariate causal mechanisms f_i are learned as **generative neural networks** (as opposed to, regression networks). The novelty is to use neural nets to model the joint distribution of the observed variables and learn a continuous FCM. This approach does not explicitly restrict the class of functions used to represent the causal models (see also Stegle et al. 2010), since neural networks are universal approximators. Instead, a regularity argument is used to enforce identifiability, in the spirit of supervised learning: the methods searches a trade-off between data fitting and model complexity.

Secondly, the data generative models are trained using a non-parametric score, the Maximum Mean Discrepancy (Gretton et al. 2007). This criterion is used instead of likelihood based criteria, hardly suited to complex data structures, or mean square criteria, implicitly assuming an additive noise (e.g. as in CAM, Eq. (4)).

Starting from a known skeleton, Sect. 4 presents a version of the proposed approach under the usual Markov, faithfulness, and causal sufficiency assumptions. The empirical validation of the approach is detailed in Sect. 5. In Sect. 6, the causal sufficiency assumption is relaxed and the model is extended to handle possible hidden confounding factors. Section 7 concludes the paper with some perspectives for future work.

4 Causal Generative Neural Networks

Let $\mathbf{X} = [X_1, \ldots, X_d]$ denote a set of continuous random variables with joint distribution P, and further assume that the joint density function h of P is continuous and strictly positive on a compact subset of $\mathbb{R}^d$ and zero elsewhere.

This section first presents the modeling of continuous FCMs with generative neural networks with a given graph structure (Sect. 4.1), the evaluation of a candidate model (Sect. 4.2), and finally, the learning of a best candidate from observational data (Sect. 4.3).

4.1 Modeling Continuous FCMs with Generative Neural Networks

We first show that there exists a (non necessarily unique) *continuous* functional causal model $(\mathcal{G}, f, \mathcal{E})$ such that the associated data generative process fits the distribution P of the observational data.

Proposition 1 *Let $X = [X_1, \ldots, X_d]$ denote a set of continuous random variables with joint distribution P, and further assume that the joint density function h of P is continuous and strictly positive on a compact and convex subset of $\mathbb{R}^d$, and zero elsewhere. Letting $\mathcal{G}$ be a DAG such that P can be factorized along $\mathcal{G}$,*

$$P(X) = \prod_i P(X_i | X_{Pa(i;\mathcal{G})})$$

there exists $f = (f_1, \ldots, f_d)$ with f_i a continuous function with compact support in $\mathbb{R}^{|Pa(i;\mathcal{G})|} \times [0, 1]$ such that $P(X)$ equals the generative model defined from FCM $(\mathcal{G}, f, \mathcal{E})$, with $\mathcal{E} = \mathcal{U}[0, 1]$ the uniform distribution on $[0, 1]$.

Proof In section "Proofs" in Appendix.

In order to model such continuous FCM $(\mathcal{G}, f, \mathcal{E})$ on d random variables $\mathbf{X} = [X_1, \ldots, X_d]$, we introduce the CGNN (Causal Generative Neural Network) depicted on Fig. 4.

Definition 1 A CGNN over d variables $[\hat{X}_1, \ldots, \hat{X}_d]$ is a triplet $\mathcal{C}_{\widehat{\mathcal{G}}, \hat{f}} = (\widehat{\mathcal{G}}, \hat{f}, \mathcal{E})$ where:

1. $\widehat{\mathcal{G}}$ is a Directed Acyclic Graph (DAG) associating to each variable $\hat{X}_i$ its set of parents noted $\hat{X}_{\mathrm{Pa}(i;\widehat{\mathcal{G}})}$ for $i \in [[1, d]]$

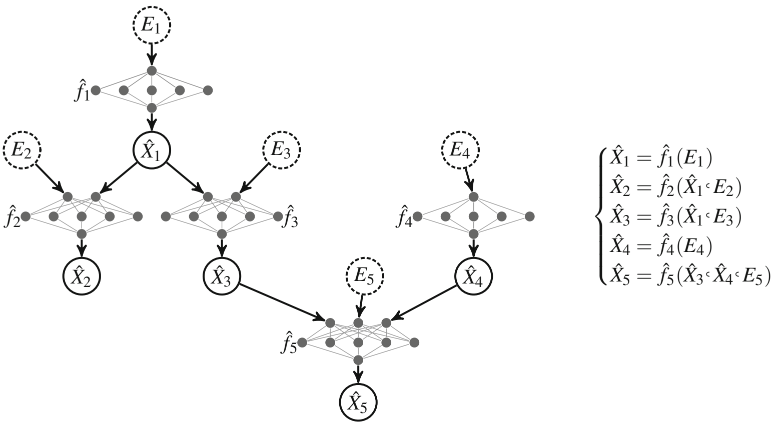

E_1
$\hat{f}_1$
$\hat{X}_1$
E_2
E_3
E_4
$\hat{f}_2$
$\hat{f}_3$
$\hat{f}_4$
$\hat{X}_2$
$\hat{X}_3$
E_5
$\hat{X}_4$
$\hat{f}_5$
$\hat{X}_5$
$\hat{X}_1 = \hat{f}_1(E_1)$
$\hat{X}_2 = \hat{f}_2(\hat{X}_1, E_2)$
$\hat{X}_3 = \hat{f}_3(\hat{X}_1, E_3)$
$\hat{X}_4 = \hat{f}_4(E_4)$
$\hat{X}_5 = \hat{f}_5(\hat{X}_3, \hat{X}_4, E_5)$

Fig. 4 Left: Causal generative neural network over variables $\hat{\mathbf{X}} = (\hat{X}_1, \ldots, \hat{X}_5)$. Right: Corresponding functional causal model equations

2. For $i \in [\![1, d]\!]$, causal mechanism $\hat{f}_i$ is a 1-hidden layer regression neural network with n_h hidden neurons:

$$\hat{X}_i = \hat{f}_i(\hat{X}_{\mathrm{Pa}(i;\widehat{\mathscr{G}})}, E_i) = \sum_{k=1}^{n_h} \bar{w}_k^i \sigma \left(\sum_{j \in \mathrm{Pa}(i;\mathscr{G})} \hat{w}_{jk}^i \hat{X}_j + w_k^i E_i + b_k^i \right) + \bar{b}^i \tag{6}$$

   with $n_h \in \mathbb{N}*$ the number of hidden units, $\bar{w}_k^i, \hat{w}_{jk}^i, w_k^i, b_k^i, \bar{b}^i \in \mathbb{R}$ the parameters of the neural network, and σ a continuous activation function.
3. Each variable E_i is independent of the *cause* X_i. Furthermore, all noise variables are mutually independent and drawn after same distribution $\mathscr{E}$.

It is clear from its definition that a CGNN defines a continuous FCM.

4.1.1 Generative Model and Interventions

A CGNN $\mathscr{C}_{\widehat{\mathscr{G}},\hat{f}} = (\widehat{\mathscr{G}}, \hat{f}, \mathscr{E})$ is a **generative** model in the sense that any sample $[e_{1,j}, \ldots, e_{d,j}]$ of the "noise" random vector $\mathbf{E} = [E_1, \ldots, E_d]$ can be used as "input" to the network to generate a data sample $[\hat{x}_{1,j}, \ldots, \hat{x}_{d,j}]$ of the estimated distribution $\hat{P}(\hat{\mathbf{X}} = [\hat{X}_1, \ldots, \hat{X}_d])$ by proceeding as follow:

1. Draw $\{[e_{1,j}, \ldots, e_{d,j}]\}_{j=1}^n$, n samples independent identically distributed from the joint distribution of independent noise variables $\mathbf{E} = [E_1, \ldots, E_d]$.
2. Generate n samples $\{[\hat{x}_{1,j}, \ldots, \hat{x}_{d,j}]\}_{j=1}^n$, where each estimate sample $\hat{x}_{i,j}$ of variable $\hat{X}_i$ is computed in the topological order of $\widehat{\mathscr{G}}$ from $\hat{f}_i$ with the jth

estimate samples $\hat{x}_{Pa(i;\hat{\mathscr{G}}),j}$ of $\hat{X}_{\text{Pa}(i;\hat{\mathscr{G}})}$ and the jth sample $e_{i,j}$ of the random noise variable E_i.

Notice that a CGNN generates a probability distribution $\hat{P}$ which is Markov with respect to $\widehat{\mathscr{G}}$, as the graph $\widehat{\mathscr{G}}$ is acyclic and the noise variables E_i are mutually independent.

Importantly, CGNN supports interventions, that is, freezing a variable X_i to some constant v_i. The resulting joint distribution noted $\hat{P}_{\text{do}(\hat{X}_i=v_i)}(\hat{X})$, called *interventional distribution* (Pearl 2009), can be computed from CGNN by discarding all causal influences on $\hat{X}_i$ and clamping its value to v_i. It is emphasized that intervening is different from conditioning (*correlation does not imply causation*). The knowledge of interventional distributions is essential for e.g., public policy makers, wanting to estimate the overall effects of a decision on a given variable.

4.2 Model Evaluation

The goal is to associate to each candidate solution $\mathscr{C}_{\widehat{\mathscr{G}},\hat{f}} = (\widehat{\mathscr{G}}, \hat{f}, \mathscr{E})$ a score reflecting how well this candidate solution describes the observational data. Firstly we define the model scoring function (Sect. 4.2), then we show that this model scoring function allows to build a CGNN generating a distribution $\hat{P}(\hat{X})$ that approximates $P(X)$ with arbitrary accuracy (Sect. 4.2.2).

4.2.1 Scoring Metric

The ideal score, to be minimized, is the distance between the joint distribution P associated with the ground truth FCM, and the joint distribution $\widehat{P}$ defined by the CGNN candidate $\mathscr{C}_{\hat{\mathscr{G}},\hat{f}} = (\hat{\mathscr{G}}, \hat{f}, \mathscr{E})$. A tractable approximation thereof is given by the Maximum Mean Discrepancy (MMD) (Gretton et al. 2007) between the n-sample observational data $\mathscr{D}$, and an n-sample $\widehat{\mathscr{D}}$ sampled after $\widehat{P}$. Overall, the CGNN $\mathscr{C}_{\hat{\mathscr{G}},\hat{f}}$ is trained by minimizing

$$S(\mathscr{C}_{\hat{\mathscr{G}},\hat{f}}, \mathscr{D}) = \widehat{\text{MMD}}_k(\mathscr{D}, \widehat{\mathscr{D}}) + \lambda|\widehat{\mathscr{G}}|, \tag{7}$$

with $\widehat{\text{MMD}}_k(\mathscr{D}, \widehat{\mathscr{D}})$ defined as:

$$\widehat{\text{MMD}}_k(\mathscr{D}, \widehat{\mathscr{D}}) = \frac{1}{n^2}\sum_{i,j=1}^{n} k(x_i, x_j) + \frac{1}{n^2}\sum_{i,j=1}^{n} k(\hat{x}_i, \hat{x}_j) - \frac{2}{n^2}\sum_{i,j=1}^{n} k(x_i, \hat{x}_j) \tag{8}$$

where kernel k usually is taken as the Gaussian kernel ($k(x, x') = \exp(-\gamma\|x - x'\|_2^2)$). The MMD statistic, with quadratic complexity in the sample size, has the good property that as n goes to infinity, it goes to zero iff $P = \hat{P}$ (Gretton et al. 2007). For scalability, a linear approximation of the MMD statistics based on $m = 100$ random features (Lopez-Paz 2016), called $\widehat{\text{MMD}}_k^m$, will also be used in the experiments (more in section “The Maximum Mean Discrepancy (MMD) Statistic” in Appendix).

Due to the Gaussian kernel being differentiable, $\widehat{\text{MMD}}_k$ and $\widehat{\text{MMD}}_k^m$ are differentiable, and backpropagation can be used to learn the CGNN made of networks $\hat{f}_i$ structured along $\hat{\mathscr{G}}$.

In order to compare candidate solutions with different structures in a fair manner, the evaluation score of Eq. (7) is augmented with a penalization term $\lambda|\widehat{\mathscr{G}}|$, with $|\widehat{\mathscr{G}}|$ the number of edges in $\hat{\mathscr{G}}$. Penalization weight λ is a hyper-parameter of the approach.

4.2.2 Representational Power of CGNN

We note $\mathscr{D} = \{[x_{1,j}, \ldots, x_{d,j}]\}_{j=1}^n$, the data samples independent identically distributed after the (unknown) joint distribution $P(\mathbf{X} = [X_1, \ldots, X_d])$, also referred to as observational data.

Under same conditions as in Proposition 1, ($P(X)$ being decomposable along graph $\mathscr{G}$, with continuous and strictly positive joint density function on a compact in $\mathbb{R}^d$ and zero elsewhere), there exists a CGNN $(\widehat{\mathscr{G}}, \hat{f}, \mathscr{E})$, that approximates $P(X)$ with arbitrary accuracy:

Proposition 2 *For $m \in [[1, d]]$, let Z_m denote the set of variables with topological order less than m and let d_m be its size. For any d_m-dimensional vector of noise values $e^{(m)}$, let $z_m(e^{(m)})$ (resp. $\widehat{z_m}(e^{(m)})$) be the vector of values computed in topological order from the FCM $(\mathscr{G}, f, \mathscr{E})$ (resp. the CGNN $(\mathscr{G}, \hat{f}, \mathscr{E})$). For any $\epsilon > 0$, there exists a set of networks $\hat{f}$ with architecture $\mathscr{G}$ such that*

$$\forall e^{(m)}, \|z_m(e^{(m)}) - \widehat{z_m}(e^{(m)})\| < \epsilon \tag{9}$$

Proof In section “Proofs” in Appendix.

Using this proposition and the $\widehat{\text{MMD}}_k$ scoring criterion presented in Eq. (8), it is shown that the distribution $\hat{P}$ of the CGNN can estimate the true observational distribution of the (unknown) FCM up to an arbitrary precision, under the assumption of an infinite observational sample:

Proposition 3 *Let $\mathscr{D}$ be an infinite observational sample generated from $(\mathscr{G}, f, \mathscr{E})$. With same notations as in Proposition 2, for every sequence ϵ_t, such that $\epsilon_t > 0$ and goes to zero when $t \to \infty$, there exists a set $\widehat{f_t} = (\hat{f}_1^t \ldots \hat{f}_d^t)$ such that $\widehat{MMD}_k$ between $\mathscr{D}$ and an infinite size sample $\widehat{\mathscr{D}}_t$ generated from the CGNN $(\mathscr{G}, \widehat{f_t}, \mathscr{E})$ is less than ε_t.*

Proof In section "Proofs" in Appendix.

Under these assumptions, as $\widehat{\mathrm{MMD}}_k(\mathscr{D}, \hat{\mathscr{D}}_t) \rightarrow 0$, as $t \rightarrow \infty$, it implies that the sequence of generated $\hat{P}_t$ converges in distribution toward the distribution P of the observed sample (Gretton et al. 2007). This result highlights the generality of this approach as we can model any kind of continuous FCM from observational data (assuming access to infinite observational data). Our class of model is not restricted to simplistic assumptions on the data generative process such as the additivity of the noise or linear causal mechanisms. But this strength comes with a new challenge relative to identifiability of such CGNNs as the result of Proposition 3 holds for any DAG $\widehat{\mathscr{G}}$ such that P can be factorized along $\mathscr{G}$ and then for any DAG in the Markov equivalence class of $\mathscr{G}$ (under classical assumption of CMA, CFA and CSA). In particular in the pairwise setting, when only two variables X and Y are observed, the joint distribution $P(X, Y)$ can be factorized in two Markov equivalent DAGs $X \rightarrow Y$ or $Y \rightarrow X$ as $P(X, Y) = P(X)P(Y|X)$ and $P(X, Y) = P(Y)P(X|Y)$. Then the CGNN can reproduce equally well the observational distribution in both directions (under the assumption of Proposition 1). We refer the reader to Zhang and Hyvärinen (2009) for more details on this problem of identifiability in the bivariate case.

As shown in Sect. 4.3.3, the proposed approach enforces the discovery of causal models in the Markov equivalence class. Within this class, the non-identifiability issue is empirically mitigated by restricting the class of CGNNs considered, and specifically limiting the number n_h of hidden neurons in each causal mechanism (Eq. 6). Formally, we restrict ourselves to the sub-class of CGNNs, noted $\mathscr{C}_{\hat{\mathscr{G}},\hat{f}^{n_h}} = (\widehat{\mathscr{G}}, \hat{f}^{n_h}, \mathscr{E})$ with exactly n_h hidden neurons in each $\hat{f}_i$ mechanism. Accordingly, any candidate $\hat{\mathscr{G}}$ with number of edges $|\hat{\mathscr{G}}|$ involves the same number of parameters: $(2d + |\hat{\mathscr{G}}|) \times n_h$ weights and $d \times (n_h + 1)$ bias parameters. As shown experimentally in Sect. 5, this parameter n_h is crucial as it governs the CGNN ability to model the causal mechanisms: too small n_h, and data patterns may be missed; too large n_h, and overly complicated causal mechanisms may be retained.

4.3 Model Optimization

Model optimization consists at finding a (nearly) optimum solution $(\hat{\mathscr{G}}, \hat{f})$ in the sense of the score defined in the previous section. The so-called *parametric* optimization of the CGNN, where structure estimate $\hat{\mathscr{G}}$ is fixed and the goal is to find the best neural estimates $\hat{f}$ conditionally to $\hat{\mathscr{G}}$ is tackled in Sect. 4.3.1. The *non-parametric* optimization, aimed at finding the best structure estimate, is considered in Sect. 4.3.2. In Sect. 4.3.3, we present an identifiability result for CGNN up to Markov equivalence classes.

4.3.1 Parametric (Weight) Optimization

Given the acyclic structure estimate $\hat{\mathscr{G}}$, the neural networks $\hat{f}_1, \ldots, \hat{f}_d$ of the CGNN are learned end-to-end using backpropagation with Adam optimizer (Kingma and Ba 2014) by minimizing losses $\widehat{\mathrm{MMD}}_k$ (Eq. (8), referred to as **CGNN** ($\widehat{\mathrm{MMD}}_k$)) or $\widehat{\mathrm{MMD}}_k^m$ (see section "The Maximum Mean Discrepancy (MMD) Statistic" in Appendix, **CGNN** ($\widehat{\mathrm{MMD}}_k^m$)).

The procedure closely follows that of supervised continuous learning (regression), except for the fact that the loss to be minimized is the MMD loss instead of the mean squared error. Neural nets $\hat{f}_i$, $i \in [[1, d]]$ are trained during n_{train} epochs, where the noise samples, independent and identically distributed, are drawn in each epoch. In the $\widehat{\mathrm{MMD}}_k^m$ variant, the parameters of the random kernel are resampled from their respective distributions in each training epoch (see section "The Maximum Mean Discrepancy (MMD) Statistic" in Appendix). After training, the score is computed and averaged over n_{eval} estimated samples of size n. Likewise, the noise samples are re-sampled anew for each evaluation sample. The overall process with training and evaluation is repeated nb_{run} times to reduce stochastic effects relative to random initialization of neural network weights and stochastic gradient descent.

4.3.2 Non-parametric (Structure) Optimization

The number of directed acyclic graphs $\hat{\mathscr{G}}$ over d nodes is super-exponential in d, making the non-parametric optimization of the CGNN structure an intractable computational and statistical problem. Taking inspiration from Tsamardinos et al. (2006); Nandy et al. (2015), we start from a graph skeleton recovered by other methods such as feature selection (Yamada et al. 2014). We focus on optimizing the edge orientations. Letting L denote the number of edges in the graph, it defines a combinatorial optimization problem of complexity $\mathscr{O}(2^L)$ (note however that not all orientations are admissible since the eventual oriented graph must be a DAG).

The motivation for this approach is to decouple the edge selection task and the causal modeling (edge orientation) tasks, and enable their independent assessment.

Any $X_i - X_j$ edge in the graph skeleton stands for a direct dependency between variables X_i and X_j. Given Causal Markov and Faithfulness assumptions, such a direct dependency either reflects a direct causal relationship between the two variables ($X_i \rightarrow X_j$ or $X_i \leftarrow X_j$), or is due to the fact that X_i and X_j admit a latent (unknown) common cause ($X_i \leftrightarrow X_j$). Under the assumption of *causal sufficiency*, the latter does not hold. Therefore the $X_i - X_j$ link is associated with a causal relationship in one or the other direction. The causal sufficiency assumption will be relaxed in Sect. 6.

The edge orientation phase proceeds as follows:

- Each $X_i - X_j$ edge is first considered in isolation, and its orientation is evaluated using CGNN. Both score $S(\mathscr{C}_{X_i \to X_j, \hat{f}}, \mathscr{D}_{ij})$ and $S(\mathscr{C}_{X_j \to X_i, \hat{f}}, \mathscr{D}_{ij})$ are computed, where $\mathscr{D}_{ij} = \{[x_{i,l}, x_{j,l}]\}_{l=1}^{n}$. The best orientation corresponding to a minimum score is retained. After this step, an initial graph is built with complexity $2L$ with L the number of edges in the skeleton graph.
- The initial graph is revised to remove all cycles. Starting from a set of random nodes, all paths are followed iteratively until all nodes are reached; an edge pointing toward an already visited node and forming a cycle is reversed. The resulting DAG is used as initial DAG for the structured optimization, below.
- The optimization of the DAG structure is achieved using a hill-climbing algorithm aimed to optimize the global score $S(\mathscr{C}_{\hat{\mathscr{G}}, \hat{f}}, \mathscr{D})$. Iteratively, (1) an edge $X_i - X_j$ is uniformly randomly selected in the current graph; (2) the graph obtained by reversing this edge is considered (if it is still a DAG and has not been considered before) and the associated global CGNN is retrained; (3) if this graph obtains a lower global score than the former one, it becomes the current graph and the process is iterated until reaching a (local) optimum. More sophisticated combinatorial optimization approaches, e.g. Tabu search, will be considered in further work. In this paper, hill-climbing is used for a proof of concept of the proposed approach, achieving a decent trade-off between computational time and accuracy.

At the end of the process each causal edge $X_i \to X_j$ in $\mathscr{G}$ is associated with a score, measuring its contribution to the global score:

$$S_{X_i \to X_j} = S(\mathscr{C}_{\hat{\mathscr{G}} - \{X_i \to X_j\}, \hat{f}}, \mathscr{D}) - S(\mathscr{C}_{\hat{\mathscr{G}}, \hat{f}}, \mathscr{D}) \tag{10}$$

During the structure (non-parametric) optimization, the graph skeleton is fixed; no edge is added or removed. The penalization term $\lambda|\widehat{\mathscr{G}}|$ entering in the score evaluation (Eq. 7) can thus be neglected at this stage and only the MMD-losses are used to compare two graphs. The penalization term will be used in Sect. 6 to compare structures with different skeletons, as the potential confounding factors will be dealt with by removing edges.

4.3.3 Identifiability of CGNN up to Markov Equivalence Classes

Assuming an infinite number of observational data, and assuming further that the generative distribution belongs to the CGNN class $\mathscr{C}_{\mathscr{G}, f}$, then there exists a DAG reaching an MMD score of 0 in the Markov equivalence class of $\mathscr{G}$:

Proposition 4 *Let $X = [X_1, \ldots, X_d]$ denote a set of continuous random variables with joint distribution P, generated by a CGNN $\mathscr{C}_{\mathscr{G}, f} = (\mathscr{G}, f, \mathscr{E})$ with $\mathscr{G}$ a directed acyclic graph. Let $\mathscr{D}$ be an infinite observational sample generated from this CGNN. We assume that P is Markov and faithful to the graph $\mathscr{G}$, and that every pair of* *variables* (X_i, X_j) *that are d-connected in the graph are not independent. We note* $\widehat{\mathscr{D}}$ *an infinite sample generated by a candidate CGNN,* $\mathscr{C}_{\widehat{\mathscr{G}},\hat{f}} = (\widehat{\mathscr{G}}, \hat{f}, \mathscr{E})$. *Then,*

(i) If $\widehat{\mathscr{G}} = \mathscr{G}$ *and* $\hat{f} = f$, *then* $\widehat{MMD}_k(\mathscr{D}, \widehat{\mathscr{D}}) = 0$.
(ii) For any graph $\widehat{\mathscr{G}}$ *characterized by the same adjacencies but not belonging to the Markov equivalence class of* $\mathscr{G}$, *for all* $\hat{f}$, $\widehat{MMD}_k(\mathscr{D}, \widehat{\mathscr{D}}) \neq 0$.

Proof In section "Proofs" in Appendix.

This result does not establish the CGNN identifiability within the Markov class of equivalence, that is left for future work. As shown experimentally in Sect. 5.1, there is a need to control the model capacity in order to recover the directed graph in the Markov equivalence class.[4]

5 Experiments

This section reports on the empirical validation of CGNN compared to the state of the art under the no confounding assumption. The experimental setting is first discussed. Thereafter, the results obtained in the bivariate case, where only asymmetries in the joint distribution can be used to infer the causal relationship, are discussed. The variable triplet case, where conditional independence can be used to uncover causal orientations, and the general case of $d > 2$ variables are finally considered. All computational times are measured on Intel Xeon 2.7 Ghz (CPU) or on Nvidia GTX 1080Ti graphics card (GPU).

5.1 Experimental Setting

The CGNN architecture is a 1-hidden layer network with ReLU activation function. The multi-scale Gaussian kernel used in the MMD scores has bandwidth γ ranging in $\{0.005, 0.05, 0.25, 0.5, 1, 5, 50\}$. The number nb_{run} used to average the score is set to 32 for CGNN-MMD (respectively 64 for CGNN-Fourier). In this section the distribution $\mathscr{E}$ of the noise variables is set to $\mathscr{N}(0, 1)$. The number n_h of neurons in the hidden layer, controlling the identifiability of the model, is the most sensitive hyper-parameter of the presented approach. Preliminary experiments are conducted to adjust its range, as follows. A 1500 sample dataset is generated from the linear structural equation model with additive uniform noise $Y = X + \mathscr{U}(0, 0.5)$, $X \sim U([-2, 2])$ (Fig. 5). Both CGNNs associated to $X \rightarrow Y$ and $Y \rightarrow X$ are trained

[4] In some specific cases, such as in the bivariate linear FCM with Gaussian noise and Gaussian input, even by restricting the class of functions considered, the DAG cannot be identified from purely observational data (Mooij et al. 2016).

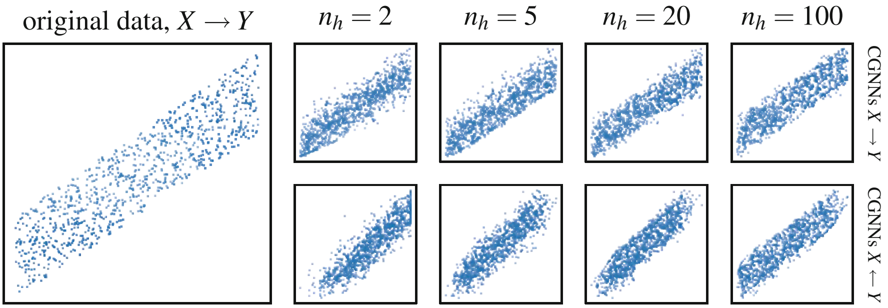
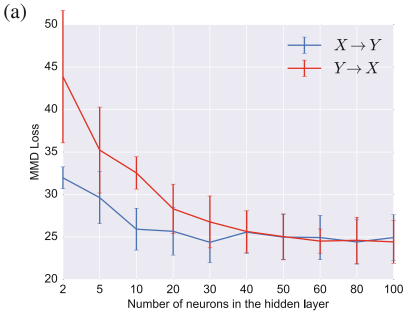


Fig. 5 Leftmost: Data samples. Columns 2–5: Estimate samples generated from CGNN with direction $X \to Y$ (top row) and $Y \to X$ (bottom row) for number of hidden neurons $n_h = 2, 5, 20, 100$



(b)

n_h	$C_{X\to Y}$	$C_{Y\to X}$	Diff.
2	32.0	43.9	11.9^{***}
5	29.6	35.2	5.6^{***}
10	25.9	32.5	6.6^{***}
20	25.7	28.3	2.6^{***}
30	24.4	26.8	2.4^{***}
40	25.6	25.6	0.7
50	25.0	25.0	0.6
100	24.9	24.4	−0.5

Fig. 6 CGNN sensitivity w.r.t. the number of hidden neurons n_h: scores associated to both causal models (average and standard deviation over 32 runs). (**a**) $C_{X\to Y}$, $C_{Y\to X}$ with various n_h values. (**b**) Scores $C_{X\to Y}$ and $C_{Y\to X}$ with their difference. *** denotes the significance at the 0.001 threshold with the t-test

until reaching convergence ($n_{epoch} = 1000$) using Adam (Kingma and Ba 2014) with a learning rate of 0.01 and evaluated over $n_{eval} = 500$ generated samples. The distributions generated from both generative models are displayed on Fig. 5 for $n_h = 2, 5, 20, 100$. The associated scores (averaged on 32 runs) are displayed on Fig. 6a, confirming that the model space must be restricted for the sake of identifiability (cf. Sect. 4.3.3 above).

5.2 *Learning Bivariate Causal Structures*

As said, under the no-confounder assumption a dependency between variables X and Y exists iff either X causes Y ($Y = f(X, E)$) or Y causes X ($X = f(Y, E)$). The identification of a *Bivariate Structural Causal Model* is based on comparing the model scores (Sect. 4.2) attached to both CGNNs.

5.2.1 Benchmarks

Five datasets with continuous variables are considered[5]:

- **CE-Cha**: 300 continuous variable pairs from the cause effect pair challenge (Guyon 2013), restricted to pairs with label $+1$ $(X \rightarrow Y)$ and -1 $(Y \rightarrow X)$.
- **CE-Net**: 300 artificial pairs generated with a neural network initialized with random weights and random distribution for the cause (exponential, gamma, lognormal, laplace…).
- **CE-Gauss**: 300 artificial pairs without confounder sampled with the generator of Mooij et al. (2016): $Y = f_Y(X, E_Y)$ and $X = f_X(E_X)$ with $E_X \sim p_{E_X}$ and $E_Y \sim p_{E_Y}$. p_{E_X} and p_{E_Y} are randomly generated Gaussian mixture distributions. Causal mechanism f_X and f_Y are randomly generated Gaussian processes.
- **CE-Multi**: 300 artificial pairs generated with linear and polynomial mechanisms. The effect variables are built with post additive noise setting $(Y = f(X) + E)$, post multiplicative noise $(Y = f(X) \times E)$, pre-additive noise $(Y = f(X + E))$ or pre-multiplicative noise $(Y = f(X \times E))$.
- **CE-Tueb**: 99 real-world cause-effect pairs from the *Tuebingen cause-effect pairs* dataset, version August 2016 (Mooij et al. 2016). This version of this dataset is taken from 37 different data sets coming from various domain: climate, census, medicine data.

For all variable pairs, the size n of the data sample is set to 1500 for the sake of an acceptable overall computational load.

5.2.2 Baseline Approaches

CGNN is assessed comparatively to the following algorithms[6]: (1) ANM (Mooij et al. 2016) with Gaussian process regression and HSIC independence test of the residual; (2) a pairwise version of LiNGAM (Shimizu et al. 2006) relying on Independent Component Analysis to identify the linear relations between variables; (3) IGCI (Daniusis et al. 2012) with entropy estimator and Gaussian reference measure; (4) the post-nonlinear model (PNL) with HSIC test (Zhang and Hyvärinen 2009); (5) GPI-MML (Stegle et al. 2010); where the Gaussian process regression with higher marginal likelihood is selected as causal direction; (6) CDS, retaining the causal orientation with lowest variance of the conditional probability distribution; (7) Jarfo (Fonollosa 2016), using a random forest causal classifier trained from the ChaLearn Cause-effect pairs on top of 150 features including ANM, IGCI, CDS, LiNGAM, regressions, HSIC tests.

[5] The first four datasets are available at http://dx.doi.org/10.7910/DVN/3757KX. The *Tuebingen cause-effect pairs* dataset is available at https://webdav.tuebingen.mpg.de/cause-effect/.

[6] Using the R program available at https://github.com/ssamot/causality for ANM, IGCI, PNL, GPI and LiNGAM.

5.2.3 Hyper-Parameter Selection

For a fair comparison, a leave-one-dataset-out procedure is used to select the key best hyper-parameter for each algorithm. To avoid computational explosion, a single hyper-parameter per algorithm is adjusted in this way; other hyper-parameters are set to their default value. For CGNN, n_h ranges over $\{5, \ldots, 100\}$. The leave-one-dataset-out procedure sets this hyper-parameter n_h to values between 20 and 40 for the different datasets. For ANM and the bivariate fit, the kernel parameter for the Gaussian process regression ranges over $\{0.01, \ldots, 10\}$. For PNL, the threshold parameter alpha for the HSIC independence test ranges over $\{0.0005, \ldots, 0.5\}$. For CDS, the $ffactor$ involved in the discretization step ranges over [[1, 10]]. For GPI-MML, its many parameters are set to their default value as none of them appears to be more critical than others. Jarfo is trained from 4000 variable pairs datasets with same generator used for **CE-Cha-train**, **CE-Net-train**, **CE-Gauss-train** and **CE-Multi-train**; the causal classifier is trained on all datasets except the test set.

5.2.4 Empirical Results

Figure 7 reports the area under the precision/recall curve for each benchmark and all algorithms.

Methods based on simple regression like the bivariate fit and Lingam are outperformed as they underfit the data generative process. CDS and IGCI obtain very good results on few datasets. Typically, IGCI takes advantage of some specific features of the dataset, (e.g. the cause entropy being lower than the effect entropy in **CE-Multi**), but remains at chance level otherwise. ANM-HSIC yields good results when the additive assumption holds (e.g. on **CE-Gauss**), but fails otherwise. PNL, less restrictive than ANM, yields overall good results compared to the former methods. Jarfo, a voting procedure, can in principle yield the best of the above methods and does obtain good results on artificial data. However, it does not perform well on the real dataset **CE-Tueb**; this counter-performance is blamed on the differences between all five benchmark distributions and the lack of generalization/transfer learning.

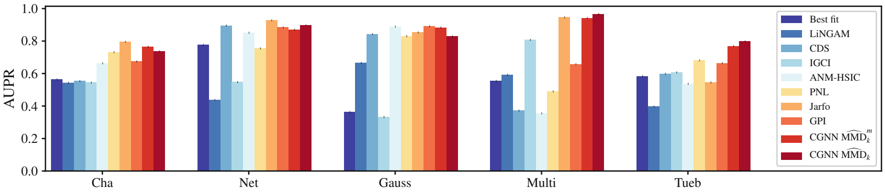


Fig. 7 Bivariate causal modelling: area under the precision/recall curve for the five datasets. A full table of the scores is given on Table 3 in section "Table of Scores for the Experiments on Cause-Effect Pairs" in Appendix

Lastly, generative methods GPI and **CGNN** ($\widehat{\text{MMD}}_k$) perform well on most datasets, including the real-world cause-effect pairs CE-Tüb, in counterpart for a higher computational cost (resp. 32 min on CPU for GPI and 24 min on GPU for CGNN). Using the linear MMD approximation Lopez-Paz (2016), **CGNN** ($\widehat{\text{MMD}}_k^m$ as explained in section "The Maximum Mean Discrepancy (MMD) Statistic" in Appendix reduces the cost by a factor of 5 without hindering the performance.

Overall, CGNN demonstrates competitive performance on the cause-effect inference problem, where it is necessary to discover distributional asymmetries.

5.3 Identifying v-structures

A second series of experiments is conducted to investigate the method performances on variable triplets, where multivariate effects and conditional variable independence must be taken into account to identify the Markov equivalence class of a DAG. The considered setting is that of variable triplets (A, B, C) in the linear Gaussian case, where asymmetries between cause and effect cannot be exploited (Shimizu et al. 2006) and conditional independence tests are required. In particular strict pairwise methods can hardly be used due to un-identifiability (as each pair involves a linear mechanism with Gaussian input and additive Gaussian noise) (Hoyer et al. 2009).

With no loss of generality, the graph skeleton involving variables (A, B, C) is $A - B - C$. All three causal models (up to variable renaming) based on this skeleton are used to generate 500-sample datasets, where the random noise variables are independent centered Gaussian variables.

Given skeleton $A - B - C$, each dataset is used to model the possible four CGNN structures (Fig. 8, with generative SEMs):

- Chain structures ABC $(A = f_1(E_1), B = f_2(A, E_2)\ , C = f_3(B, E_3)$ and CBA $(C = f_1(E_1), B = f_2(C, E_2)\ , A = f_3(B, E_3))$
- V structure: $A = f_1(E_1), C = f_2(E_2)\ , B = f_3(A, C, E_3)$
- reversed V structure: $B = f_1(E_1), A = f_2(B, E_2)\ , C = f_3(B, E_3)$

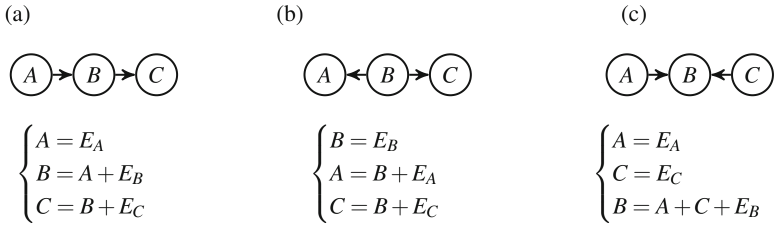


Fig. 8 Datasets generated from the three DAG configurations with skeleton $A - B - C$. (**a**) Chain structure. (**b**) Reversed v-structure. (**c**) V-structure

Table 1 CGNN-MMD scores for all models on all datasets

	Non v-structures		v-structure
Score	Chain str.	Reversed v-str.	v-structure
C_{ABC}	*0.122 (0.009)*	*0.124 (0.007)*	0.172 (0.005)
C_{CBA}	*0.121 (0.006)*	*0.127 (0.008)*	0.171 (0.004)
$C_{reversedV}$	*0.122 (0.007)*	*0.125 (0.006)*	0.172 (0.004)
$C_{Vstructure}$	0.202 (0.004)	0.180 (0.005)	**0.127** (0.005)

Smaller scores indicate a better match. CGNN correctly identifies v-structure vs. other structures. Bold value corresponds to best match for v-structure

Let C_{ABC}, C_{CBA}, $C_{v-structure}$ and $C_{reversedV}$ denote the scores of the CGNN models respectively attached to these structures. The scores computed on all three datasets are displayed in Table 1 (average over 64 runs; the standard deviation is indicated in parenthesis).

CGNN scores support a clear and significant discrimination between the V-structure and all other structures (noting that the other structures are Markov equivalent and thus can hardly be distinguished).

This second series of experiments thus shows that CGNN can effectively detect, and take advantage of, conditional independence between variables.

5.4 *Multivariate Causal Modeling Under Causal Sufficiency Assumption*

Let $\mathbf{X} = [X_1, \ldots, X_d]$ be a set of continuous variables, satisfying the Causal Markov, faithfulness and causal sufficiency assumptions. To that end, all experiments provide all algorithms *the true graph skeleton*, so their ability to orient edges is compared in a fair way. This allows us to separate the task of orienting the graph from that of uncovering the skeleton.

5.4.1 Results on Artificial Graphs with Additive and Multiplicative Noises

We draw 500 samples from 20 training artificial causal graphs and 20 test artificial causal graphs on 20 variables. Each variable has a number of parents uniformly drawn in $[[0, 5]]$; f_is are randomly generated polynomials involving additive/multiplicative noise.[7]

We compare CGNN to the PC algorithm (Spirtes et al. 1993), the score-based methods GES (Chickering 2002), LiNGAM (Shimizu et al. 2006), causal additive

[7] The data generator is available at https://github.com/GoudetOlivie/CGNN. The datasets considered are available at http://dx.doi.org/10.7910/DVN/UZMB69.

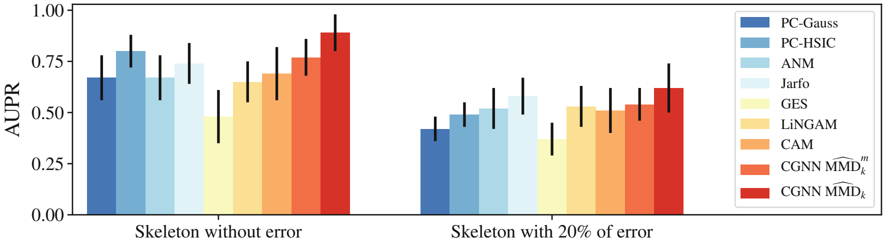


Fig. 9 Average (std. dev.) AUPR results for the orientation of 20 artificial graphs given true skeleton (left) and artificial graphs given skeleton with 20% error (right). A full table of the scores, including the metrics Structural Hamming Distance (SHD) and Structural Intervention (SID) (Peters and Bühlmann 2013) is shown on Table 4 in section "Table of Scores for the Experiments on Graphs" in Appendix

model (CAM) (Bühlmann et al. 2014) and with the pairwise methods ANM and Jarfo. For PC, we employ the better-performing, order-independent version of the PC algorithm proposed by Colombo and Maathuis (2014). PC needs the specification of a conditional independence test. We compare PC-Gaussian, which employs a Gaussian conditional independence test on Fisher z-transformations, and PC-HSIC, which uses the HSIC conditional independence test with the Gamma approximation (Gretton et al. 2005). PC and GES are implemented in the *pcalg* package (Kalisch et al. 2012).

All hyperparameters are set on the training graphs in order to maximize the Area Under the Precision/Recall score (AUPR). For the Gaussian conditional independence test and the HSIC conditional independence test, the significance level achieving best result on the training set are respectively 0.1 and 0.05 . For GES, the penalization parameter is set to 3 on the training set. For CGNN, n_h is set to 20 on the training set. For CAM, the cutoff value is set to 0.001.

Figure 9 (left) displays the performance of all algorithms obtained by starting from the exact skeleton on the test set of artificial graphs and measured from the AUPR (Area Under the Precision/Recall curve), the Structural Hamming Distance (SHD, the number of edge modifications to transform one graph into another) and the Structural Intervention Distance (SID, the number of equivalent two-variable interventions between two graphs) (Peters and Bühlmann 2013).

CGNN obtains significant better results with SHD and SID compared to the other algorithms when the task is to discover the causal from the true skeleton. One resulting graph is shown on Fig. 10. There are three mistakes on this graph (red edges) (in lines with an SHD on average of 2.5).

Constraints based method PC with powerful HSIC conditional independence test is the second best performing method. It highlights the fact that when the skeleton is known, exploiting the structure of the graph leads to good results compared to pairwise methods using only local information. Notably, as seen on Fig. 10, this type of DAG has a lot of v-structures, as many nodes have more than one parent in the graph, but this is not always the case as shown in the next subsection.

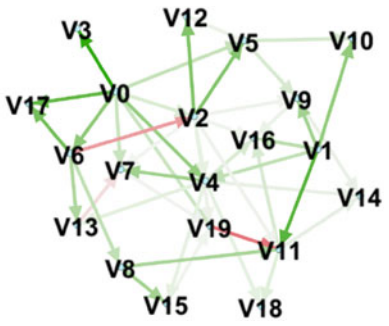


Fig. 10 Orientation by CGNN of artificial graph with 20 nodes. Green edges are good orientation and red arrows false orientation. Three edges are red and 42 are green. The strength of the line refers to the confidence of the algorithm

Overall CGNN and PC-HSIC are the most computationally expensive methods, taking an average of 4 h on GPU and 15 h on CPU, respectively.

The robustness of the approach is validated by randomly perturbing 20% edges in the graph skeletons provided to all algorithms (introducing about 10 false edges over 50 in each skeleton). As shown on Table 4 (right) in Appendix, and as could be expected, the scores of all algorithms are lower when spurious edges are introduced. Among the least robust methods are constraint-based methods; a tentative explanation is that they heavily rely on the graph structure to orient edges. By comparison pairwise methods are more robust because each edge is oriented separately. As CGNN leverages conditional independence but also distributional asymmetry like pairwise methods, it obtains overall more robust results when there are errors in the skeleton compared to PC-HSIC. However one can notice that a better SHD score is obtained by CAM, on the skeleton with 20% error. This is due to the exclusive last edge pruning step of CAM, which removes spurious links in the skeleton.

CGNN obtains overall good results on these artificial datasets. It offers the advantage to deliver a full generative model useful for simulation (while e.g., Jarfo and PC-HSIC only give the causality graph). To explore the scalability of the approach, five artificial graphs with 100 variables have been considered, achieving an AUPRC of 85.5 ± 4, in 30 h of computation on four NVIDIA 1080Ti GPUs.

5.4.2 Result on Biological Data

We now evaluate CGNN on biological networks. First we apply it on simulated gene expression data and then on real protein network.

Syntren Artificial Simulator

First we apply CGNN on SynTREN (Van den Bulcke et al. 2006) from sub-networks of E. coli (Shen-Orr et al. 2002). SynTREN creates synthetic transcriptional regulatory networks and produces simulated gene expression data that approximates

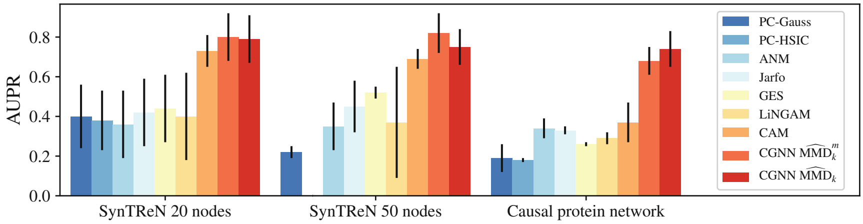


Fig. 11 Average (std. dev.) AUPR results for the orientation of 20 artificial graphs generated with the SynTReN simulator with 20 nodes (left), 50 nodes (middle), and real protein network given true skeleton (right). A full table of the scores, including the metrics Structural Hamming Distance (SHD) and Structural Intervention (SID) (Peters and Bühlmann 2013) is included in section "Table of Scores for the Experiments on Graphs" in Appendix

experimental data. Interaction kinetics are modeled by complex mechanisms based on Michaelis-Menten and Hill kinetics (Mendes et al. 2003).

With Syntren, we simulate 20 subnetworks of 20 nodes and 5 subnetworks with 50 nodes. For the sake of reproducibility, we use the random seeds of 0, 1 . . . 19 and 0, 1 . . . 4 for each graph generation with respectively 20 nodes and 50 nodes. The default Syntren parameters are used: a probability of 0.3 for complex 2-regulator interactions and a value of 0.1 for Biological noise, experimental noise and Noise on correlated inputs. For each graph, Syntren give us expression datasets with 500 samples.

Figure 11 (left and middle) and Table 5 in section "Table of Scores for the Experiments on Graphs" in Appendix display the performance of all algorithms obtained by starting from the exact skeleton of the causal graph with same hyper-parameters as in the previous subsection. As a note, we canceled the PC-HSIC algorithm after 50 h of running time.

Constraint based methods obtain low score on this type of graph dataset. It may be explained by the type of structure involved. Indeed as seen of Fig. 12, there are very few v-structures in this type of network, making impossible the orientation of an important number of edges by using only conditional independence tests. Overall the methods CAM and CGNN that take into account of both distributional asymmetry and multivariate interactions, get the best scores. CGNN obtain the best results in AUPR, SHD and SID for graph with 20 nodes and 50 nodes, showing that this method can be used to infer networks having complex distribution, complex causal mechanisms and interactions. The Fig. 12 shows the resulting graph obtain with CGNN. Edges with good orientation are displayed in green and edge with false orientation in red.

5.4.3 Results on Biological Real-World Data

CGNN is applied to the protein network problem Sachs et al. (2005), using the Anti-CD3/CD28 dataset with 853 observational data points corresponding to general

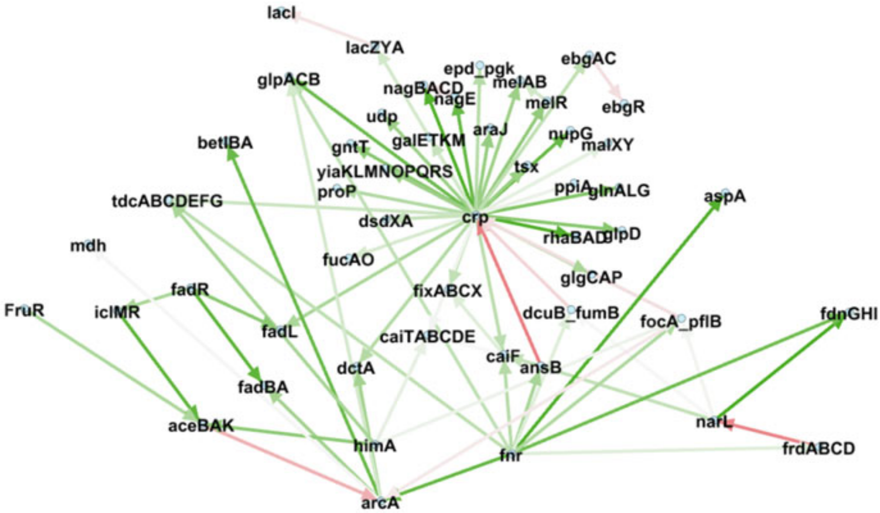


Fig. 12 Orientation by CGNN of E. coli subnetwork with 50 nodes and corresponding to Syntren simulation with random seed 0. Green edges are good orientation and red arrows false orientation. The strength of the line refers to the confidence of the algorithm

perturbations without specific interventions. All algorithms were given the skeleton of the causal graph (Sachs et al. 2005, Fig. 2) with same hyper-parameters as in the previous subsection. We run each algorithm on 10-fold cross-validation. Table 6 in Appendix reports average (std. dev.) results.

Constraint-based algorithms obtain surprisingly low scores, because they cannot identify many v-structures in this graph. We confirm this by evaluating conditional independence tests for the adjacent tuples of nodes *pip3-akt-pka*, *pka-pmek-pkc*, *pka-raf-pkc* and we do not find strong evidences for v-structure. Therefore methods based on distributional asymmetry between cause and effect seem better suited to this dataset. CGNN obtains good results compared to the other algorithms. Notably, Fig. 13 shows that CGNN is able to recover the strong signal transduction pathway *raf*→*mek*→*erk* reported in Sachs et al. (2005) and corresponding to clear direct enzyme-substrate causal effect. CGNN gives important scores for edges with good orientation (green line), and low scores (thinnest edges) to the wrong edges (red line), suggesting that false causal discoveries may be controlled by using the confidence scores defined in Eq. (10).

6 Towards Predicting Confounding Effects

In this subsection we propose an extension of our algorithm relaxing the causal sufficiency assumption. We are still assuming the Causal Markov and faithfulness assumptions, thus three options have to be considered for each edge (X_i, X_j) of the

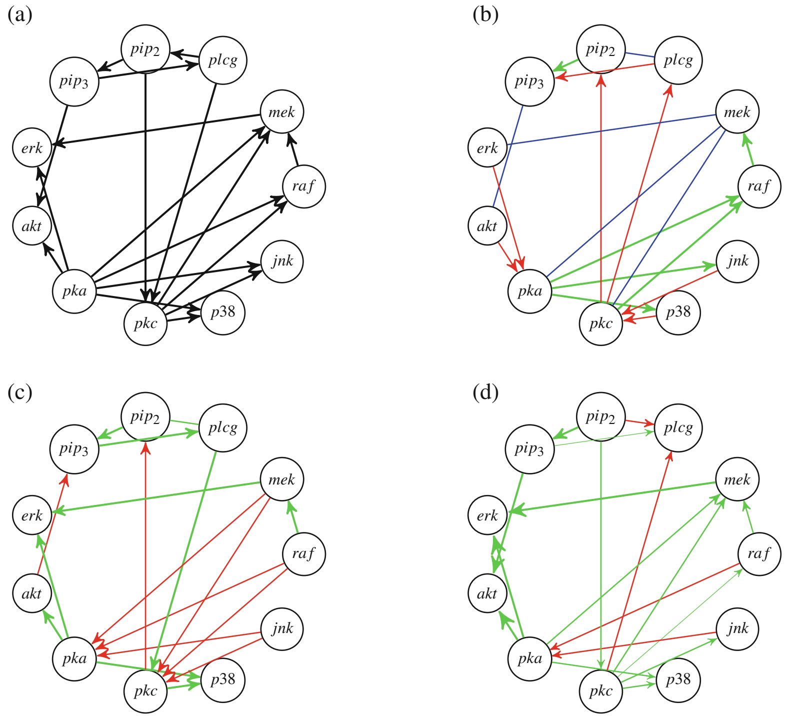


Fig. 13 Causal protein network. (**a**) Ground truth. (**b**) GES. (**c**) CAM. (**d**) CGNN

skeleton representing a direct dependency: $X_i \to X_j$, $X_j \to X_i$ and $X_i \leftrightarrow X_j$ (both variables are consequences of common hidden variables).

6.1 Principle

Hidden common causes are modeled through correlated random noise. Formally, an additional noise variable $E_{i,j}$ is associated to each $X_i - X_j$ edge in the graph skeleton.

We use such new models with correlated noise to study the robustness of our graph reconstruction algorithm to increasing violations of causal sufficiency, by occluding variables from our datasets. For example, consider the FCM on $\mathbf{X} = [X_1, \ldots, X_5]$ that was presented on Fig. 1. If variable X_1 would be missing from data, the correlated noise $E_{2,3}$ would be responsible for the existence of a double headed arrow connection $X_2 \leftrightarrow X_3$ in the skeleton of our new type of model. The resulting FCM is shown in Fig. 14. Notice that direct causal effects such as $X_3 \to X_5$ or $X_4 \to X_5$ may persist, even in presence of possible confounding effects.

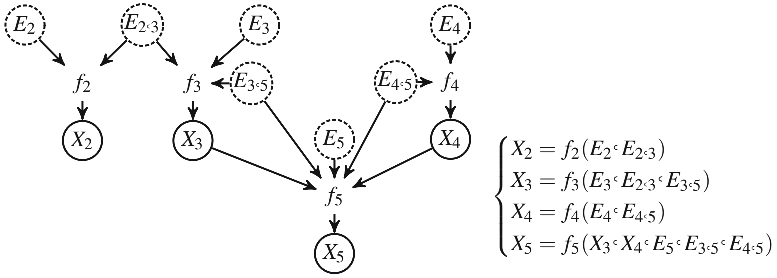


Fig. 14 The Functional Causal Model (FCM) on $\mathbf{X} = [X_1, \ldots, X_5]$ with the missing variable X_1

Formally, given a graph skeleton $\mathscr{S}$, the FCM with correlated noise variables is defined as:

$$X_i \leftarrow f_i(X_{\text{Pa}(i;\mathscr{G})}, E_i, E_{\text{Ne}(i;\mathscr{S})}), \tag{11}$$

where $\text{Ne}(i; \mathscr{S})$ is the set of indices of all the variables adjacent to variable X_i in the skeleton $\mathscr{S}$.

One can notice that this model corresponds to the most general formulation of the FCM with potential confounders for each pair of variables in a given skeleton (representing direct dependencies) where each random variable $E_{i,j}$ summarizes all the unknown influences of (possibly multiple) hidden variables influencing the two variables X_i and X_j.

Here we make a clear distinction between the directed acyclic graph denoted $\mathscr{G}$ and the skeleton $\mathscr{S}$. Indeed, due to the presence of confounding correlated noise, any variable in $\mathscr{G}$ can be removed without altering $\mathscr{S}$. We use the same generative neural network to model the new FCM presented in Eq. (11). The difference is the new noise variables having effect on pairs of variables simultaneously. However, since the correlated noise FCM is still defined over a directed acyclic graph $\mathscr{G}$, the functions $\hat{f}_1, \ldots, \hat{f}_d$ of the model, which we implement as neural networks, the model can still be learned end-to-end using backpropagation based on the CGNN loss.

All edges are evaluated with these correlated noises, the goal being to see whether introducing a correlated noise explains the dependence between the two variables X_i and X_j.

As mentioned before, the score used by CGNN is:

$$S(\mathscr{C}_{\hat{\mathscr{G}},\hat{f}}, \mathscr{D}) = \widehat{\text{MMD}}_k(\mathscr{D}, \widehat{\mathscr{D}}) + \lambda|\widehat{\mathscr{G}}| \tag{12}$$

where $|\widehat{\mathscr{G}}|$ is the total number of edges in the DAG. In the graph search, for any given edge, we compare the score associated to the graph considered with and without this edge. If the contribution of this edge is negligible compared to a given threshold lambda, the edge is considered as spurious.

The non-parametric optimization of the $\hat{\mathscr{G}}$ structure is also achieved using a Hill-Climbing algorithm; in each step an edge of $\mathscr{S}$ is randomly drawn and modified in $\hat{\mathscr{G}}$ using one out of the possible three operators: reverse the edge, add an edge and remove an edge. Other algorithmic details are as in Sect. 4.3.2: the greedy search optimizes the penalized loss function (Eq. 12). For CGNN, we set the hyperparameter $\lambda = 5 \times 10^{-5}$ fitted on the training graph dataset.

The algorithm stops when no improvement is obtained. Each causal edge $X_i \rightarrow X_j$ in $\mathscr{G}$ is associated with a score, measuring its contribution to the global score:

$$S_{X_i \rightarrow X_j} = S(\mathscr{C}_{\hat{\mathscr{G}}-\{X_i \rightarrow X_j\}, \hat{f}}, \mathscr{D}) - S(\mathscr{C}_{\hat{\mathscr{G}}, \hat{f}}, \mathscr{D}) \tag{13}$$

Missing edges are associated with a score 0.

6.2 *Experimental Validation*

6.2.1 Benchmarks

The empirical validation of this extension of CGNN is conducted on same benchmarks as in Sect. 5.4 ($\mathscr{G}_i$, $i \in [[2, 5]]$), where three variables (causes for at least two other variables in the graph) have been randomly removed.[8] The true graph skeleton is augmented with edges $X - Y$ for all X, Y that are consequences of a same removed cause. All algorithms are provided with the same graph skeleton for a fair comparison. The task is to both orient the edges in the skeleton, and remove the spurious direct dependencies created by latent causal variables.

6.2.2 Baselines

CGNN is compared with state of art methods: (1) constraint-based RFCI (Colombo et al. 2012), extending the PC method equipped with Gaussian conditional independence test (RFCI-Gaussian) and the gamma HSIC conditional independence test (Gretton et al. 2005) (RFCI-HSIC). We use the order-independent constraint-based version proposed by Colombo and Maathuis (2014) and the majority rules for the orientation of the edges. For CGNN, we set the hyperparameter $\lambda = 5 \times 10^{-5}$ fitted on the training graph dataset. Jarfo is trained on the 16,200 pairs of the cause-effect pair challenge (Guyon 2013, 2014) to detect for each pair of variable if $X_i \rightarrow Y_i$, $Y_i \rightarrow X_i$ or $X_i \leftrightarrow Y_i$.

[8] The datasets considered are available at http://dx.doi.org/10.7910/DVN/UZMB69.

Table 2 AUPR, SHD and SID on causal discovery with confounders

Method	AUPR	SHD	SID
RFCI-Gaussian	0.22 (0.08)	21.9 (7.5)	174.9 (58.2)
RFCI-HSIC	0.41 (0.09)	17.1 (6.2)	124.6 (52.3)
Jarfo	0.54 (0.21)	20.1 (14.8)	98.2 (49.6)
CGNN ($\widehat{\text{MMD}}_k$)	0.71[a] (0.13)	11.7[a] (5.5)	53.55[a] (48.1)

[a]Denotes significance at $p = 10^{-2}$

6.2.3 Results

Comparative performances are shown in Table 2, reporting the area under the precision/recall curve. Overall, these results confirm the robustness of the CGNN proposed approach w.r.t. confounders, and its competitiveness w.r.t. RFCI with powerful conditional independence test (RFCI-HSIC). Interestingly, the effective causal relations between the visible variables are associated with a high score; spurious links due to hidden latent variables get a low score or are removed.

7 Discussion and Perspectives

This paper introduces CGNN, a new framework and methodology for functional causal model learning, leveraging the power and non-parametric flexibility of Generative Neural Networks.

CGNN seamlessly accommodates causal modeling in presence of confounders, and its extensive empirical validation demonstrates its merits compared to the state of the art on medium-size problems. We believe that our approach opens new avenues of research, both from the point of view of leveraging the power of deep learning in causal discovery and from the point of view of building deep networks with better structure *interpretability*. Once the model is learned, the CGNNs present the advantage to be fully parametrized and may be used to simulate interventions on one or more variables of the model and evaluate their impact on a set of target variables. This usage is relevant in a wide variety of domains, typically among medical and sociological domains.

The main limitation of CGNN is its computational cost, due to the quadratic complexity of the CGNN learning criterion w.r.t. the data size, based on the Maximum Mean Discrepancy between the generated and the observed data. A linear approximation thereof has been proposed, with comparable empirical performances.

The main perspective for further research aims at a better scalability of the approach from medium to large problems. On the one hand, the computational scalability could be tackled by using embedded framework for the structure optimization (inspired by lasso methods). Another perspective regards the extension of the approach to categorical variables.

Appendix

The Maximum Mean Discrepancy (MMD) Statistic

The Maximum Mean Discrepancy (MMD) statistic (Gretton et al. 2007) measures the distance between two probability distributions P and $\hat{P}$, defined over $\mathbb{R}^d$, as the real-valued quantity

$$\text{MMD}_k(P, \hat{P}) = \left\| \mu_k(P) - \mu_k(\hat{P}) \right\|_{\mathscr{H}_k}.$$

Here, $\mu_k = \int k(x, \cdot) \mathrm{d}P(x)$ is the *kernel mean embedding* of the distribution P, according to the real-valued symmetric kernel function $k(x, x') = \langle k(x, \cdot), k(x', \cdot) \rangle_{\mathscr{H}_k}$ with associated reproducing kernel Hilbert space $\mathscr{H}_k$. Therefore, μ_k summarizes P as the expected value of the features computed by k over samples drawn from P.

In practical applications, we do not have access to the distributions P and $\hat{P}$, but to their respective sets of samples $\mathscr{D}$ and $\hat{\mathscr{D}}$, defined in Sect. 4.2.1. In this case, we approximate the kernel mean embedding $\mu_k(P)$ by the *empirical kernel mean embedding* $\mu_k(\mathscr{D}) = \frac{1}{|\mathscr{D}|} \sum_{x \in \mathscr{D}} k(x, \cdot)$, and respectively for $\hat{P}$. Then, the empirical MMD statistic is

$$\begin{aligned} \widehat{\text{MMD}}_k(\mathscr{D}, \hat{\mathscr{D}}) &= \left\| \mu_k(\mathscr{D}) - \mu_k(\hat{\mathscr{D}}) \right\|_{\mathscr{H}_k} \\ &= \frac{1}{n^2} \sum_{i,j}^{n} k(x_i, x_j) + \frac{1}{n^2} \sum_{i,j}^{n} k(\hat{x}_i, \hat{x}_j) - \frac{2}{n^2} \sum_{i,j}^{n} k(x_i, \hat{x}_j). \end{aligned}$$

Importantly, the empirical MMD tends to zero as $n \to \infty$ if and only if $P = \hat{P}$, as long as k is a characteristic kernel (Gretton et al. 2007). This property makes the MMD an excellent choice to model how close the observational distribution P is to the estimated observational distribution $\hat{P}$. Throughout this paper, we will employ a particular characteristic kernel: the Gaussian kernel $k(x, x') = \exp(-\gamma \|x - x'\|_2^2)$, where $\gamma > 0$ is a hyperparameter controlling the smoothness of the features.

In terms of computation, the evaluation of $\text{MMD}_k(\mathscr{D}, \hat{\mathscr{D}})$ takes $O(n^2)$ time, which is prohibitive for large n. When using a shift-invariant kernel, such as the Gaussian kernel, one can invoke Bochner's theorem (Edwards 1964) to obtain a linear-time approximation to the empirical MMD (Lopez-Paz et al. 2015), with form

$$\widehat{\text{MMD}}_k^m(\mathscr{D}, \hat{\mathscr{D}}) = \left\| \hat{\mu}_k(\mathscr{D}) - \hat{\mu}_k(\hat{\mathscr{D}}) \right\|_{\mathbb{R}^m}$$

and $O(mn)$ evaluation time. Here, the *approximate empirical kernel mean embedding* has form

$$\hat{\mu}_k(\mathscr{D}) = \sqrt{\frac{2}{m}} \frac{1}{|\mathscr{D}|} \sum_{x \in \mathscr{D}} [\cos(\langle w_1, x \rangle + b_1), \ldots, \cos(\langle w_m, x \rangle + b_m)],$$

where w_i is drawn from the normalized Fourier transform of k, and $b_i \sim U[0, 2\pi]$, for $i = 1, \ldots, m$. In our experiments, we compare the performance and computation times of both $\widehat{\text{MMD}}_k$ and $\widehat{\text{MMD}}_k^m$.

Proofs

Proposition 1 *Let $X = [X_1, \ldots, X_d]$ denote a set of continuous random variables with joint distribution P, and further assume that the joint density function h of P is continuous and strictly positive on a compact and convex subset of $\mathbb{R}^d$, and zero elsewhere. Letting $\mathscr{G}$ be a DAG such that P can be factorized along $\mathscr{G}$,*

$$P(X) = \prod_i P(X_i | X_{Pa(i;\mathscr{G})})$$

there exists $f = (f_1, \ldots, f_d)$ with f_i a continuous function with compact support in $\mathbb{R}^{|Pa(i;\mathscr{G})|} \times [0, 1]$ such that $P(X)$ equals the generative model defined from FCM $(\mathscr{G}, f, \mathscr{E})$, with $\mathscr{E} = \mathscr{U}[0, 1]$ the uniform distribution on $[0, 1]$.

Proof By induction on the topological order of $\mathscr{G}$. Let X_i be such that $|\text{Pa}(i; \mathscr{G})| = 0$ and consider the cumulative distribution $F_i(x_i)$ defined over the domain of X_i ($F_i(x_i) = Pr(X_i < x_i)$). F_i is strictly monotonous as the joint density function is strictly positive therefore its inverse, the quantile function $Q_i : [0, 1] \mapsto dom(X_i)$ is defined and continuous. By construction, $Q_i(e_i) = F_i^{-1}(e_i)$ and setting $Q_i = f_i$ yields the result.

Assume f_i be defined for all variables X_i with topological order less than m. Let X_j with topological order m and Z the vector of its parent variables. For any noise vector $e = (e_i, i \in \text{Pa}(j; \mathscr{G}))$ let $z = (x_i, i \in \text{Pa}(j; \mathscr{G}))$ be the value vector of variables in Z defined from e. The conditional cumulative distribution $F_j(x_j | Z = z) = Pr(X_j < x_j | Z = z)$ is strictly continuous and monotonous wrt x_j, and can be inverted using the same argument as above. Then we can define $f_j(z, e_j) = F_j^{-1}(z, e_j)$.

Let $K_j = dom(X_j)$ and $K_{\text{Pa}(j;\mathscr{G})} = dom(Z)$. We will show now that the function f_j is continuous on $K_{\text{Pa}(j;\mathscr{G})} \times [0, 1]$, a compact subset of $\mathbb{R}^{|\text{Pa}(j;\mathscr{G})|} \times [0, 1]$.

By assumption, there exist $a_j \in \mathscr{R}$ such that, for $(x_j, z) \in K_j \times K_{\text{Pa}(j;\mathscr{G})}$, $F(x_j|z) = \int_{a_j}^{x_j} \frac{h_j(u,z)}{h_j(z)} \mathrm{d}u$, with h_j a continuous and strictly positive density function. For $(a, b) \in K_j \times K_{\text{Pa}(j;\mathscr{G})}$, as the function $(u, z) \to \frac{h_j(u,z)}{h_j(z)}$ is continuous on the compact $K_j \times K_{\text{Pa}(j;\mathscr{G})}$, $\lim_{x_j \to a} F(x_j|z) = \int_{a_j}^{a} \frac{h_j(u,z)}{h_j(z)} \mathrm{d}u$ uniformly on $K_{\text{Pa}(j;\mathscr{G})}$ and

$\lim_{z \to b} F(x_j|z) = \int_{a_j}^{x_j} \frac{h_j(u,b)}{h_j(b)}$ on K_j, according to exchanging limits theorem, F is continuous on (a, b).

For any sequence $z_n \to z$, we have that $F(x_j|z_n) \to F(x_j|z)$ uniformly in x_j. Let define two sequences u_n and $x_{j,n}$, respectively on [0, 1] and K_j, such that $u_n \to u$ and $x_{j,n} \to x_j$. As $F(x_j|z) = u$ has unique root $x_j = f_j(z, u)$, the root of $F(x_j|z_n) = u_n$, that is, $x_{j,n} = f_j(z_n, u_n)$ converge to x_j. Then the function $(z, u) \to f_j(z, u)$ is continuous on $K_{\mathrm{Pa}(i;\mathscr{G})} \times [0, 1]$.

Proposition 2 *For $m \in [[1, d]]$, let Z_m denote the set of variables with topological order less than m and let d_m be its size. For any d_m-dimensional vector of noise values $e^{(m)}$, let $z_m(e^{(m)})$ (resp. $\widehat{z_m}(e^{(m)})$) be the vector of values computed in topological order from the FCM $(\mathscr{G}, f, \mathscr{E})$ (resp. the CGNN $(\mathscr{G}, \hat{f}, \mathscr{E})$). For any $\epsilon > 0$, there exists a set of networks $\hat{f}$ with architecture $\mathscr{G}$ such that*

$$\forall e^{(m)}, \|z_m(e^{(m)}) - \widehat{z_m}(e^{(m)})\| < \epsilon \tag{14}$$

Proof By induction on the topological order of $\mathscr{G}$. Let X_i be such that $|\mathrm{Pa}(i; \mathscr{G})| = 0$. Following the universal approximation theorem Cybenko (1989), as f_i is a continuous function over a compact of $\mathbb{R}$, there exists a neural net $\hat{f}_i$ such that $\|f_i - \hat{f}_i\|_\infty < \epsilon/d_1$. Thus Eq. (14) holds for the set of networks $\hat{f}_i$ for i ranging over variables with topological order 0.

Let us assume that Proposition 2 holds up to m, and let us assume for brevity that there exists a single variable X_j with topological order $m + 1$. Letting $\hat{f}_j$ be such that $\|f_j - \hat{f}_j\|_\infty < \epsilon/3$ (based on the universal approximation property), letting δ be such that for all u $\|\hat{f}_j(u) - \hat{f}_j(u+\delta)\| < \epsilon/3$ (by absolute continuity) and letting $\hat{f}_i$ satisfying Eq. (14) for i with topological order less than m for $min(\epsilon/3, \delta)/d_m$, it comes: $\|(z_m, f_j(z_m, e_j)) - (\hat{z}_m, \hat{f}_j(\hat{z_m}, e_j))\| \leq \|z_m - \hat{z}_m\| + |f_j(z_m, e_j) - \hat{f}_j(z_m, e_j)| + |\hat{f}_j(z_m, e_j) - \hat{f}_j(\hat{z_m}, e_j)| < \epsilon/3 + \epsilon/3 + \epsilon/3$, which ends the proof.

Proposition 3 *Let $\mathscr{D}$ be an infinite observational sample generated from $(\mathscr{G}, f, \mathscr{E})$. With same notations as in Proposition 2, for every sequence ϵ_t such that $\epsilon_t > 0$ goes to zero when $t \to \infty$, there exists a set $\widehat{f_t} = (\hat{f}_1^t \dots \hat{f}_d^t)$ such that $\widehat{MMD_k}$ between $\mathscr{D}$ and an infinite size sample $\widehat{\mathscr{D}_t}$ generated from the CGNN $(\mathscr{G}, \widehat{f_t}, \mathscr{E})$ is less than ϵ_t.*

Proof According to Proposition 2 and with same notations, letting $\epsilon_t > 0$ go to 0 as t goes to infinity, consider $\hat{f}_t = (\hat{f}_1^t \dots \hat{f}_d^t)$ and $\hat{z}_t$ defined from $\hat{f}_t$ such that for all $e \in [0, 1]^d$, $\|z(e) - \widehat{z_t}(e)\| < \epsilon_t$.

Let $\{\hat{\mathscr{D}}_t\}$ denote the infinite sample generated after $\hat{f}_t$. The score of the CGNN $(\mathscr{G}, \hat{f}_t, \mathscr{E})$ is $\widehat{\mathrm{MMD}}_k(\mathscr{D}, \hat{\mathscr{D}}_t) = \mathbb{E}_{e,e'}[k(z(e), z(e')) - 2k(z(e), \widehat{z_t}(e')) + k(\widehat{z_t}(e), \widehat{z_t}(e'))]$.

As $\hat{f}_t$ converges towards f on the compact $[0, 1]^d$, using the bounded convergence theorem on a compact subset of $\mathbb{R}^d$, $\widehat{z_t}(e) \to z(e)$ uniformly for $t \to \infty$, it follows from the Gaussian kernel function being bounded and continuous that $\widehat{\text{MMD}}_k(\mathscr{D}, \hat{\mathscr{D}}_t) \to 0$, when $t \to \infty$.

Proposition 4 *Let $X = [X_1, \ldots, X_d]$ denote a set of continuous random variables with joint distribution P, generated by a CGNN $\mathscr{C}_{\mathscr{G},f} = (\mathscr{G}, f, \mathscr{E})$ with $\mathscr{G}$, a directed acyclic graph. And let $\mathscr{D}$ be an infinite observational sample generated from this CGNN. We assume that P is Markov and faithful to the graph $\mathscr{G}$, and that every pair of variables (X_i, X_j) that are d-connected in the graph are not independent. We note $\widehat{\mathscr{D}}$ an infinite sample generated by a candidate CGNN, $\mathscr{C}_{\widehat{\mathscr{G}},\hat{f}} = (\widehat{\mathscr{G}}, \hat{f}, \mathscr{E})$. Then,*

(i) If $\widehat{\mathscr{G}} = \mathscr{G}$ and $\hat{f} = f$, then $\widehat{MMD}_k(\mathscr{D}, \widehat{\mathscr{D}}) = 0$.
(ii) For any graph $\widehat{\mathscr{G}}$ characterized by the same adjacencies but not belonging to the Markov equivalence class of $\mathscr{G}$, for all $\hat{f}$, $\widehat{MMD}_k(\mathscr{D}, \widehat{\mathscr{D}}) \neq 0$.

Proof The proof of (i) is obvious, as with $\widehat{\mathscr{G}} = \mathscr{G}$ and $\hat{f} = f$, the joint distribution $\hat{P}$ generated by $\mathscr{C}_{\widehat{\mathscr{G}},\hat{f}} = (\widehat{\mathscr{G}}, \hat{f}, \mathscr{E})$ is equal to P, thus we have $\widehat{\text{MMD}}_k(\mathscr{D}, \widehat{\mathscr{D}}) = 0$.

(ii) Let consider $\widehat{\mathscr{G}}$ a DAG characterized by the same adjacencies but that do not belong to the Markov equivalence class of $\mathscr{G}$. According to Verma and Pearl (1991), as the DAG $\mathscr{G}$ and $\widehat{\mathscr{G}}$ have the same adjacencies but are not Markov equivalent, there are not characterized by the same v-structures.

a) First, we consider that a v-structure $\{X, Y, Z\}$ exists in $\mathscr{G}$, but not in $\widehat{\mathscr{G}}$. As the distribution P is faithful to $\mathscr{G}$ and X and Z are not d-separated by Y in $\mathscr{G}$, we have that $(X \not\perp\!\!\!\perp Z|Y)$ in P. Now we consider the graph $\widehat{\mathscr{G}}$. Let $\hat{f}$ be a set of neural networks. We note $\hat{P}$ the distribution generated by the CGNN $\mathscr{C}_{\widehat{\mathscr{G}},\hat{f}}$. As $\widehat{\mathscr{G}}$ is a directed acyclic graph and the variables E_i are mutually independent, $\hat{P}$ is Markov with respect to $\widehat{\mathscr{G}}$. As $\{X, Y, Z\}$ is not a v-structure in $\widehat{\mathscr{G}}$, X and Z are d-separated by Y. By using the causal Markov assumption, we obtain that $(X \perp\!\!\!\perp Z|Y)$ in $\hat{P}$.
b) Second, we consider that a v-structure $\{X, Y, Z\}$ exists in $\widehat{\mathscr{G}}$, but not in $\mathscr{G}$. As $\{X, Y, Z\}$ is not a v-structure in $\mathscr{G}$, there is an "unblocked path" between the variables X and Z, the variables X and Z are d-connected. By assumption, there do not exist a set D not containing Y such that $(X \perp\!\!\!\perp Z|D)$ in P. In $\widehat{\mathscr{G}}$, as $\{X, Y, Z\}$ is a v-structure, there exists a set D not containing Y that d-separates X and Z. As for all CGNN $\mathscr{C}_{\widehat{\mathscr{G}},\hat{f}}$ generating a distribution $\hat{P}$, $\hat{P}$ is Markov with respect to $\widehat{\mathscr{G}}$, we have that $X \perp\!\!\!\perp Z|D$ in $\hat{P}$.

In the two cases a) and b) considered above, P and $\hat{P}$ do not encode the same conditional independence relations, thus are not equal. We have then $\widehat{\text{MMD}}_k(\mathscr{D}, \mathscr{D}')$ $\neq 0$.

Table of Scores for the Experiments on Cause-Effect Pairs

See Table 3.

Table 3 Cause-effect relations: area under the precision recall curve on five benchmarks for the cause-effect experiments (weighted accuracy in parenthesis for Tüb). Underline values correspond to best scores

Method	Cha	Net	Gauss	Multi	Tüb
Best fit	56.4	77.6	36.3	55.4	58.4 (44.9)
LiNGAM	54.3	43.7	66.5	59.3	39.7 (44.3)
CDS	55.4	89.5	84.3	37.2	59.8 (65.5)
IGCI	54.4	54.7	33.2	80.7	60.7 (62.6)
ANM	66.3	85.1	88.9	35.5	53.7 (59.5)
PNL	73.1	75.5	83.0	49.0	68.1 (66.2)
Jarfo	$\underline{79.5}$	$\underline{92.7}$	85.3	94.6	54.5 (59.5)
GPI	67.4	88.4	$\underline{89.1}$	65.8	66.4 (62.6)
CGNN ($\widehat{\mathrm{MMD}}_k$)	73.6	89.6	82.9	$\underline{96.6}$	$\underline{79.8}$ (74.4)
CGNN ($\widehat{\mathrm{MMD}}_k^m$)	76.5	87.0	88.3	94.2	76.9 (72.7)

Table of Scores for the Experiments on Graphs

See Tables 4, 5 and 6.

Table 4 Average (std. dev.) results for the orientation of 20 artificial graphs given true skeleton (left), artificial graphs given skeleton with 20% error (middle). Underline values correspond to best scores

	Skeleton without error			Skeleton with 20% of error		
	AUPR	SHD	SID	AUPR	SHD	SID
Constraints						
PC-Gauss	0.67 (0.11)	9.0 (3.4)	131 (70)	0.42 (0.06)	21.8 (5.5)	191.3 (73)
PC-HSIC	0.80 (0.08)	6.7 (3.2)	80.1 (38)	0.49 (0.06)	19.8 (5.1)	165.1 (67)
Pairwise						
ANM	0.67 (0.11)	7.5 (3.0)	135.4 (63)	0.52 (0.10)	19.2 (5.5)	171.6 (66)
Jarfo	0.74 (0.10)	8.1 (4.7)	147.1 (94)	0.58 (0.09)	20.0 (6.8)	184.8 (88)
Score-based						
GES	0.48 (0.13)	14.1 (5.8)	186.4 (86)	0.37 (0.08)	20.9 (5.5)	209 (83)
LiNGAM	0.65 (0.10)	9.6 (3.8)	171 (86)	0.53 (0.10)	20.9 (6.8)	196 (83)
CAM	0.69 (0.13)	7.0 (4.3)	122 (76)	0.51 (0.11)	$\underline{15.6}$ (5.7)	175 (80)
CGNN ($\widehat{\mathrm{MMD}}_k^m$)	0.77 (0.09)	7.1 (2.7)	141 (59)	0.54 (0.08)	20 (10)	179 (102)
CGNN ($\widehat{\mathrm{MMD}}_k$)	$\underline{0.89}$[a] (0.09)	$\underline{2.5}$[a] (2.0)	$\underline{50.45}$[a] (45)	$\underline{0.62}$ (0.12)	16.9 (4.5)	$\underline{134.0}$[a] (55)

[a] Denotes statistical significance at $p = 10^{-2}$

Table 5 Average (std. dev.) results for the orientation of 20 and 50 artificial graphs coming from Syntren simulator given true skeleton. Underline values correspond to best scores

	Syntren network 20 nodes			Syntren network 50 nodes		
	AUPR	SHD	SID	AUPR	SHD	SID
Constraints						
PC-Gauss	0.40 (0.16)	16.3 (3.1)	198 (57)	0.22 (0.03)	61.5 (32)	993 (546)
PC-HSIC	0.38 (0.15)	23 (1.7)	175 (16)	–	–	–
Pairwise						
ANM	0.36 (0.17)	10.1 (4.2)	138 (56)	0.35 (0.12)	29.8 (13.5)	677 (313)
Jarfo	0.42 (0.17)	10.5 (2.6)	148 (64)	0.45 (0.13)	26.2 (14)	610 (355)
Score-based						
GES	0.44 (0.17)	9.8 (5.0)	116 (64)	0.52 (0.03)	21 (11)	462 (248)
LiNGAM	0.40 (0.22)	10.1 (4.4)	135 (57)	0.37 (0.28)	33.4 (19)	757 (433)
CAM	0.73 (0.08)	4.0 (2.5)	49 (24)	0.69 (0.05)	14.8 (7)	285 (136)
CGNN ($\widehat{\mathrm{MMD}}_k^m$)	$\underline{0.80}$[a] (0.12)	3.2 (1.6)	45 (25)	$\underline{0.82}$[a] (0.1)	$\underline{10.2}$[a] (5.3)	$\underline{247}$ (134)
CGNN ($\widehat{\mathrm{MMD}}_k$)	0.79 (0.12)	$\underline{3.1}$[a] (2.2)	$\underline{43}$ (26)	0.75 (0.09)	12.2 (5.5)	309 (140)

[a] Denotes statistical significance at $p = 10^{-2}$

Table 6 Average (std. dev.) results for the orientation of the real protein network given true skeleton. Underline values correspond to best scores

	Causal protein network		
	AUPR	SHD	SID
Constraints			
PC-Gauss	0.19 (0.07)	16.4 (1.3)	91.9 (12.3)
PC-HSIC	0.18 (0.01)	17.1 (1.1)	90.8 (2.6)
Pairwise			
ANM	0.34 (0.05)	8.6 (1.3)	85.9 (10.1)
Jarfo	0.33 (0.02)	10.2 (0.8)	92.2 (5.2)
Score-based			
GES	0.26 (0.01)	12.1 (0.3)	92.3 (5.4)
LiNGAM	0.29 (0.03)	10.5 (0.8)	83.1 (4.8)
CAM	0.37 (0.10)	8.5 (2.2)	78.1 (10.3)
CGNN ($\widehat{\mathrm{MMD}}_k^m$)	0.68 (0.07)	5.7 (1.7)	56.6 (10.0)
CGNN ($\widehat{\mathrm{MMD}}_k$)	$\underline{0.74}$[a] (0.09)	$\underline{4.3}$[a] (1.6)	$\underline{46.6}$[a] (12.4)

[a] Denotes statistical significance at $p = 10^{-2}$

References

Bühlmann, P., Peters, J., Ernest, J., et al. (2014). Cam: Causal additive models, high-dimensional order search and penalized regression. *The Annals of Statistics*, 42(6):2526–2556.

Chickering, D. M. (2002). Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3(Nov):507–554.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv*.

Colombo, D. and Maathuis, M. H. (2014). Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, 15(1):3741–3782.

Colombo, D., Maathuis, M. H., Kalisch, M., and Richardson, T. S. (2012). Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, pages 294–321.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314.

Daniusis, P., Janzing, D., Mooij, J., Zscheischler, J., Steudel, B., Zhang, K., and Schölkopf, B. (2012). Inferring deterministic causal relations. *arXiv preprint arXiv:1203.3475*.

Drton, M. and Maathuis, M. H. (2016). Structure learning in graphical modeling. *Annual Review of Statistics and Its Application*, (0).

Edwards, R. (1964). Fourier analysis on groups.

Fonollosa, J. A. (2016). Conditional distribution variability measures for causality detection. *arXiv preprint arXiv:1601.06680*.

Goldberger, A. S. (1984). Reverse regression and salary discrimination. *Journal of Human Resources*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Neural Information Processing Systems (NIPS)*, pages 2672–2680.

Gretton, A., Borgwardt, K. M., Rasch, M., Schölkopf, B., Smola, A. J., et al. (2007). A kernel method for the two-sample-problem. 19:513.

Gretton, A., Herbrich, R., Smola, A., Bousquet, O., and Schölkopf, B. (2005). Kernel methods for measuring independence. *Journal of Machine Learning Research*, 6(Dec):2075–2129.

Guyon, I. (2013). Chalearn cause effect pairs challenge.

Guyon, I. (2014). Chalearn fast causation coefficient challenge.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*.

Hoyer, P. O., Janzing, D., Mooij, J. M., Peters, J., and Schölkopf, B. (2009). Nonlinear causal discovery with additive noise models. In *Neural Information Processing Systems (NIPS)*, pages 689–696.

Kalisch, M., Mächler, M., Colombo, D., Maathuis, M. H., Bühlmann, P., et al. (2012). Causal inference using graphical models with the r package pcalg. *Journal of Statistical Software*, 47(11):1–26.

Kingma, D. P. and Ba, J. (2014). Adam: A Method for Stochastic Optimization. *ArXiv e-prints*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *NIPS*.

Lopez-Paz, D. (2016). *From dependence to causation*. PhD thesis, University of Cambridge.

Lopez-Paz, D., Muandet, K., Schölkopf, B., and Tolstikhin, I. O. (2015). Towards a learning theory of cause-effect inference. In *ICML*, pages 1452–1461.

Lopez-Paz, D. and Oquab, M. (2016). Revisiting classifier two-sample tests. *arXiv preprint arXiv:1610.06545*.

Mendes, P., Sha, W., and Ye, K. (2003). Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics*, 19(suppl_2):ii122–ii129.

Mooij, J. M., Peters, J., Janzing, D., Zscheischler, J., and Schölkopf, B. (2016). Distinguishing cause from effect using observational data: methods and benchmarks. *Journal of Machine Learning Research*, 17(32):1–102.

Nandy, P., Hauser, A., and Maathuis, M. H. (2015). High-dimensional consistency in score-based and hybrid structure learning. *arXiv preprint arXiv:1507.02608*.

Ogarrio, J. M., Spirtes, P., and Ramsey, J. (2016). A hybrid causal search algorithm for latent variable models. In *Conference on Probabilistic Graphical Models*, pages 368–379.

Pearl, J. (2003). Causality: models, reasoning and inference. *Econometric Theory*, 19(675-685):46.

Pearl, J. (2009). *Causality*. Cambridge university press.

Pearl, J. and Verma, T. (1991). *A formal theory of inductive causation*. University of California (Los Angeles). Computer Science Department.

Peters, J. and Bühlmann, P. (2013). Structural intervention distance (sid) for evaluating causal graphs. *arXiv preprint arXiv:1306.1043*.

Peters, J., Janzing, D., and Schölkopf, B. (2017). *Elements of Causal Inference - Foundations and Learning Algorithms*. MIT Press.

Quinn, J. A., Mooij, J. M., Heskes, T., and Biehl, M. (2011). Learning of causal relations. In *ESANN*.

Ramsey, J. D. (2015). Scaling up greedy causal search for continuous variables. *arXiv preprint arXiv:1507.07749*.

Richardson, T. and Spirtes, P. (2002). Ancestral graph markov models. *The Annals of Statistics*, 30(4):962–1030.

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., and Nolan, G. P. (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529.

Scheines, R. (1997). An introduction to causal inference.

Sgouritsa, E., Janzing, D., Hennig, P., and Schölkopf, B. (2015). Inference of cause and effect with unsupervised inverse regression. In *AISTATS*.

Shen-Orr, S. S., Milo, R., Mangan, S., and Alon, U. (2002). Network motifs in the transcriptional regulation network of escherichia coli. *Nature genetics*, 31(1):64.

Shimizu, S., Hoyer, P. O., Hyvärinen, A., and Kerminen, A. (2006). A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(Oct):2003–2030.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*.

Spirtes, P., Glymour, C., and Scheines, R. (1993). Causation, prediction and search. 1993. *Lecture Notes in Statistics*.

Spirtes, P., Glymour, C. N., and Scheines, R. (2000). *Causation, prediction, and search*. MIT press.

Spirtes, P., Meek, C., Richardson, T., and Meek, C. (1999). An algorithm for causal inference in the presence of latent variables and selection bias.

Spirtes, P. and Zhang, K. (2016). Causal discovery and inference: concepts and recent methodological advances. In *Applied informatics*, volume 3, page 3. Springer Berlin Heidelberg.

Statnikov, A., Henaff, M., Lytkin, N. I., and Aliferis, C. F. (2012). New methods for separating causes from effects in genomics data. *BMC genomics*, 13(8):S22.

Stegle, O., Janzing, D., Zhang, K., Mooij, J. M., and Schölkopf, B. (2010). Probabilistic latent variable models for distinguishing between cause and effect. In *Neural Information Processing Systems (NIPS)*, pages 1687–1695.

Tsamardinos, I., Brown, L. E., and Aliferis, C. F. (2006). The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78.

Van den Bulcke, T., Van Leemput, K., Naudts, B., van Remortel, P., Ma, H., Verschoren, A., De Moor, B., and Marchal, K. (2006). Syntren: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC bioinformatics*, 7(1):43.

Verma, T. and Pearl, J. (1991). Equivalence and synthesis of causal models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, UAI '90, pages 255–270, New York, NY, USA. Elsevier Science Inc.

Yamada, M., Jitkrittum, W., Sigal, L., Xing, E. P., and Sugiyama, M. (2014). High-dimensional feature selection by feature-wise kernelized lasso. *Neural computation*, 26(1):185–207.

Zhang, K. and Hyvärinen, A. (2009). On the identifiability of the post-nonlinear causal model. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 647–655. AUAI Press.

Zhang, K., Peters, J., Janzing, D., and Schölkopf, B. (2012). Kernel-based conditional independence test and application in causal discovery. *arXiv preprint arXiv:1202.3775*.

Zhang, K., Wang, Z., Zhang, J., and Schölkopf, B. (2016). On estimation of functional causal models: general results and application to the post-nonlinear causal model. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(2):13.

Learning Interpretable Rules for Multi-Label Classification

Eneldo Loza Mencía, Johannes Fürnkranz, Eyke Hüllermeier, and Michael Rapp

Abstract Multi-label classification (MLC) is a supervised learning problem in which, contrary to standard multiclass classification, an instance can be associated with several class labels simultaneously. In this chapter, we advocate a rule-based approach to multi-label classification. Rule learning algorithms are often employed when one is not only interested in accurate predictions, but also requires an interpretable theory that can be understood, analyzed, and qualitatively evaluated by domain experts. Ideally, by revealing patterns and regularities contained in the data, a rule-based theory yields new insights in the application domain. Recently, several authors have started to investigate how rule-based models can be used for modeling multi-label data. Discussing this task in detail, we highlight some of the problems that make rule learning considerably more challenging for MLC than for conventional classification. While mainly focusing on our own previous work, we also provide a short overview of related work in this area.

Keywords Multi-label classification · Label-dependencies · Rule learning · Separate-and-conquer

1 Introduction

Multi-label classification (MLC) is a problem in the realm of supervised learning. Contrary to conventional, single-label classification, MLC allows an instance to be associated with multiple class labels simultaneously. Dealing with and taking advantage of (statistical) dependencies between the presence and absence (relevance

E. Loza Mencía (✉) · J. Fürnkranz · M. Rapp
Knowledge Engineering Group, Technische Universität Darmstadt, Darmstadt, Germany
e-mail: eneldo@ke.tu-darmstadt.de; juffi@ke.tu-darmstadt.de; mrapp@ke.tu-darmstadt.de

E. Hüllermeier
Intelligent Systems, Universität Paderborn, Paderborn, Germany
e-mail: eyke@upb.de

H. J. Escalante et al. (eds.), *Explainable and Interpretable Models in Computer Vision and Machine Learning*, The Springer Series on Challenges in Machine Learning, https://doi.org/10.1007/978-3-319-98131-4_4

and irrelevance) of different labels has been identified as a key issue in previous work on MLC. To improve predictive performance, essentially all state-of-the-art MLC algorithms therefore seek to capture label dependencies in one way or the other.

In this chapter, we will argue that inductive rule learning is a promising approach for tackling MLC problems. In particular, rules provide an interpretable model for mapping inputs to outputs, and allow for tightly integrating input variables and labels into coherent comprehensible theories. For example, so-called global dependencies between labels can be explicitly modeled and expressed in the form of rules. Moreover, these can be easily generalized to local dependencies, which include regular input features as a local context in which a label dependency holds. Such rules, which mix labels and features, are nevertheless directly interpretable and comprehensible for humans. Even if complex and long rules are generated, the implication between labels can be easily grasped by focusing on the part of the rules that actually considers the labels. Hence, in contrast to many other model types that capture label dependencies implicitly, such dependencies can be analyzed and interpreted more directly.

We will start with a brief definition and formalization of the multi-label learning problem (Sect. 2), in which we also introduce a dataset that will serve as a running example. In Sect. 3, we then define multi-label rules, highlighting the differences to conventional classification rules, discuss various dimensions and choices that have to be made, and list some challenges for learning such rules. Sections 4 and 5 then deal with descriptive and predictive multi-label rule learning, respectively. The former recalls association-rule based approaches and discusses how properties like anti-monotonicity can be used to efficiently search for a suitable head for a given rule body, whereas the latter discusses two approaches for learning predictive rule-based theories: one based on stacking different label prediction layers, and another one based on adapting the separate-and-conquer or covering strategy to the multi-label case. Finally, in Sect. 6, we present and discuss rule-based theories for a few well-known sample multi-label databases, before we conclude in Sect. 7.

2 Multi-Label Classification

Multi-label classification has received a lot of attention in the recent machine learning literature (Tsoumakas et al. 2010, 2012; Gibaja and Ventura 2014, 2015; Varma and Cissé 2015; Herrera et al. 2016; Zhang and Zhou 2014). The motivation for MLC originated in the field of text categorization (Hayes and Weinstein 1991; Lewis 1992, 2004), but nowadays multi-label methods are used in applications as diverse as music categorization (Trohidis et al. 2008), semantic scene classification (Boutell et al. 2004), or protein function classification (Elisseeff and Weston 2001).

2.1 Problem Definition

The task of MLC is to associate an instance with one or several labels λ_i out of a finite label space $\mathscr{L}$. with $n = |\mathscr{L}|$ being the number of available labels. Contrary to ordinary classification, MLC allows each instance to be associated with more than one (class) label, but, in contrast to multiclass learning, alternatives are not assumed to be mutually exclusive, such that multiple labels may be associated with a single instance. Figure 1a shows an example, which relates persons described with some demographic characteristics to the newspapers and magazines they subscribe. Obviously, the number of subscriptions can vary. For example, subject #1 (a single

(a)

	Person				Subscribed Magazines
No.	**Education**	**Marital**	**Sex**	**Children?**	
1	Primary	Single	Male	No	∅
2	Primary	Single	Male	Yes	∅
3	Primary	Married	Male	No	{tabloid}
4	University	Divorced	Female	No	{quality, fashion}
5	University	Married	Female	Yes	{quality, fashion}
6	Secondary	Single	Male	No	{tabloid}
7	University	Single	Male	No	{quality, tabloid}
8	Secondary	Divorced	Female	No	{quality, sports}
9	Secondary	Single	Female	Yes	{tabloid, fashion}
10	Secondary	Married	Male	Yes	{quality, tabloid}
11	Primary	Married	Female	No	∅
12	Secondary	Divorced	Male	Yes	∅
13	University	Divorced	Male	Yes	{quality, tabloid}
14	Secondary	Divorced	Male	No	{quality, sports}

(b)

	Person				Subscribed Magazines			
No.	**Education**	**Marital**	**Sex**	**Children?**	**Quality**	**Tabloid**	**Fashion**	**Sports**
1	Primary	Single	Male	No	0	0	0	0
2	Primary	Single	Male	Yes	0	0	0	0
3	Primary	Married	Male	No	0	1	0	0
4	University	Divorced	Female	No	1	0	1	0
5	University	Married	Female	Yes	1	0	1	0
6	Secondary	Single	Male	No	0	1	0	0
7	University	Single	Male	No	1	1	0	0
8	Secondary	Divorced	Female	No	1	0	0	1
9	Secondary	Single	Female	Yes	0	1	1	0
10	Secondary	Married	Male	Yes	1	1	0	0
11	Primary	Married	Female	No	0	0	0	0
12	Secondary	Divorced	Male	Yes	0	0	0	0
13	University	Divorced	Male	Yes	1	1	0	0
14	Secondary	Divorced	Male	No	1	0	0	1

Fig. 1 Two representations of a sample multi-label classification problem, which relates demographic characteristics to subscribed newspapers and magazines. (**a**) With set-valued outputs. (**b**) With binary output vectors

male with primary education and no kids) has subscribed to no magazines at all, whereas #13 (a divorced male with university degree and children) obtains a quality newspaper and a tabloid.

Potentially, there are 2^n different allowed allocations of the output space, which is a dramatic growth compared to the n possible states in the multiclass setting. However, not all possible combinations need to occur in the database. For example, nobody in this database has subscribed to both a fashion and a sports magazine. Note that these label attributes are not independent. The fact that there may be correlations and dependencies between the labels in $\mathscr{L}$ makes the multi-label setting particularly challenging and interesting compared to the classical setting of binary and multiclass classification.

Formally, MLC refers to the task of learning a predictor $f : \mathscr{X} \rightarrow 2^{\mathscr{L}}$ that maps elements $\mathbf{x}$ of an instance space $\mathscr{X}$ to subsets P of a set of labels $\mathscr{L} = \{\lambda_1, \ldots, \lambda_n\}$. Equivalently, predictions can be expressed as binary vectors $\mathbf{y} = f(\mathbf{x}) = (y_1, \ldots, y_n) \in \{0, 1\}^n$, where each attribute y_i encodes the presence (1) or absence (0) of the corresponding label λ_i, We will use these two notations interchangeably, i.e., y will be used to refer to an element in a binary prediction vector, whereas λ refers to an element in a predicted label set.

An instance $\mathbf{x}_j$ is in turn represented in attribute-value form, i.e., it consists of a vector $\mathbf{x}_j := \langle x_1, \ldots, x_a \rangle \in \mathscr{X} = \phi_1 \times \ldots \times \phi_a$, where ϕ_i is a numeric or nominal attribute.

Consequently, the training data set of an MLC problem can be defined as a sequence of tuples $\mathscr{T} := \langle(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_m, \mathbf{y}_m)\rangle \subseteq \mathscr{X} \times \mathscr{Y}$ with $m = |\mathscr{T}|$. Figure 1 shows both representations, once with sets as outputs (Fig. 1a) and once with binary vectors as outputs (Fig. 1b).

2.2 Dependencies in Multi-Label Classification

The simplest and best known approach to multi-label classification is *binary relevance* (BR) learning (e.g. Tsoumakas et al. 2010). It tackles a multi-label problem by learning one classifier for each label, using all its instances as positive and the others as negative examples. The obvious disadvantage of this transformation is the ignorance of possible dependencies between the labels. More advanced methods seek to exploit such dependencies, mainly with the goal of improving predictive accuracy.

The goal of most classification algorithms is to capture dependencies between input variables x_j and the output variables y_i. In fact, the prediction $\hat{y} = f(\mathbf{x})$ of a scoring classifier f is often regarded as an approximation of the conditional probability $\Pr(y = \hat{y} \mid \mathbf{x})$, i.e., the probability that $\hat{y}$ is the true label for the given instance $\mathbf{x}$. In MLC, dependencies may not only exist between $\mathbf{x}$ and each target y_i, but also between the labels $y_1, \ldots, y_n$ themselves.

A key distinction is between *unconditional* and *conditional independence* of labels. In the first case, the joint distribution $\Pr(\mathbf{y})$ in the label space factorizes into the product of the marginals $\Pr(y_i)$, i.e., $\Pr(\mathbf{y}) = \Pr(y_1)\cdots\Pr(y_n)$, whereas in the latter case, the factorization $\Pr(\mathbf{y} \mid \mathbf{x}) = \Pr(y_1 \mid \mathbf{x})\cdots\Pr(y_n \mid \mathbf{x})$ holds conditioned on $\mathbf{x}$, for every instance $\mathbf{x}$. In other words, unconditional dependence is a kind of global dependence (for example originating from a hierarchical structure on the labels), whereas conditional dependence is a dependence locally restricted to a single point in the instance space.

In the literature, both types of dependence have been explored. For example, Sucar et al. (2014) model label dependence in the form of a Bayesian network. Chekina et al. (2013) provide an empirical analysis of the different types of dependencies between pairs of labels on standard benchmark datasets, and analyze the usefulness of modeling them. Unconditional dependencies were analyzed by a simple χ^2 test on the label co-occurrence matrix, whereas for detecting unconditional dependencies they compared the performance of a classifier f_i for a label y_i trained on the instance features ($\mathbf{x}$) to the same learning algorithm being applied to the input space $(\mathbf{x}, y_j)$ augmented by the label feature of a second label y_j. If the predictions differ statistically significantly, then y_i is assumed to be conditionally dependent on y_j. Their evaluations show that pairwise unconditional dependencies occur more frequently than pairwise conditional dependencies, and that, surprisingly, modeling global dependencies is more beneficial in terms of predictive performance. However, this finding is very specific to their setting, where the dependence information is basically used to guide a decomposition into smaller problems with less labels that are either independent or dependent. In addition, only pairwise co-occurrence and pairwise exclusion can effectively be exploited by their approach. As we will see in Sect. 3.2, rules can be used to flexibly formulate a variety of different dependencies, including partially label-dependent or local dependencies.

2.3 Evaluation of Multi-Label Predictions

2.3.1 Bipartition Evaluation Functions

To evaluate the quality of multi-label predictions, we use bipartition evaluation measures (cf. Tsoumakas et al. 2010) which are based on evaluating differences between true (*ground truth*) and predicted label vectors. They can be considered as functions of two-dimensional *label confusion matrices* which represent the *true positive* (TP), *false positive* (FP), *true negative* (TN) and *false negative* (FN) label predictions. For a given example $\mathbf{x}_j$ and a label y_i the elements of an atomic confusion matrix C_i^j are computed as

$$C_i^j = \begin{pmatrix} TP_i^j & FP_i^j \\ FN_i^j & TN_i^j \end{pmatrix} = \begin{pmatrix} y_i^j \hat{y}_i^j & (1 - y_i^j)\hat{y}_i^j \\ (1 - y_i^j)(1 - \hat{y}_i^j) & y_i^j(1 - \hat{y}_i^j) \end{pmatrix} \qquad (1)$$

where the variables y_i^j and $\hat{y}_i^j$ denote the absence (0) or presence (1) of label λ_i of example $\mathbf{x}_j$ according to the ground truth or the predicted label vector, respectively.

Note that for candidate rule selection we assess TP, FP, TN, and FN differently. To ensure that absent and present labels have the same impact on the performance of a rule, we always count correctly predicted labels as TP and incorrect predictions as FP, respectively. Labels for which no prediction is made are counted as TN if they are absent, or as FN if they are present.

2.3.2 Multi-Label Evaluation Functions

In the following some of the most common bipartition metrics $\delta(C)$ used for MLC are presented (cf., e.g., Tsoumakas et al. 2010). They are mappings $\mathbb{N}^{2x2} \to \mathbb{R}$ that assign a real-valued score (often normalized to [0, 1]) to a confusion matrix C. Predictions that reach a greater score outperform those with smaller values.

- **Precision:** Percentage of correct predictions among all predicted labels.

$$\delta_{prec}(C) := \frac{TP}{TP + FP} \tag{2}$$

- **Hamming accuracy:** Percentage of correctly predicted present and absent labels among all labels.

$$\delta_{hamm}(C) := \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

- **F-measure:** Weighted harmonic mean of precision and recall. If $\beta < 1$, precision has a greater impact. If $\beta > 1$, the F-measure becomes more recall-oriented.

$$\delta_F(C) := \frac{\beta^2 + 1}{\frac{\beta^2}{\delta_{rec}(C)} + \frac{1}{\delta_{prec}(C)}} \text{, with } \delta_{rec}(C) = \frac{TP}{TP + FN} \text{ and } \beta \in [0, \infty] \tag{4}$$

- **Subset accuracy:** Percentage of perfectly predicted label vectors among all examples. Per definition, it is always calculated using example-based averaging.

$$\delta_{acc}(C) := \frac{1}{m} \sum_j \left[\mathbf{y}_j = \hat{\mathbf{y}}_j \right] \text{, with } [x] = \begin{cases} 1, & \text{if } x \text{ is true} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

2.3.3 Aggregation and Averaging

When evaluating multi-label predictions that have been made for m examples with n labels, one has to deal with the question of how to aggregate the resulting $m \cdot n$ atomic confusion matrices. Essentially, there are four possible averaging strategies – either *(label- and example-based) micro-averaging*, *label-based (macro-)averaging*, *example-based (macro-) averaging* or *(label- and example-based) macro-averaging*. Due to the space limitations, we restrict our analysis to the most popular aggregation strategy employed in the literature, namely *micro-averaging*. This particular averaging strategy is formally defined as

$$\delta(C) = \delta\left(\sum_j \sum_i C_i^j\right) \equiv \delta\left(\sum_i \sum_j C_i^j\right), \tag{6}$$

where the $\sum$ operator denotes the cell-wise addition of confusion matrices.

3 Multi-Label Rule Learning

In this section, we discuss rule-based approaches for multi-label classification. We start with a brief recapitulation of inductive rule learning.

3.1 Rule Learning

Rule learning has a very long history and is a well-known problem in the machine learning community (Fürnkranz et al. 2012). Over the years many different algorithms to learn a set of rules were introduced. The main advantage of rule-based classifiers is the interpretability of the models as rules can be more easily comprehended by humans than other models such as neural networks. Also, it is easy to define a syntactic generality relation, which helps to structure the search space. The structure of a rule offers the calculation of overlapping of rules as well as *more specific* and *more general*-relations. Thus, the rule set can be easily modified as opposed to most statistical models such as SVMs or neural networks. However, most rule learning algorithms are currently limited to binary or multiclass classification. Depending on the goal, one may discriminate between predictive and descriptive approaches.

3.1.1 Predictive Rule Learning

Classification rules are commonly expressed in the form

$$\mathbf{r} : H \leftarrow B, \tag{7}$$

where the body B consists of a number of *conditions*, which are typically formed from the attributes of the instance space, and the head H simply assigns a value to the output attribute (e.g., $y = 0$ or $y = 1$ in binary classification). We refer to this type of rules as *single-label head* rules.

For combining such rules into predictive theories, most algorithms follow the *covering* or *separate-and-conquer* strategy (Fürnkranz 1999), i.e., they proceed by learning one rule at a time. After adding a rule to the growing rule set, all examples covered by this rule are removed, and the next rule is learned from the remaining examples. In order to prevent overfitting, the two constraints that all examples have to be covered (*completeness*) and that negative examples must not be covered (*consistency*) can be relaxed so that some positive examples may remain uncovered and/or some negative examples may be covered by the set of rules. Typically, heuristics are used that trade off these two objectives and guide the search towards solutions that excel according to both criteria (Janssen and Fürnkranz 2010; Minnaert et al. 2015).[1] This may also be viewed as a simple instance of the *LeGo* framework for combining local patterns (individual rules) to global theories (rule sets or decision lists) (Knobbe et al. 2008).

3.1.2 Descriptive Rule Learning

Contrary to predictive approaches, descriptive rule learning algorithms typically focus on individual rules. For example, *subgroup discovery* algorithms (Kralj Novak et al. 2009) aim at discovering groups of data that have an unusual class distribution, or *exceptional model mining* (Duivesteijn et al. 2016) generalizes this notion to differences with respect to data models instead of data distributions. Duivesteijn et al. (2012) extended the latter approach to MLC for finding local exceptionalities in the dependence relations between labels.

Arguably the best-known descriptive approach are *association rules* (Goethals 2005; Hipp et al. 2000; Zhang and Zhang 2002), which relate properties of the data in the body of the rule to other properties in the head of the rule. Thus, contrary to classification rules, where the head consists of a single class label, multiple conditions may appear in the head. Typically, association rules are found by exhaustive search, i.e., all rules that satisfy a minimum support and minimum confidence threshold are found (Agrawal et al. 1995; Zaki et al. 1997; Han et al. 2004), and subsequently filtered and/or ordered according to heuristics. Only few algorithms directly find rules that optimize a given score function (Webb 2000). They can also be combined into theories with class association rule learning algorithms such as CBA (Liu et al. 1998, 2000; Sulzmann and Fürnkranz 2008).

3.2 Multi-Label Rules

The goal of multi-label rule learning is to discover rules of the form

$$\mathbf{r} : \hat{\mathbf{y}} \leftarrow B \tag{8}$$

[1] Some algorithms, such as ENDER (Dembczyński et al. 2010), also find rules that directly minimize a regularized loss function.

Table 1 Examples of different forms of multi-label rules based on the sample dataset in Fig. 1

Head		Body	Example rule
Single-label	Positive Negative	Label-independent	*quality* ← University, Female $\overline{\textit{tabloid}}$ ← Secondary, Divorced
Single-label	Positive Negative	Partially label-dependent	*quality* ← $\overline{\textit{tabloid}}$, University $\overline{\textit{quality}}$ ← $\overline{\textit{tabloid}}$, Primary
Single-label	Positive Negative	Fully label-dependent	$\overline{\textit{sports}}$ ← *fashion* $\overline{\textit{sports}}$ ← *quality*, *tabloid*
Multi-label	Partial Complete	Label-independent	*quality*, *fashion* ← University, Female *quality*, $\overline{\textit{tabloid}}$, *fashion*, $\overline{\textit{sports}}$ ← University, Female
Multi-label	Partial	Partially label-dependent Fully label-dependent	*tabloid*, $\overline{\textit{sports}}$ ← *fashion*, Children $\overline{\textit{fashion}}$, $\overline{\textit{sports}}$ ← $\overline{\textit{quality}}$, $\overline{\textit{tabloid}}$

Attribute names in italic denote label attributes, attributes with an overline denote negated conditions

The head of the rule may be viewed as a binary prediction vector $\hat{\mathbf{y}}$, or as a set of predicted labels $\hat{P} \subset \mathscr{L}$. The body may consist of several conditions, which the examples that are covered by the rule have to satisfy. In this work, only conjunctive, propositional rules are considered, i.e., each condition compares an attribute's value to a constant by either using equality (nominal attributes) or inequalities (numerical attributes).

In mixed representation rules, labels may occur both as rule features (in the body of the rule) and as predictions (in the head of the rule). Formally, we intend to learn rules of the form

$$\mathbf{r} : y^{(i+j+k)}, \ldots, y^{(i+j+1)} \leftarrow y^{(i+j)}, \ldots, y^{(i+i)}, \phi^{(i)}, \ldots, \phi^{(1)} \tag{9}$$

in which $i \geq 0$ Boolean features, which characterize the input instances, can be mixed with $j \geq 0$ labels in the body of the rule, and are mapped to $k > 0$ different labels in the head of the rule.

Table 1 shows examples of different types of rules that can be learned from the sample dataset shown in Fig. 1. One can distinguish rules according to several dimensions:

- *multi-label* vs. *single-label*: Does the head of the rule contain only a single or multiple predictions?
- *positive* vs. *negative*: Can we predict only the presence of labels or also their absence?
- *dependent* vs. *independent*: Do the predictions in part or fully depend on other labels?

The predictions in the head of the rule may also have different semantics. We differentiate between *full predictions* and *partial predictions*.

- *full predictions:* Each rule predicts a full label vector $\hat{\mathbf{y}}$, i.e., if a label λ_i is not contained in the head, its absence is predicted, i.e., $y_i = 0$.
- *partial predictions:* Each rule predicts the presence or absence of the label only for a subset of the possible labels. For the remaining labels the rule does not make a prediction (but other rules might).

For denoting absence of labels, we will sometimes also use a bar above the labels, i.e., $\overline{\lambda}$ denotes that label λ is predicted as non-relevant or not observed. We also allow $y = ?$ in heads $\mathbf{y}$ and $P \subset \mathscr{L} \cup \{\overline{\lambda_1}, \ldots, \overline{\lambda_n}\}$ to denote that certain labels are not concerned by a rule, i.e., that the label is neither predicted as present nor as absent.

Alternative categorizations of dependencies are possible. For example, Park and Fürnkranz (2008) categorized full label dependencies into *subset constraints* $\lambda_i \leftarrow \lambda_j$ (the instances labeled with λ_j are a subset of those labeled with λ_i) and *exclusion constraints* $\overline{\lambda_i} \leftarrow \lambda_j$ (the instances labeled with λ_i are disjoint from those labeled with λ_j), which can be readily expressed in a rule-based manner. Fully label dependencies are also known as *global dependencies* whereas partially label-dependent rules are also known as *local* and *semi-local dependencies*. For example, in rule (9), the features used in the body of the rule $\phi^{(1)}, \phi^{(2)}, \ldots, \phi^{(i)}$ form the local context in which the dependency $\lambda^{(i+1)}, \ldots, \lambda^{(i+j)} \to \lambda^{(i+j+1)}, \lambda^{(i+j+2)}, \ldots, \lambda^{(k)}$ holds.

3.3 Challenges for Multi-Label Rule Learning

Proposing algorithms that directly learn sets of such rules is a very challenging problem, which involves several subproblems that are not or only inadequately addressed by existing rule learning algorithms.

Firstly, rule-based models expand the class of dependency models that are commonly used when learning from multi-label data. As already explained, one commonly distinguishes between *conditional* and *unconditional* label dependencies (Dembczyński et al. 2012), where the former is of a *global* nature and holds (unconditionally) in the entire instance space (regardless of any features of the instances), whereas the latter is of a *local* nature and only holds for a specific instance. By modeling semi-local dependencies that hold in a certain part of the instance space, i.e., for subgroups of the population characterized by specific features, rule-based models allow for a smooth interpolation between these two extremes. Such dependencies can be formulated elegantly via rules that mix regular features and labels in the condition part of the rule, as illustrated in Table 1. Besides, rule models offer interesting alternatives for the interpretation of dependencies. While the conventional definition of dependency is based on probabilistic concepts, rule models are typically associated with deterministic dependencies. Yet, single rules may also be equipped with probabilistic semantics (e.g., the condition specified in the head of the rule holds with a certain probability within the region specified in the rule body).

Secondly, a rule-based formulation adds a considerable amount of flexibility to the learning process. Contrary to single-label classification, there is a large variety of loss functions according to which the performance of multi-label learning algorithms can be assessed (see Sect. 2.3). In a rule-based framework, a loss-minimizing head could be found for individual rules, so that the same rule body could be adapted to different target loss functions. Conversely, while conventional rule learning heuristics are targeted towards minimizing classification error aka 0/1-loss, their adaptation to different multi-target loss functions is not straightforward. Moreover, different loss functions may require different heuristics in the underlying rule learner.

Moving from learning single rules, a process which is also known as subgroup discovery, to the learning of rule sets adds another layer of complexity to the rule learning algorithms (Fürnkranz 2005). Even an adaptation of the simple and straightforward covering strategy, which is predominantly used for learning rule sets in inductive rule learning (Fürnkranz 1999), is a non-trivial task. For example, when learning rules with partial label heads, one has to devise strategies for dealing with examples that are partially covered, in the sense that some of their labels are covered by a rule whereas others are not. One also has to deal with possible conflicts that may arise from mixed positive and negative rules. Last but not least, one has to recognize and avoid circular dependency structures, where, e.g., the prediction of label λ_i depends on the knowledge of a different label λ_j, which in turn depends on knowledge of λ_i. Algorithmically, we consider this the most challenging problem.

Finally, rule-based representations are directly interpretable and comprehensible to humans, at least in principle. Hence, one is able to analyze the induced rule models, including dependencies between labels discovered in the data, and may greatly benefit from the insight they provide. This is in contrast to many other types of models, for which the key information is not directly accessible. Interestingly, the possibility to inspect the algorithmic decision-making process and the right for explanation might play a major role in the up-coming European legislation (Bryce Goodman 2016), which might even raise liability issues for manufacturers, owners and users of artificial intelligence systems.[2]

We note in passing, however, that while rules are commonly perceived to be more comprehensible than other types of hypothesis spaces that are commonly used in machine learning, the topic of learning *interpretable* rules is still not very well explored (Freitas 2013). For example, in many studies, the comprehensibility of learned rules is assumed to be negatively correlated with their complexity, a point of view that has been questioned in recent work (Allahyari and Lavesson 2011; Stecher et al. 2016; Fürnkranz et al. 2018). In this chapter, our main focus is on arguing that the fact that input data and labels can be used to formulate explicit mixed-dependency rules has a strong potential for increasing the interpretability of multi-label learning.

[2] *General Data Protection Regulation* 2016/679 and *Civil law rules on robotics* under the ID of 2015/2103(INL).

4 Discovery of Multi-Label Rules

In this section, we review work on the problem of discovering individual multi-label rules. In Sect. 4.1, we discuss algorithms that are based on association rule discovery, which allow to quickly find mixed dependency rules. However, for these algorithms it often remains unclear what loss is minimized by their predictions. Other approaches, which we will discuss in Sect. 4.2, aim at discovering loss-minimizing rules.

4.1 Association Rule-Based Algorithms

A simple, heuristic way of discovering multi-label rules is to convert the problem into an association rule discovery problem (cf. Sect. 3.1.2). To this end, one can use the union of labels and features as the basic itemset, discover all frequent itemsets, and derive association rules from these frequent itemsets, as most association rule discovery algorithms do. The only modification is that only rules with labels in the head are allowed, whereas potential rules with features in the head will be disregarded.

For instance, Thabtah et al. (2004) and similarly Li et al. (2008) induce single-label association rules, based on algorithms for class association rule discovery (Liu et al. 1998, 2000). Their idea is to use a multiclass, multi-label associative classification approach where single-label class association rules are merged to create multi-label rules. Allamanis et al. (2013) employ a more complex approach based on an genetic search algorithm which integrates the discovery of multi-label heads into the evolutionary process. Similarly, Arunadevi and Rajamani (2011) and Ávila et al. (2010), use evolutionary algorithms or classifier systems for evolving multi-label classification rules thereby avoiding the problem of devising search algorithms that are targeted towards that problem.

An associate multi-label rule learner with several possible labels in the head of the rules was developed by Thabtah et al. (2006). These labels are found in the whole training set, while the multi-label lazy associative approach of Veloso et al. (2007) generates the rules from the neighborhood of a test instance during prediction. The advantage then is that fewer training instances are used to compute the coverage statistics which is beneficial when small rules are a problem as they are often predicted wrong due to whole training set statistics.

A few algorithms focus on discovering global dependencies, i.e., fully label-dependent rules. Park and Fürnkranz (2008) use an association rule miner (Apriori) to discover pairwise subset (implication) and exclusion constraints, which may be viewed as global dependencies. These are then applied in the classification process to correct predicted label rankings that violate the globally found constraints. Similarly, Charte et al. (2014) infer global dependencies between the labels in the form of association rules, and use them as a post-processor for refining the predictions of conventional multi-label learning systems. Papagiannopoulou et al. (2015) propose a method for discovering deterministic positive entailment (implication) and exclusion relationships between labels and sets of labels.

However, while all these approaches allow to quickly discover multi-label rules, it remains mostly unclear what multi-label loss the discovered rules are actually minimizing.

4.2 Choosing Loss-Minimizing Rule Heads

A key advantage of rule-based methods is that learned rules can be flexibly adapted to different loss functions by choosing an appropriate head for a given rule body. Partial prediction rules, which do not predict the entire label vector, require particular attention. Very much in the same way as completeness in terms of covered examples is only important for complete rule-based theories and not so much for individual rules, completeness in terms of predicted labels is less of an issue when learning individual rules. Instead of evaluating the rule candidates with respect to only one target label, multi-label rule learning algorithms need to evaluate candidates w.r.t. all possible target variables and choose the best possible head for each candidate.

Algorithmically, the key problem is therefore to find the empirical loss minimizer of a rule, i.e., the prediction that minimizes the loss on the covered examples, i.e., we need to find the multi-label head **y** which reaches the best possible performance

$$h_{max} = \max_{\mathbf{y}} \; h(\mathbf{r}) = \max_{\mathbf{y}} \; h(\mathbf{y} \leftarrow B) \tag{10}$$

given an evaluation function $h(.)$ and a body B. As recently shown for the case of the F-measure, this problem is highly non-trivial for certain loss functions (Waegeman et al. 2014). Bosc et al. (2016) adapt an algorithm for subgroup discovery so that it can find the top-k multi-label rules, but the quality measure they use is based on subgroup discovery and not related to commonly used multi-label classification losses.

To illustrate the difference between measures used in association rule discovery and in multi-label rule learning, assume that the rule $\lambda_1, \lambda_2 \leftarrow B$ covers three examples $(\mathbf{x}_1, \{\lambda_2\})$, $(\mathbf{x}_2, \{\lambda_1, \lambda_2\})$ and $(\mathbf{x}_3, \{\lambda_1\})$. In conventional association rule discovery the head is considered to be satisfied for one of the three covered examples $(\mathbf{x}_2)$, yielding a precision/confidence value of $\frac{1}{3}$. This essentially corresponds to subset accuracy. On the other hand, micro-averaged precision would correspond to the fraction of 4 correctly predicted labels among 6 predictions, yielding a value of $\frac{2}{3}$.

4.2.1 Anti-Monotonicity and Decomposability

Rapp et al. (2018) investigate the behavior of multi-label loss functions w.r.t. two such properties, namely anti-monotonicity and decomposability. The first property which can be exploited for pruning searches—while still being able to find the best solution—is *anti-monotonicity*. This property is already well known from association rule learning (Agrawal et al. 1995; Goethals 2005; Hipp et al. 2000) and subgroup discovery (Kralj Novak et al. 2009; Atzmüller 2015). In the multi-label context it basically states that, if we use an anti-monotonic heuristic h for evaluating rules, using adding additional labels to a head cannot improve its value if adding the previous label already decreased the heuristic value. An even stronger criterion for pruning the searches can be found particularly for decomposable multi-label evaluation measures. In few words, *decomposability* allows to find the best head by combining the single-label heads which reach the equal maximum heuristic value for a given body and set of examples. Hence, finding the best head for a decomposable heuristic comes at linear costs, as the best possible head can be deduced from considering each available label separately.

Decomposability is a stronger criterion, i.e., an evaluation measure that is decomposable is also anti-monotonic. Decomposable multi-label evaluation measures include micro-averaged rule-dependent precision, F-measure, and Hamming accuracy. Subset accuracy only fulfills the anti-monotonicity property. This can also be seen from Table 2, which shows for a large variety of evaluation measures if maximizing them for a given body can benefit from both criteria. Detailed proofs are provided by Rapp et al. (2018) and Rapp (2016).

4.2.2 Efficient Generation of Multi-Label Heads

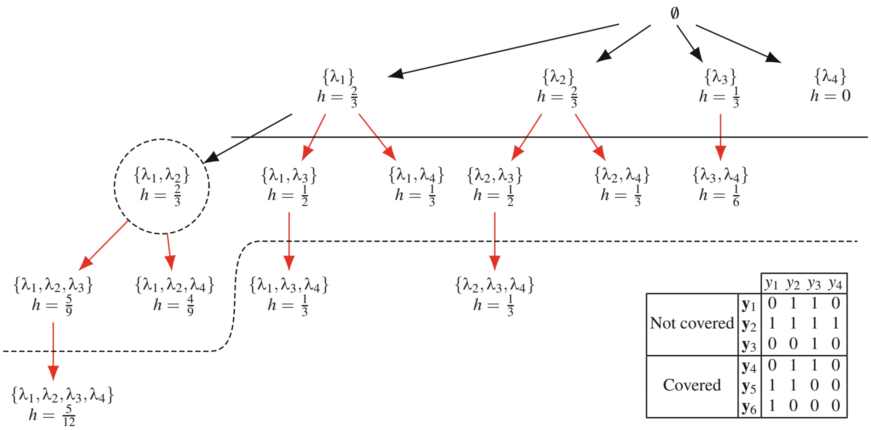
To find the best head for a given body different label combinations must be evaluated by calculating a score based on the used averaging and evaluation strategy. The algorithm described in the following performs a breadth-first search by recursively adding additional label attributes to the (initially empty) head and keeps track of the best rated head. Instead of performing an exhaustive search, the search space is pruned according to the findings in Sect. 4.1. When pruning according to anti-monotonicity unnecessary evaluations of label combinations are omitted in two ways: On the one hand, if adding a label attribute causes the performance to decrease, the recursion is not continued at deeper levels of the currently searched subtree. On the other hand, the algorithm keeps track of already evaluated or pruned heads and prevents these heads from being evaluated in later iterations. When a decomposable evaluation metric is used no deep searches through the label space must be performed. Instead, all possible single-label heads are evaluated in order to identify those that reach the highest score and merge them into one multi-label head rule.

Figure 2 illustrates how the algorithm prunes a search through the label space using anti-monotonicity and decomposability. The nodes of the given search tree

Table 2 Anti-monotonicity and decomposability of selected evaluation functions with respect to different averaging and evaluation strategies

Evaluation function	Evaluation strategy	Averaging strategy	Anti-monotonicity	Decomposability
Precision	Partial predictions	Micro-averaging	Yes	Yes
		Label-based	Yes	Yes
		Example-based	Yes	Yes
		Macro-averaging	Yes	Yes
	Full predictions	Micro-averaging	Yes	–
		Label-based	Yes	–
		Example-based	Yes	–
		Macro-averaging	Yes	–
Recall	Partial predictions	Micro-averaging	Yes	Yes
		Label-based	Yes	Yes
		Example-based	–	–
		Macro-averaging	Yes	Yes
	Full predictions	Micro-averaging	Yes	–
		Label-based	Yes	–
		Example-based	–	–
		Macro-averaging	Yes	–
Hamming accuracy	Partial predictions	Micro-averaging	Yes	Yes
		Label-based	Yes	Yes
		Example-based	Yes	Yes
		Macro-averaging	Yes	Yes
	Full predictions	Micro-averaging	Yes	–
		Label-based	Yes	–
		Example-based	Yes	–
		Macro-averaging	Yes	–
F-measure	Partial predictions	Micro-averaging	Yes	Yes
		Label-based	Yes	Yes
		Example-based	Yes	Yes
		Macro-averaging	Yes	Yes
	Full predictions	Micro-averaging	Yes	–
		Label-based	Yes	–
		Example-based	Yes	–
		Macro-averaging	Yes	–
Subset accuracy	Partial predictions	Example-based	Yes	–
	Full predictions		–	–

correspond to the evaluations of label combinations, resulting in heuristic values h. The edges correspond to adding an additional label to the head which is represented by the preceding node. As equivalent heads must not be evaluated multiple times, the tree is unbalanced.


$\emptyset$
$\{\lambda_1\}$ $h = \frac{2}{3}$
$\{\lambda_2\}$ $h = \frac{2}{3}$
$\{\lambda_3\}$ $h = \frac{1}{3}$
$\{\lambda_4\}$ $h = 0$
$\{\lambda_1, \lambda_2\}$ $h = \frac{2}{3}$
$\{\lambda_1, \lambda_3\}$ $h = \frac{1}{2}$
$\{\lambda_1, \lambda_4\}$ $h = \frac{1}{3}$
$\{\lambda_2, \lambda_3\}$ $h = \frac{1}{2}$
$\{\lambda_2, \lambda_4\}$ $h = \frac{1}{3}$
$\{\lambda_3, \lambda_4\}$ $h = \frac{1}{6}$
$\{\lambda_1, \lambda_2, \lambda_3\}$ $h = \frac{5}{9}$
$\{\lambda_1, \lambda_2, \lambda_4\}$ $h = \frac{4}{9}$
$\{\lambda_1, \lambda_3, \lambda_4\}$ $h = \frac{1}{3}$
$\{\lambda_2, \lambda_3, \lambda_4\}$ $h = \frac{1}{3}$
$\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$ $h = \frac{5}{12}$

		y_1	y_2	y_3	y_4
Not covered	$\mathbf{y}_1$	0	1	1	0
	$\mathbf{y}_2$	1	1	1	1
	$\mathbf{y}_3$	0	0	1	0
Covered	$\mathbf{y}_4$	0	1	1	0
	$\mathbf{y}_5$	1	1	0	0
	$\mathbf{y}_6$	1	0	0	0

Fig. 2 Search through the label space $2^{\mathscr{L}}$ with $\mathscr{L} = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$ using micro-averaged precision of partial predictions. The examples corresponding to label sets $\mathbf{y}_4, \mathbf{y}_5, \mathbf{y}_6$ are assumed to be covered, whereas those of $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3$ are not. The dashed line (- - -) indicates label combinations that can be pruned with anti-monotonicity, the solid line (—) corresponds to decomposability

5 Learning Predictive Rule-Based Multi-Label Models

Predictive, rule-based theories are formed by combining individual rules into a theory. Such an aggregation step is necessary because each individual rule will only cover a part of the example space. When mixed dependency rules, i.e., rules with both labels and features in the rule bodies, are combined into a predictive theory, several problems arise that make the problem considerably harder than the aggregation of local rules into a global rule-based model (Fürnkranz 2005).

As a very simple example, consider the case when two labels λ_i and λ_j always co-occur in the training data. The algorithms discussed in the previous section would then find the inclusion constraints $\lambda_i \rightarrow \lambda_j$ and $\lambda_j \rightarrow \lambda_i$. These are valid and interesting insights into the domain, but in a predictive setting, they will not help to identify both labels as positive unless at least one of the two can be predicted by another rule.[3]

As shown by this example, the problem of circular reasoning is a major concern in the inference with mixed dependency rules. There are two principal ways for tackling this problem. The simplest strategy is to avoid circular dependencies from the very beginning. This means that rules discovered in the learning process have to be organized in a structure that prevents cycles or, alternatively, that additional rules have to be learned with certain constraints on the set of valid conditions in the rule body.

[3] Similar problems have been encountered in inductive logic programming, where the learning of recursive and multi-predicate programs has received some attention (Malerba 2003; De Raedt et al. 1993; Cameron-Jones and Quinlan 1993).

Another way of tackling this problem is to allow for circular dependencies and generalize the inference strategy in a suitable manner. This approach has not yet received much attention in the literature. One notable exception is the work of Montañés et al. (2014) who realized this idea in so-called dependent binary relevance (DBR) learning, which is based on techniques similar to those used in conditional dependency networks (Guo and Gu 2011).

In this section, we will describe two different approaches for tackling the first problem. One, which we call *layered learning* tries to avoid label cycles by requiring an initial guess for labels regardless of any label dependence (Sect. 5.1). While this approach is rather coarse in that batches of rules are learned, we will then also consider approaches that try to adapt the covering or separate-and-conquer strategy, which is frequently used in inductive rule learning (Sect. 5.2).

5.1 Layered Multi-Label Learning

The recently very popular classifier chains (CC; Read et al. 2011) were found to be an effective approach for avoiding label cycles. Their key idea is to use an arbitrary order $\lambda_1, \lambda_2, \dots, \lambda_n$ on the labels, and learn rules that involve predictions for λ_i only from the input attributes and all labels $\lambda_j, j < i$. This, however, has some obvious disadvantages, which have been addressed by many variants that have been investigated in the literature.

One drawback is the (typically randomly chosen) predetermined, fixed order of the classifiers (and hence the labels) in the chain, which makes it impossible to learn dependencies in the opposite direction. This was already recognized by Malerba et al. (1997), who built up a very similar system in order to learn multiple dependent concepts. In this case, the chain on the labels was determined beforehand by a statistical analysis of the label dependencies. Still, using a rule learner for solving the resulting binary problems would only allow to induce rules between two labels in one direction.

5.1.1 Stacked Binary Relevance

An alternative approach without this limitation is to use two levels of classifiers: the first one tries to predict labels independently of each other, whereas the second level of classifiers makes additional use of the predictions of the previous level. More specifically, the training instances for the second level are expanded by the label information of the other labels, i.e., a training example $\mathbf{x}$ for label y_i is transformed into $(x_1, \dots, y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_n)$. During training, the prediction of the first level of classifiers is used as additional features for the second level, i.e., the final prediction $\hat{y}_j$ depends on predictions $f_i(\mathbf{x})$ and $f'_j(\mathbf{x}, f_1(\mathbf{x}), \dots, f_n(\mathbf{x}))$. Hence, each label will be characterized by two sets of rule models, namely the rules $\mathscr{R}_i$ which depend only on instance features, and a second set of rule models

$\textit{quality} \leftarrow \text{University}$
$\overline{\textit{quality}} \leftarrow \text{Single}$
$\textit{quality} \leftarrow \text{Female}$
$\textit{quality} \leftarrow \text{Secondary}$
$\overline{\textit{quality}} \leftarrow \text{true}$
—
$\textit{quality} \leftarrow \text{University}$
$\textit{quality} \leftarrow \textit{sports}, \text{Secondary}$
$\overline{\textit{quality}} \leftarrow \text{Male}$
$\overline{\textit{quality}} \leftarrow \text{true}$

$\overline{\textit{tabloid}} \leftarrow \text{Primary, Single}$
$\overline{\textit{tabloid}} \leftarrow \text{Female, Married}$
$\overline{\textit{tabloid}} \leftarrow \text{Divorced, Secondary}, \overline{\text{Children}}$
$\textit{tabloid} \leftarrow \text{true}$
—
$\overline{\textit{tabloid}} \leftarrow \text{Primary, Single}$
$\textit{tabloid} \leftarrow \overline{\textit{quality}} \leftarrow \text{true}$
$\overline{\textit{tabloid}} \leftarrow \text{Female, Married}$
$\overline{\textit{tabloid}} \leftarrow \text{Secondary, Divorced}$
$\textit{tabloid} \leftarrow \text{true}$

$\overline{\textit{fashion}} \leftarrow \text{Male}$
$\overline{\textit{fashion}} \leftarrow \overline{\text{Children}}$
$\textit{fashion} \leftarrow \text{true}$
—
$\overline{\textit{fashion}} \leftarrow \text{Male}$
$\overline{\textit{fashion}} \leftarrow \overline{\text{Children}}, \overline{\textit{tabloid}} \leftarrow$
$\textit{fashion} \leftarrow \text{true}$

$\textit{sports} \leftarrow \text{Divorced, Secondary}, \overline{\text{Children}}$
$\textit{sports} \leftarrow \text{Children, Male}$
$\overline{\textit{sports}} \leftarrow \text{true}$
—
$\overline{\textit{sports}} \leftarrow \overline{\textit{quality}}, \textit{tabloid}$
$\textit{sports} \leftarrow \text{Secondary}$
$\textit{sports} \leftarrow \text{Children , Male}$
$\overline{\textit{sports}} \leftarrow \text{true}$

Fig. 3 Rule set obtained from layered learning on the example dataset (Fig. 2). Decision lists from first and second level are separated by —

$\mathscr{R}_i^*$ depending (possibly) also on other labels. $\mathscr{R}_i$ can then provide the predictions that are necessary for executing the rules in $\mathscr{R}_i^*$. Loza Mencía and Janssen (2014, 2016) refer to this technique as *stacked binary relevance* (SBR) in contrast to plain, unstacked binary relevance learning.

Figure 3 shows a rule set that can be obtained with SBR for the sample dataset of Fig. 2. One can see that two separate decision lists are learned for each label using a conventional rule learner such as *Ripper* (Cohen 1995). The top list $\mathscr{R}_i$ is learned only from the input features, and the bottom part $\mathscr{R}_i^*$ is learned from input features and the predictions originating from the top rule set.

Despite being able to learn, in contrast to CC, relationships in either direction and in any constellation, this method still has its shortcomings. Firstly, it requires comprehensive predictive rules for each label on the first level even though the labels may effectively be predicted based on other labels. For example, assume the global relation $\lambda_i \leftarrow \lambda_j$, the approach would need to learn a rule model $\mathscr{R}_j$ once for predicting λ_j and once implicitly as part of $\mathscr{R}_i$.

Secondly, a limitation of the stacking approach may appear when circular dependencies exist between labels. A very simple example is if two labels exclude each other, i.e., if both relationships $\lambda_i \leftarrow \overline{\lambda_j}$, and $\lambda_j \leftarrow \overline{\lambda_i}$ hold. Such rules could lead to contradictions and inconsistent states. For instance, assume that both labels λ_i and λ_j were predicted as relevant at the first layer. Both predictions would be flipped at the next layer, leading again to an (irresolvable) inconsistency according to the learned rules.

Another problem that needs to be addressed by layered algorithms is the problem of *error propagation*: If the prediction for a label λ_j depends on another label λ_i, a mistake on the latter is likely to imply a mistake on the former (Senge et al. 2012).

Finally and most notably, the method of (Loza Mencía and Janssen 2016) is limited to produce single-label head rules.

Several variants were proposed in the literature, which in deviation from the basic technique may use predictions instead of the true label information as input (Montañés et al. 2014), only the predictions in a pure stacking manner (Godbole and Sarawagi 2004), or Gibbs sampling instead of the first level of classifiers (Guo and Gu 2011). *Dependent binary relevance* (Montañés et al. 2014) and *conditional dependency networks* (Guo and Gu 2011) are particularly concerned with estimating probability distributions (especially joint distribution). They both use logistic regression as their base classifier, which is particularly adequate for estimating probabilities. This type of models are obviously much harder to comprehend than rules, especially for higher number of input features. Therefore, the label dependencies would remain hidden somewhere in the model, even though they may have been taken into account and accurate classifiers may have been obtained. A more general approach is to integrate the stacking of label features directly into the covering loop. Adaptations of the separate-and-conquer strategy to the multi-label case will be discussed in the next section.

5.2 Multi-Label Separate-and-Conquer

The most frequently used strategy for learning a rule-based predictive theory is the so-called *covering* or *separate-and-conquer* strategy, which is either integrated into the rule learning algorithm (Fürnkranz 1999) or used as a post-processor for selecting a suitable subset among previously learned rules (Liu et al. 1998). Although it may seem quite straightforward, its adaptation to the multi-label case is only trivial if complete rules are learned, i.e., if each rule predicts a complete assignment of the label vector. In this case, one may learn a decision list with the covering strategy, which removes all examples that are covered by previous rules before subsequent rules are learned. In this way, the learning strategy essentially mirrors the sequential nature in which predictions are made with a decision list. In the context of multi-label classification, this strategy corresponds to applying the well known *label powerset* transformation which converts each label combination in the data into a meta-label and then solves the resulting multiclass problem (cf. Tsoumakas et al. 2010).

However, in the case of partial-prediction rules, the situation becomes considerably more complex. One can, e.g., not simply remove all examples that are covered by a rule, because the rule will in general only predict a subset of the relevant labels. An alternative strategy might be to remove all predicted *labels* from the examples that are covered by the rule: if a rule $\mathbf{r} : \hat{\mathbf{y}} \leftarrow B$ covers a training example $(\mathbf{x}, \mathbf{y})$, the example is not removed but replaced with the example $(\mathbf{x}, \mathbf{y} \setminus \hat{\mathbf{y}})$. In this way, each training example remains in the training set until all of its labels are covered by at least one rule. However, even this strategy may be problematic, because removing labels from covered training instances in this way may distort the label dependencies in the training data.

By using separate-and-conquer strategy to induce a rule model, two of the shortcomings of the layered approach from the previous section are addressed. Firstly, the iterative, non-parallel induction of rules in the covering process ensures that redundant rules are avoided because of the separation step. Secondly, cyclic dependencies cannot longer harm the induction or prediction process since the order in which labels are covered or predicted is naturally chosen by the covering process. Similarly, the learner may also dynamically model local label dependencies and does not depend on a global order as in classifier chains.

5.2.1 A Multi-Label Covering Algorithm

Figure 4 shows the multi-label covering algorithm proposed by Loza Mencía and Janssen (2016). The algorithm essentially proceeds as sketched described above, i.e., covered examples are not removed entirely but only the subset of predicted labels is deleted from the example.

For learning a new multi-label rule (line 3), the algorithm performs a top-down greedy search, starting with the most general rule. By adding conditions to the rule's body it can successively be specialized, resulting in fewer examples being covered. Potential conditions result from the values of nominal attributes or from averaging two adjacent values of the sorted examples in case of numerical attributes. Whenever a new condition is added, a corresponding single- or multi-label head that predicts the labels of the covered examples as accurate as possible must be found (cf. Sect. 4.2 and, in particular, Fig. 2).

If a new rule is found, the predicted labels from the examples are marked as covered by these rules, i.e., $(\mathbf{y}_i)_j$ are set to 0 or 1, respectively. As depicted in lines

```
Require: New training example pairs $\mathcal{T} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_m, \mathbf{y}_m)\}$
 1: $\mathcal{T} = \{(\mathbf{x}_1, \hat{\mathbf{y}}_1), \dots, (\mathbf{x}_m, \hat{\mathbf{y}}_m)\}$ with $\hat{\mathbf{y}}_i = (?, ?, \dots, ?)$, $i = 1 \triangleright\triangleright m$
 2: while $\mathcal{T}$ not empty do
 3:     $\mathbf{r} \leftarrow$ findBestGlobalRule($\mathcal{T}$)  ▷ find best rule by refining rule body (and head) w.r.t. some
    heuristic $h$
 4:     apply $\mathbf{r}$: apply header on covered $\mathbf{x}_i \in \mathcal{T}$ and put them into $\mathcal{T}_{cov}$
 5:     if enough $\mathbf{x}_i$ in $\mathcal{T}_{cov}$ with fully covered labels, i.e., $\forall j \,.\, (\hat{\mathbf{y}}_i)_j \neq ?$, then
 6:         make $\mathbf{r}$ full prediction rule and do not add $\mathcal{T}_{cov}$ to $\mathcal{T}$
 7:     else
 8:         re-add $\mathcal{T}_{cov}$ to $\mathcal{T}$
 9:     add $\mathbf{r}$ to decision list $\mathcal{R}$
10: return decision list $\mathcal{R}$
```

Fig. 4 General training algorithm for the multi-label separate-and-conquer algorithm

5–6 in the pseudo-code of Fig. 4, only examples for which enough labels have been predicted can be entirely removed from the training set. A rule that predicts many of such examples is marked as full prediction rule, which means that the execution of the decision list may stop after this rule has fired.

To classify test examples, the learned rules are applied in the order of their induction. If a rule fires, the labels in its head are applied unless they were already set by a previous rule.[4] The process continues with the next rule in the multi-label decision list until either a specially marked full prediction rule is encountered or all rules of the decision list have been processed.

Note that if we had only a single binary (or multiclass) label, i.e. $n = 1$, the described algorithm would behave exactly as the original separate-and-conquer approach. However, for $n > 1$ the algorithm re-adds partially and even fully covered examples instead of removing them (line 8). These examples may serve as an anchor point for subsequent rules and facilitate in such a manner the rule induction process. Moreover, this step enables the algorithm to induce rules which test for the presence of labels. These type of rules are of particular interest since they explicitly reveal label dependencies discovered in the dataset.

Figure 5 shows the results of applying these algorithms to our sample dataset. The top part shows the rules obtained with the single-label head version of Loza Mencía and Janssen (2016), whereas the lower part shows those of the multi-label head extension by Rapp et al. (2018).

6 Case Studies

In this section, we show a few sample result obtained with some of the algorithms described in the previous sections on commonly used benchmark data. Our main focus lies on the inspection and the analysis of the induced rule models, and not so much on their predictive accuracy in comparison to state-of-the-art multi-label classification methods (generally, the predictive performance of rule-based models will be lower). We primarily show some sample rule models, but also discuss statistics on the revealed dependencies.

We experimented with several datasets from the MULAN repository.[5] Table 3 gives a brief overview of the used datasets, along with characteristics such as the number of instances, the number and nature of the attributes, as well as some characteristics on the distribution of labels. The datasets are from different domains and have varying properties. Details of the data are given in the analysis when needed.

[4] This corresponds to the default strategy in classification rule learning, where rules are appended at the end of a list. Note however, that there are also good arguments for prepending rules at the beginning of the list, so that, e.g., exceptions are handled before the general rule (Webb 1994).

[5] http://mulan.sf.net/datasets.html

(a)

$\overline{fashion} \leftarrow \text{Male}$

$\overline{sports} \leftarrow \text{Children}$

$\overline{quality} \leftarrow \text{Primary}$

$quality \leftarrow \text{University}$

$\overline{sports} \leftarrow \overline{quality} \leftarrow \text{true}$

$\overline{tabloid} \leftarrow \text{Female}, \overline{\text{Children}}$

$fashion \leftarrow \text{Children}$

$\overline{sports} \leftarrow \text{University}$

$\overline{quality} \leftarrow \text{Single}$

$quality \leftarrow \overline{\text{Children}}$

$sports \leftarrow \text{Divorced}$

$tabloid \leftarrow \text{Married, Male}$

$\overline{tabloid} \leftarrow \text{Primary}$

$tabloid \leftarrow \text{Single}$

$\overline{tabloid} \leftarrow \text{Secondary}$

$\overline{sports} \leftarrow \text{true}$

$tabloid \leftarrow \text{Divorced}$

$\overline{tabloid} \leftarrow \text{true}$

$\overline{fashion} \leftarrow \text{Married}$

$\overline{fashion}, * \leftarrow sports$

$fashion \leftarrow \text{true}$

$quality, * \leftarrow tabloid$

$\overline{quality}, * \leftarrow \text{true}$

(b)

$\overline{quality}, \overline{fashion}, \overline{sports} \leftarrow \text{Primary}$

$quality, \overline{sports} \leftarrow \text{University}$

$\overline{fashion} \leftarrow \text{Male}$

$\overline{quality}, tabloid, fashion, \overline{sports} \leftarrow \text{Single, Secondary}$

$quality, \overline{tabloid}, sports \leftarrow \text{Female}$

$quality, tabloid, fashion, \overline{sports} \leftarrow \text{Married}$

$quality, \overline{tabloid}, \overline{fashion}, sports \leftarrow \text{Secondary}, \overline{\text{Children}}$

$\overline{quality}, fashion, \overline{sports} \leftarrow \text{true}$

$\overline{tabloid}, * \leftarrow \overline{quality}$

$tabloid, * \leftarrow \text{true}$

Fig. 5 Decision lists induced from the sample dataset of Fig. 2 with precision as heuristic. The stars (*) indicate full prediction rules, after which the prediction stops if the rule fires. (**a**) Single-label head rules (read column-wise). (**b**) Multi-label head rules

6.1 Case Study 1: Single-Label Head Rules

In the first case study, we compared several single-head multi-label rule learning algorithms, namely conventional binary relevance (*BR*), the layered algorithm stacked binary relevance (*SBR*), and a separate-and-conquer learner seeking for rules with only a single label in the head (*Single*). The rule learner *Ripper* (Cohen 1995) was used for finding the label-specific candidate single-head candidate rules. Among these, the best was selected according to the micro-averaged F-measure (Loza Mencía and Janssen 2016).

In the following, we first take a closer look on the actual rules, comparing them to the rules induced separately for each label and by separate-and-conquer. Subsequently, we put the focus on visualizing dependencies between labels found by the stacking approach. We refer to Loza Mencía and Janssen (2016) for extensive statistics and more detailed evaluations.

Table 3 Statistics of the used datasets: name of the dataset, domain of the input instances, number of instances, number of nominal/binary and numeric features, total number of unique labels, average number of labels per instance (cardinality), average percentage of relevant labels (label density), number of distinct label sets in the data

Name	Domain	Instances	Nominal	Numeric	Labels	Cardinality	Density	Distinct
EMOTIONS	Music	593	0	72	6	1.869	0.311	27
SCENE	Image	2407	0	294	6	1.074	0.179	15
FLAGS	Image	194	9	10	7	3.392	0.485	54
YEAST	Biology	2417	0	103	14	4.237	0.303	198
BIRDS	Audio	645	2	258	19	1.014	0.053	133
GENBASE	Biology	662	1186	0	27	1.252	0.046	32
MEDICAL	Text	978	1449	0	45	1.245	0.028	94
ENRON	Text	1702	1001	0	53	3.378	0.064	753
CAL500	Music	502	0	68	174	26.0	0.150	502

6.1.1 Exemplary Rule Models

Examples of learned rule sets are shown in Fig. 6. In the case of YEAST, we see a much more compact and less complex rule set for *Class4* for the layered learner *SBR* than for the independently learned *BR* classifier. The rule set also seems more appropriate for a domain expert to understand coherences between proteins (instance features) and protein functions (labels). The separate-and-conquer model *Single* is less explicit in this sense, but it shows that certainly *Class3* is an important class for expressing *Class4*.[6]

Figure 6 also shows the models for the diagnosis *Cough* in the MEDICAL task. This dataset is concerned with the assignment of international diseases codes (ICD) to real, free-text radiological reports. Interestingly, the model found by *SBR* reads very well, and the found relationship seems to be even comprehensible to non-experts. For example, the first rule can be read as

> *If the patient does not have* Pneumonia*, a* Pulmonary_collapse *or* Asthma *and "cough"s or is "coughing", he just has a* Cough*. Otherwise, he may also have a "mild"* Asthma*, in which case he is also considered to have a* Cough.

The theory learned by *Single* is quite similar to the one learned by simple *BR*, which shows that the textual rules were considered to be more powerful than the dependency-based rule. Only at the end, a local dependency is learned: *Cough* only depends on the word "cough" if the label for *Fever* has also been set.

In ENRON, which is concerned with the categorization of emails during the Enron scandal, the learned models are generally less comprehensible. The observed relation between *Personal* and *Joke* can clearly be explained from the hierarchical structure on the topics.

[6] For convenience, we only show the rules with this label in the head.

Approach	YEAST	
BR	*Class4* ← x23 > 0.08, x49 < -0.09 *Class4* ← x68 < 0.05, x33 > 0.00, x24 > 0.00, x66 > 0.00, x88 > -0.06 *Class4* ← x3 < -0.03, x71 > 0.03, x91 > -0.01 *Class4* ← x68 < 0.03, x83 > -0.00, x44> 0.029, x93 < 0.01 *Class4* ← x96 < -0.03, x10 > 0.01, x78< -0.07	
SBR	*Class4* ← *Class3*, $\overline{\textit{Class2}}$ *Class4* ← *Class5*, $\overline{\textit{Class6}}$ *Class4* ← *Class3*, *Class1*, x22 > -0.02	
Single	*Class4* ← *Class3*, x91 > -0.02, x50 < -0.02, x68 < 0.03 *Class4* ← *Class3*, x90 > -0.02, x77 < -0.04 *Class4* ← x60 < -0.03, x57 < -0.07, x19 > -0.01	
	MEDICAL	ENRON
BR	*Cough* ← “cough”, $\overline{\text{“lobe”}}$ *Cough* ← “cough”, “atelectasis” *Cough* ← “cough”, “opacity” *Cough* ← “cough”, “airways” *Cough* ← “cough” , $\overline{\text{“pneumonia”}}$, $\overline{\text{“2”}}$ *Cough* ← “coughing” *Cough* ← “cough”, “early”	*Joke* ← “mail”, “fw”, ”didn”
SBR	*Cough* ← “cough” , $\overline{\textit{Pneumonia}}$, $\overline{\textit{Pulmonary\_collapse}}$, $\overline{\textit{Asthma}}$ *Cough* ← “coughing” *Cough* ← *Asthma*, “mild”	*Joke* ← *Personal*, “day”, “mail”
Single	*Cough* ← “cough”, $\overline{\text{“lobe”}}$, $\overline{\text{“asthma”}}$ *Cough* ← “cough”, “opacity” *Cough* ← “cough”, “atelectasis” *Cough* ← “cough”, “airways” *Cough* ← “cough”, *Fever*	*Joke* ← “didn”, $\overline{\text{“wednesday”}}$ *Joke* ← *Personal*, “forwarded”

Fig. 6 Example rule sets for one exemplary label, respectively, learned by BR, SBR and separate-and-conquer (single)

Regarding the sizes of the models, we found between 50 and 100 rules for YEAST and MEDICAL, and between 76 (*BR*) and 340 (*Single*) for ENRON. Note, however, that even for ENRON this results in an average of only 6.4 rules per label for the largest rule model. Moreover, only a fraction of them are necessary in order to track and comprehend the prediction for a particular test instance. For instance, the report

> *Clinical history: Cough for one month.*
> *Impression: Mild hyperinflation can be seen in viral illness or reactive airway disease. Streaky opacity at the right base is favored to represent atelectasis.*

in the MEDICAL dataset was classified by experts as normal *Cough*, as well as by the rule sets in Fig. 6. Furthermore, the rule models allow to identify the relationships found by the algorithm responsible for the prediction and even the training examples responsible for finding such patterns. This type of inspection may facilitate in a more convenient way than with black box approaches the integration of expert feedback—for instance on the semantic coherence (Gabriel et al. 2014) or plausibility of rules—and also an interaction with the user. For example, Beckerle (2009) explored an

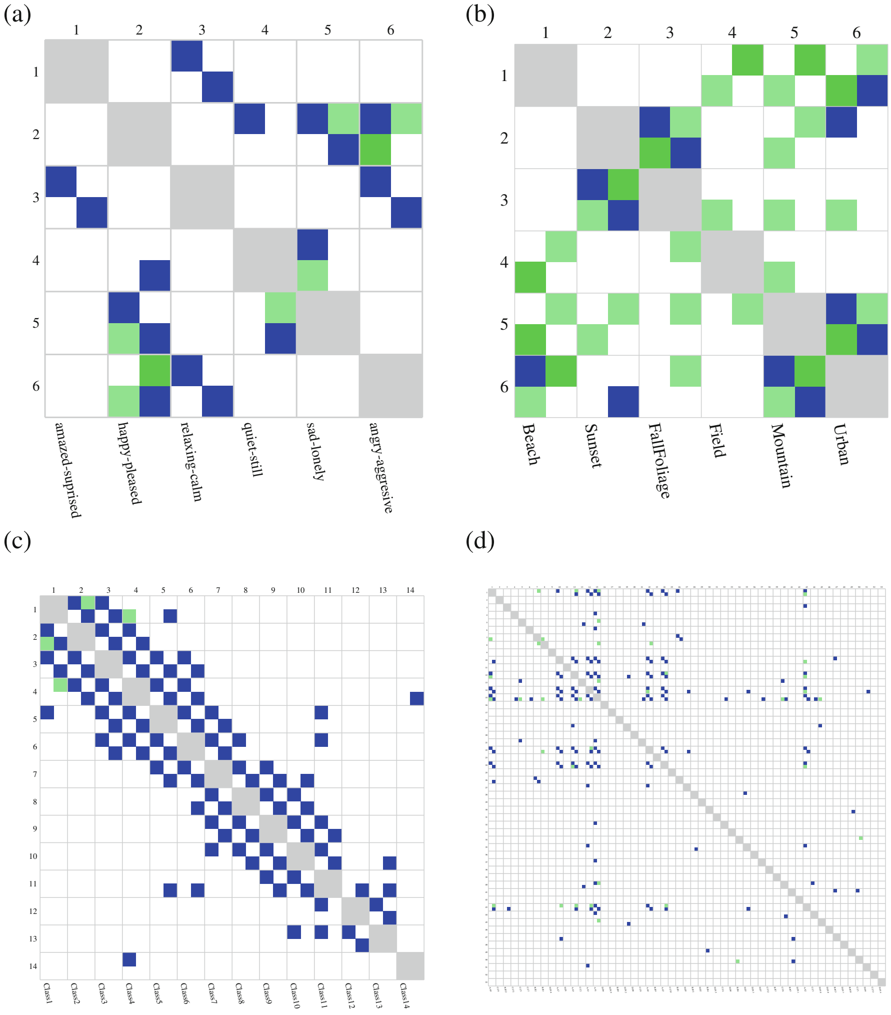


Fig. 7 Visualization of the label-dependent rules for *SBR*. Rows and columns correspond to labels, green entries represent local dependencies and blue entries global dependencies that involve the corresponding label pairs (more details in the text). (**a**) EMOTIONS. (**b**) SCENE. (**c**) YEAST. (**d**) ENRON

interactive rule learning process where learned rules could be directly modified by the user, thereby causing the learner to re-learn subsequently learned rules.

6.1.2 Visualization of Dependencies

Figure 7 shows a graphical representation of the label-dependent rules found by SBR on some of the smaller datasets. Each graph shows a correlation between label pairs. Labels are enumerated from 1 to the number of labels, and the corresponding label names are shown at the bottom of the coordinate system. Blue boxes in the intersection square between a *row label* and a *column label* depict fully label-dependent rules, green boxes show partially label-dependent rules. A colored box at the top corners indicates a rule of the type

$$\textit{row label} \leftarrow \ldots, \textit{column label}, \ldots,$$

whereas the bottom corners represent the opposite

$$\textit{column label} \leftarrow \ldots, \textit{row label}, \ldots$$

rules. As an example, the blue box in the upper left corner of the square in the second row and fourth column in Fig. 7a (EMOTIONS) indicates that the algorithm found a rule of the type

$$\textit{happy} - \textit{pleased} \leftarrow \ldots, \text{quiet-still}, \ldots,$$

i.e., that quiet or still audio sample cause (possibly together with other factors) happy or pleased emotions. Note, however, that the graphs do not show whether the head or conditions are positive or negative.

In particular in SCENE (Fig. 7b), we find many local dependencies, which also depend on some instance features. This is reasonable, since the task in this dataset is to predict elements of a scenery image, and although some label combinations may be more likely than others, whether an element is present or not will still depend on the content of the picture at hand. In YEAST the labels seem to be organized in a special way since we encounter the pattern that a label depends on its preceding and the two succeeding labels. ENRON has a hierarchical structure on its labels, which can be recognized from the vertical and horizontal patterns originating from parent labels.

6.1.3 Discussion

In our experiments, a layered learning approach such as *SBR* proved to be particularly effective at inducing rules with labels as conditions in the bodies of the rules. The resulting models turned out to be indeed very useful for discovering interesting aspects of the data, which a conventional single-label rule learner is unable to uncover. The visualizations shown above also confirm that numerous explicit local and global dependencies can be found in these database. However, we also found that the GENBASE dataset exhibits only very weak label dependencies, which can hardly be exploited in order to improve the predictive performance, despite the fact that this dataset is frequently used for evaluating multi-label algorithms.

$\mathit{red}, \mathit{green}, \mathit{blue}, \mathit{yellow}, \mathit{white} \leftarrow \text{colors}>5, \text{stripes}\leq 3$	(65,0)
$\overline{\mathit{red}}, \mathit{green}, \overline{\mathit{blue}}, \mathit{yellow}, \mathit{white}, \overline{\mathit{black}}, \overline{\mathit{orange}} \leftarrow \text{animate}, \text{stripes}\leq 0, \text{crosses}\leq 0$	(11,0)

$\mathit{yellow} \leftarrow \text{colors}>4$	(21,0)	$\mathit{green} \leftarrow \text{text}$	(11,0)	
$\mathit{red} \leftarrow \mathit{yellow}$	(21,0)	$\overline{\mathit{orange}} \leftarrow \text{saltires}<1$	(1,0)	
$\mathit{blue} \leftarrow \text{colors}>5$	(14,0)	$\overline{\mathit{black}} \leftarrow \text{area}<11$	(12,0)	
$\mathit{white} \leftarrow \mathit{blue}$	(14,0)			

Fig. 8 Example of learned multi- and single-label head rule lists learned in the FLAGS dataset. In parentheses, we show (TP, FP), the number of positive and negative examples covered by each rule. Shown are all rules that cover the flag of the US Virgin Islands, which is shown in the lower right corner

6.2 *Case Study 2: Multi-Label Heads*

The second case study compares *BR*, *Single* and *Multi* for candidate rule selection (Rapp et al. 2018).[7] Its main purpose was to demonstrate the applicability of a covering approach for inducing multi-label head rules despite the exponentially large search space.

6.2.1 Exemplary Rule Models

The extended expressiveness of multi-label head rules can be illustrated by the rules shown in Fig. 8 that have been learned on the data set FLAGS, which maps characteristics of a flag and its corresponding country to the colors appearing on the flag. Shown are all rules that concern the flag of the US Virgin Islands, which is also shown in the table. Whereas in this case the single-label heads allow an easier visualization of the pairwise dependencies between characteristics/labels and labels, the multi-label head rules allow to represent more complex relationships and provide a more direct explanation of why the respective colors are predicted for the flag. Note that the rules form decision lists, which are applied in order until all labels are set, and later rules cannot overwrite earlier rules. Thus the first rule sets the colors *red*, *green*, *blue*, *yellow*, and *white*, whereas the second rule determines that *black* and *orange* do not occur. The other labels are already set by the previous rule and are not overwritten. No further rules would be considered for the prediction because all labels are already assigned.

This example also illustrates that while decision lists are conceptually easy to extend to the multi-label case by removing covered labels, the interpretability of the resulting rules may suffer. Learning rule sets that collectively determine

[7] The source code of the employed algorithms and more extensive evaluations are available at https://github.com/keelm/SeCo-MLC.

the predicted label set from multiple possibly overlapping or contradicting partial predictions is an open question for future work.

Whether more labels in the head are more desirable or not highly depends on the data set at hand, the particular scenario and the preferences of the user, as generally do comprehensibility and interpretability of rules. These issues cannot be solved by the presented methods. However, the flexibility of being able to efficiently find loss-minimizing multi-label heads for a variety of loss functions can lay the foundation to further improvements, gaining better control over the characteristics of the induced model and hence better adaption to the requirements of a particular use case.

When analyzing the general characteristics of the models which have been learned by the proposed algorithm, it becomes apparent that multi-label head rules are particularly learned when using the precision metric, rather than one of the other metrics. The reason is that precision only considers the covered examples whereas for the other metrics the performance also depends on uncovered examples. Hence, it is very likely that the performance of a rule slightly decreases when adding an additional label to its head, which in turn causes single-label heads to be preferred.[8]

6.2.2 Predictive Performance

Because of this bias towards single-label rules for most of the metrics, large differences in predictive performance of single-label and multi-label head decision lists cannot be expected. We therefore only summarize the main finding, which compared the algorithms' performance using a Friedman test (Friedman 1937) and a Nemenyi post-hoc test (Nemenyi 1963) following the methodology described by Demšar (2006). The null hypothesis of the Friedman test ($\alpha = 0.05, N = 8, k = 10$) that all 10 algorithms have the same predictive quality on the eight datasets shown in Fig. 3 (excluding ENRON) could not be rejected for many of the evaluation measures, such as subset accuracy and micro- and macro-averaged F1. In the other cases, the Nemenyi post-hoc test was not able to assess a statistical difference between different algorithms that used the same objective for optimizing the rules.

6.2.3 Computational Cost

Figure 9 shows the relation between the time spent for finding single- vs. multi-label head rules using the same objective and data set. The empty forms denote the single-label times multiplied by the number of labels in the data set and represent an approach with a computational complexity increased by one polynomial order w.r.t. number of labels. Note that full exploration of the labels space was already intractable for the smaller data sets on our system, and became only feasible through

[8] The inclusion of a factor which takes the head's size in account could resolve this bias and lead to heads with more labels, but this is subject to future work.

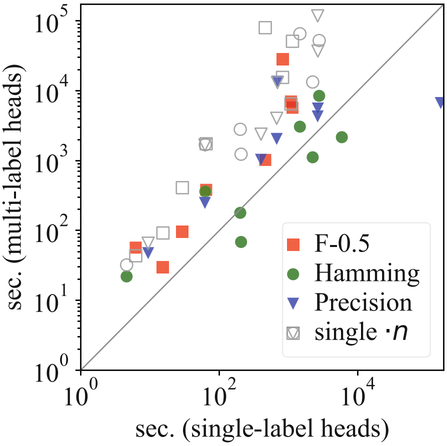


Fig. 9 Training times for the separate-and-conquer algorithm. Direct comparison between learning single-label and multi-label heads

the use of anti-monotonicity and decomposability pruning, as described in Sect. 4.2. We can observe that the costs for learning multi-label head rules are in the same order of magnitude as the costs for learning single-label head rules, despite the need for exploring the full label space for each candidate body.

7 Conclusion

In this work, we recapitulated recent work on inductive rule learning for multi-label classification problems. The main advantage of such an approach is that mixed dependencies between input variables and labels can be seamlessly integrated into a single coherent representation, which facilitates the interpretability of the learned multi-label models. However, we have also seen that combining multi-label rules into interpretable predictive theories faces several problems, which are not yet sufficiently well addressed by current solutions. One problem is that mixed-dependency rules needs to be structured in a way that allows each label that occurs in the body of a rule to be predicted by some other rule in a way that avoids cyclic reasoning. We have seen two principal approaches to solve this problem, a layered technique that relies on a pre-defined structure of the prediction and rule induction process, and a second approach that relies on adapting the separate-and-conquer or covering strategy from single-label rule learning to the multi-label case. The results we have shown in several domains are encouraging, but it is also clear that they are still somewhat limited. For example, the multi-label decision lists that result from the latter approach are hard to interpret because of the implicit dependencies that are captured in the sequential interpretation of the rules. Thus, multi-label rule learning remains an interesting research goal, which combines challenging algorithmic problems with a strong application potential.

Acknowledgements We would like to thank Frederik Janssen for his contributions to this work. Computations for this research were conducted on the Lichtenberg high performance computer of the TU Darmstadt.

References

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. In: Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.) Advances in Knowledge Discovery and Data Mining, pp. 307–328. AAAI Press (1995)

Allahyari, H., Lavesson, N.: User-oriented assessment of classification model understandability. In: Kofod-Petersen, A., Heintz, F., Langseth, H. (eds.) Proceedings of the 11th Scandinavian Conference on Artificial Intelligence (SCAI-11). Frontiers in Artificial Intelligence and Applications, vol. 227, pp. 11–19. IOS Press, Trondheim, Norway (2011)

Allamanis, M., Tzima, F., Mitkas, P.: Effective Rule-Based Multi-label Classification with Learning Classifier Systems. In: Adaptive and Natural Computing Algorithms, 11th International Conference, ICANNGA 2013. pp. 466–476 (2013)

Arunadevi, J., Rajamani, V.: An evolutionary multi label classification using associative rule mining for spatial preferences. IJCA Special Issue on Artificial Intelligence Techniques - Novel Approaches and Practical Applications (3), 28–37 (2011)

Atzmüller, M.: Subgroup discovery. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 5(1), 35–49 (2015)

Ávila, J., Galindo, E., Ventura, S.: Evolving Multi-label Classification Rules with Gene Expression Programming: A Preliminary Study. In: Hybrid Artificial Intelligence Systems. vol. 6077, pp. 9–16. Springer (2010)

Beckerle, M.: Interaktives Regellernen. Diploma thesis, Technische Universtität Darmstadt (2009), in German

Bosc, G., Golebiowski, J., Bensafi, M., Robardet, C., Plantevit, M., Boulicaut, J.F., Kaytoue, M.: Local subgroup discovery for eliciting and understanding new structure-odor relationships. In: Calders, T., Ceci, M., Malerba, D. (eds.) Proceedings of the 19th International Conference on Discovery Science (DS-16). Lecture Notes in Computer Science, vol. 9956, pp. 19–34. Bari, Italy (2016)

Boutell, M.R., Luo, J., Shen, X., Brown, C.M.C.M.: Learning multi-label scene classification. Pattern Recognition 37(9), 1757–1771 (2004)

Bryce Goodman, S.F.: European union regulations on algorithmic decision-making and a “right to explanation”. In: Proceedings of the 2016 ICML Workshop on Human Interpretability in Machine Learning (WHI 2016). pp. 26–30 (2016), arXiv:1606.08813 [stat.ML]

Cameron-Jones, R.M., Quinlan, J.R.: Avoiding pitfalls when learning recursive theories. In: Bajcsy, R. (ed.) Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI-93). pp. 1050–1057. Chambéry, France (1993)

Charte, F., Rivera, A.J., del Jesús, M.J., Herrera, F.: LI-MLC: A label inference methodology for addressing high dimensionality in the label space for multilabel classification. IEEE Transactions on Neural Networks and Learning Systems 25(10), 1842–1854 (2014)

Chekina, L., Gutfreund, D., Kontorovich, A., Rokach, L., Shapira, B.: Exploiting label dependencies for improved sample complexity. Machine Learning 91(1), 1–42 (2013)

Cohen, W.W.: Fast effective rule induction. In: Prieditis, A., Russell, S. (eds.) Proceedings of the 12th International Conference on Machine Learning (ML-95). pp. 115–123. Morgan Kaufmann, Lake Tahoe, CA (1995)

De Raedt, L., Lavrač, N., Džeroski, S.: Multiple predicate learning. In: Bajcsy, R. (ed.) Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI-93). pp. 1037–1043. Morgan Kaufmann, Chambéry, France (1993)

Dembczyński, K., Kotłowski, W., SłowiÅĎski, R.: ENDER: a statistical framework for boosting decision rules. Data Mining and Knowledge Discovery 21(1), 52–90 (2010)

Dembczyński, K., Waegeman, W., Cheng, W., Hüllermeier, E.: On label dependence and loss minimization in multi-label classification. Machine Learning 88(1–2), 5–45 (2012)

Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)

Duivesteijn, W., Feelders, A., Knobbe, A.J.: Exceptional model mining – supervised descriptive local pattern mining with complex target concepts. Data Mining and Knowledge Discovery 30(1), 47–98 (2016)

Duivesteijn, W., Loza Mencía, E., Fürnkranz, J., Knobbe, A.J.: Multi-label lego – enhancing multi-label classifiers with local patterns. In: Hollmén, J., Klawonn, F., Tucker, A. (eds.) Advances in Intelligent Data Analysis XI – Proceedings of the 11th International Symposium on Data Analysis (IDA-11). Lecture Notes in Computer Science, vol. 7619, pp. 114–125. Springer (2012)

Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Dietterich, T.G., Becker, S., Ghahramani, Z. (eds.) Advances in Neural Information Processing Systems. vol. 14, pp. 681–687. MIT Press (2001)

Freitas, A.A.: Comprehensible classification models: a position paper. SIGKDD Explorations 15(1), 1–10 (2013)

Friedman, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the American Statistical Association 32, 675–701 (1937)

Fürnkranz, J.: Separate-and-conquer rule learning. Artificial Intelligence Review 13(1), 3–54 (1999)

Fürnkranz, J.: From local to global patterns: Evaluation issues in rule learning algorithms. In: Morik, K., Boulicaut, J.F., Siebes, A. (eds.) Local Pattern Detection. pp. 20–38. Springer-Verlag (2005)

Fürnkranz, J., Gamberger, D., Lavrač, N.: Foundations of Rule Learning. Springer-Verlag (2012)

Fürnkranz, J., Kliegr, T., Paulheim, H.: On cognitive preferences and the interpretability of rule-based models. arXiv preprint arXiv:1803.01316 (2018)

Gabriel, A., Paulheim, H., Janssen, F.: Learning semantically coherent rules. In: Cellier, P., Charnois, T., Hotho, A., Matwin, S., Moens, M.F., Toussaint, Y. (eds.) Proceedings of the 1st International Workshop on Interactions between Data Mining and Natural Language Processing co-located with The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2014). vol. 1202, pp. 49–63. CEUR Workshop Proceedings, Nancy, France (2014)

Gibaja, E., Ventura, S.: Multi-label learning: a review of the state of the art and ongoing research. Wiley Interdisciplinary Review: Data Mining and Knowledge Discovery 4(6), 411–444 (2014)

Gibaja, E., Ventura, S.: A tutorial on multilabel learning. ACM Comput. Surv. 47(3), 52 (2015)

Godbole, S., Sarawagi, S.: Discriminative methods for multi-labeled classification. In: Advances in Knowledge Discovery and Data Mining (PAKDD 2004). pp. 22–30 (2004)

Goethals, B.: Frequent set mining. In: Maimon, O., Rokach, L. (eds.) The Data Mining and Knowledge Discovery Handbook, pp. 377–397. Springer-Verlag (2005)

Guo, Y., Gu, S.: Multi-label classification using conditional dependency networks. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Two. pp. 1300–1305. IJCAI'11, AAAI Press (2011)

Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Mining and Knowledge Discovery 8(1), 53–87 (2004)

Hayes, P.J., Weinstein, S.P.: CONSTRUE/TIS: A system for content-based indexing of a database of news stories. In: Rappaport, A.T., Smith, R.G. (eds.) Proceedings of the 2nd Conference on Innovative Applications of Artificial Intelligence (IAAI-90), May 1–3, 1990, Washington, DC, USA. pp. 49–64. IAAI '90, AAAI Press, Chicago, IL, USA (1991)

Herrera, F., Charte, F., Rivera, A.J., del Jesús, M.J.: Multilabel Classification - Problem Analysis, Metrics and Techniques. Springer (2016)

Hipp, J., Güntzer, U., Nakhaeizadeh, G.: Algorithms for association rule mining – a general survey and comparison. SIGKDD explorations 2(1), 58–64 (2000)

Janssen, F., Fürnkranz, J.: On the quest for optimal rule learning heuristics. Machine Learning 78(3), 343–379 (2010)

Knobbe, A.J., Crémilleux, B., Fürnkranz, J., Scholz, M.: From local patterns to global models: The LeGo approach to data mining. In: Knobbe, A.J. (ed.) From Local Patterns to Global Models: Proceedings of the ECML/PKDD-08 Workshop (LeGo-08). pp. 1–16. Antwerp, Belgium (2008)

Kralj Novak, P., Lavrač, N., Webb, G.I.: Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. Journal of Machine Learning Research 10, 377–403 (2009)

Lewis, D.D.: An evaluation of phrasal and clustered representations on a text categorization task. In: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Devlopment in Information Retrieval. pp. 37–50 (1992)

Lewis, D.D.: Reuters-21578 text categorization test collection distribution 1.0. README file (V 1.3) (2004)

Li, B., Li, H., Wu, M., Li, P.: Multi-label Classification based on Association Rules with Application to Scene Classification. In: Proceedings of the 2008 The 9th International Conference for Young Computer Scientists. pp. 36–41. IEEE Computer Society (2008)

Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: Agrawal, R., Stolorz, P., Piatetsky-Shapiro, G. (eds.) Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD-98). pp. 80–86 (1998)

Liu, B., Ma, Y., Wong, C.K.: Improving an exhaustive search based rule learner. In: Zighed, D.A., Komorowski, H.J., Zytkow, J.M. (eds.) Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-2000). pp. 504–509. Lyon, France (2000)

Loza Mencía, E., Janssen, F.: Stacking label features for learning multilabel rules. In: Discovery Science - 17th International Conference, DS 2014, Bled, Slovenia, October 8–10, 2014, Proceedings, Lecture Notes in Computer Science, vol. 8777, pp. 192–203. Springer (2014)

Loza Mencía, E., Janssen, F.: Learning rules for multi-label classification: a stacking and a separate-and-conquer approach. Machine Learning 105(1), 77–126 (2016)

Malerba, D.: Learning recursive theories in the normal ilp setting. Fundamenta Informaticae 57(1), 39–77 (2003)

Malerba, D., Semeraro, G., Esposito, F.: A multistrategy approach to learning multiple dependent concepts. In: Machine Learning and Statistics: The Interface, chap. 4, pp. 87–106 (1997)

Minnaert, B., Martens, D., Backer, M.D., Baesens, B.: To tune or not to tune: Rule evaluation for metaheuristic-based sequential covering algorithms. Data Mining and Knowledge Discovery 29(1), 237–272 (2015)

Montañés, E., Senge, R., Barranquero, J., Quevedo, J.R., del Coz, J.J., Hüllermeier, E.: Dependent binary relevance models for multi-label classification. Pattern Recognition 47(3), 1494–1508 (2014)

Nemenyi, P.: Distribution-free multiple comparisons. Ph.D. thesis, Princeton University (1963)

Papagiannopoulou, C., Tsoumakas, G., Tsamardinos, I.: Discovering and exploiting deterministic label relationships in multi-label learning. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 915–924. KDD '15, ACM, New York, NY, USA (2015)

Park, S.H., Fürnkranz, J.: Multi-label classification with label constraints. In: Hüllermeier, E., Fürnkranz, J. (eds.) Proceedings of the ECML PKDD 2008 Workshop on Preference Learning (PL-08, Antwerp, Belgium). pp. 157–171 (2008)

Rapp, M.: A Separate-and-Conquer Algorithm for Learning Multi-Label Head Rules. Master thesis, TU Darmstadt, Knowledge Engineering Group (2016)

Rapp, M., Loza Mencía, E., Fürnkranz, J.: Exploiting anti-monotonicity of multi-label evaluation measures for inducing multi-label rules. In: Proceedings of the 22nd Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-18). Springer-Verlag (2018), to appear

Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Machine Learning 85(3), 333–359 (2011)

Senge, R., del Coz, J.J., Hüllermeier, E.: On the problem of error propagation in classifier chains for multi-label classification. In: Spiliopoulou, M., Schmidt-Thieme, L., Janning, R. (eds.) Proceedings of the 36th Annual Conference of the Gesellschaft für Klassifikation (GfKl-12). pp. 163–170. Hildesheim, Germany (2012)

Stecher, J., Janssen, F., Fürnkranz, J.: Shorter rules are better, aren't they? In: Calders, T., Ceci, M., Malerba, D. (eds.) Proceedings of the 19th International Conference on Discovery Science (DS-16). pp. 279–294. Springer-Verlag (2016)

Sucar, L.E., Bielza, C., Morales, E.F., Hernandez-Leal, P., Zaragoza, J.H., Larrañaga, P.: Multi-label classification with Bayesian network-based chain classifiers. Pattern Recognition Letters 41, 14–22 (2014)

Sulzmann, J.N., Fürnkranz, J.: A comparison of techniques for selecting and combining class association rules. In: Knobbe, A.J. (ed.) From Local Patterns to Global Models: Proceedings of the ECML/PKDD-08 Workshop (LeGo-08). pp. 154–168. Antwerp, Belgium (2008)

Thabtah, F., Cowling, P., Peng, Y.: MMAC: A New Multi-Class, Multi-Label Associative Classification Approach. In: Proceedings of the 4th IEEE ICDM. pp. 217–224 (2004)

Thabtah, F., Cowling, P., Peng, Y.: Multiple labels associative classification. Knowledge and Information Systems 9(1), 109–129 (2006)

Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.P.: Multilabel classification of music into emotions. In: Proc. 9th International Conference on Music Information Retrieval (ISMIR 2008). pp. 325–330 (2008)

Tsoumakas, G., Katakis, I., Vlahavas, I.P.: Mining Multi-label Data. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 667–685 (2010)

Tsoumakas, G., Zhang, M., Zhou, Z.: Introduction to the special issue on learning from multi-label data. Machine Learning 88(1–2), 1–4 (2012)

Varma, M., Cissé, M. (eds.): Proceedings of the NIPS-15 Workshop on Extreme Classification: Multi-class and Multi-label Learning in Extremely Large Label Spaces (XC-15) (2015)

Veloso, A., Meira, Jr., W., Gonçalves, M., Zaki, M.: Multi-label lazy associative classification. In: Proceedings of the 11th European conference on Principles and Practice of Knowledge Discovery in Databases. pp. 605–612. PKDD 2007 (2007)

Waegeman, W., Dembczyńki, K., Jachnik, A., Cheng, W., Hüllermeier, E.: On the bayes-optimality of f-measure maximizers. Journal of Machine Learning Research 15(1), 3333–3388 (2014)

Webb, G.I.: Recent progress in learning decision lists by prepending inferred rules. In: Proceedings of the 2nd Singapore International Conference on Intelligent Systems. pp. B280–B285 (1994)

Webb, G.I.: Efficient search for association rules. In: Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2000). pp. 99–107. Boston, MA (2000)

Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W.: New algorithms for fast discovery of association rules. In: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD-97). pp. 283–286. Newport, CA (1997)

Zhang, C., Zhang, S.: Association Rule Mining: Models and Algorithms. Springer-Verlag (2002)

Zhang, M., Zhou, Z.: A review on multi-label learning algorithms. IEEE Transactions on Knowledge and Data Engineering 26(8), 1819–1837 (2014)

Structuring Neural Networks for More Explainable Predictions

Laura Rieger, Pattarawat Chormai, Grégoire Montavon, Lars Kai Hansen, and Klaus-Robert Müller

Abstract Machine learning algorithms such as neural networks are more useful, when their predictions can be explained, e.g. in terms of input variables. Often simpler models are more interpretable than more complex models with higher performance. In practice, one can choose a readily interpretable (possibly less predictive) model. Another solution is to directly explain the original, highly predictive model. In this chapter, we present a middle-ground approach where the original neural network architecture is modified parsimoniously in order to reduce common biases observed in the explanations. Our approach leads to explanations that better separate classes in feed-forward networks, and that also better identify relevant time steps in recurrent neural networks.

Keywords Interpretable machine learning · Convolutional neural networks · Recurrent neural networks

Laura Rieger and Pattarawat Chormai contributed equally to this work.

L. Rieger (✉) · L. K. Hansen
DTU Compute, Technical University of Denmark, Kongens Lyngby, Denmark
e-mail: lauri@dtu.dk; lkai@dtu.dk

P. Chormai · G. Montavon
Department of Electrical Engineering and Computer Science, Technische Universität Berlin, Berlin, Germany
e-mail: p.chormai@campus.tu-berlin.d; gregoire.montavon@tu-berlin.de

K.-R. Müller (✉)
Department of Electrical Engineering and Computer Science, Technische Universität Berlin, Berlin, Germany

Department of Brain and Cognitive Engineering, Korea University, Seongbuk-gu, Seoul, South Korea

Max Planck Institute for Informatics, Saarbrücken, Germany
e-mail: klaus-robert.mueller@tu-berlin.de

H. J. Escalante et al. (eds.), *Explainable and Interpretable Models in Computer Vision and Machine Learning*, The Springer Series on Challenges in Machine Learning, https://doi.org/10.1007/978-3-319-98131-4_5

1 Introduction

Neural networks are powerful learning machines that derive their power from the interconnection of a large number of elementary computational units (neurons). A significant body of work has focused on finding appropriate neural network structures for specific problems (LeCun 1989; Jarrett et al. 2009; Collobert et al. 2011). For example, on image classification tasks, convolution-type architectures have proven to be highly efficient (Krizhevsky et al. 2012; Szegedy et al. 2015). Similar models are also being used increasingly in fields such as computational biology (Angermueller et al. 2016) or physics (Schütt et al. 2017).

Motivated by these successes, there is a renewed interest in developing techniques to interpret how these highly predictive neural network models reach their decisions. Some explanation techniques choose the architecture in a way that it becomes interpretable, by defining the function as a simple sum over readily interpretable quantities (Poulin et al. 2006; Caruana et al. 2015; Zhou et al. 2016). Other methods seek to explain a more general set of deep neural network architectures (Simonyan et al. 2013; Zeiler and Fergus 2014; Bach et al. 2015; Ribeiro et al. 2016). Three methods, sensitivity analysis (Gevrey et al. 2003; Baehrens et al. 2010; Simonyan et al. 2013), guided backprop (Springenberg et al. 2014) and deep Taylor decomposition (Montavon et al. 2017), all of them applicable to sequences of linear and ReLU layers, will be considered in this paper.

The paper asks the question whether high prediction accuracy is a sufficient condition for high explanation quality, and what additional steps are then necessary to also reach high explainability. The role of regularization and the interplay between performance and robustness of global sensitivity maps has been investigated, e.g. in Rasmussen et al. (2012). Here we focus on interpretability of the individual decisions. More precisely, we will test whether explanations exhibit a systematic bias, i.e. a constant divergence between the features identified by the explanation technique and the actual features used by the model to predict, and how the structure of the neural network can be adapted to reduce such bias. Section 2 introduces the explanation techniques considered in this paper. In Sects. 3 and 4, we present examples of highly predictive models for which explanations are difficult to extract, and how simple and parsimonious structural modifications of the neural network allow to maintain high predictive accuracy, while improving the explanations.

2 Explanation Techniques

This section reviews a set of techniques for explaining the decisions made by neural networks. It focuses on sequences of linear and ReLU layers. Highly predictive convolutional neural networks (CNNs) or recurrent neural networks (RNNs) can be built from these sequences of layers. Let $\mathbf{x} = (x_1, \ldots, x_d)$ be the d-dimensional input presented to the neural network, and $f(\mathbf{x})$ the value of some output neuron.

We focus on explanation methods, that aim to score input relevance according to additive contributions to the function output. An explanation is defined as a vector of scores $(R_1, \ldots, R_d)$ identifying the contribution of each input variable to the function value $f(\mathbf{x})$.

2.1 Sensitivity Analysis

A common way of defining these scores is based on the locally evaluated gradient $\nabla_{\mathbf{x}} f(\mathbf{x})$. The gradient can be efficiently computed with the backpropagation algorithm. Consider a deep network composed of multiple layers, where each layer is composed of a linear transformation followed by an element-wise ReLU nonlinearity. Letting j and k index neurons of two consecutive layers, activations $(a_j)_j$ and $(a_k)_k$ in the respective layers can be related as $a_k = \max(0, z_k)$, where $z_k = \sum_j a_j w_{jk} + b_k$ is called the pre-activation.

The backpropagation algorithm transmits partial derivatives from the top of the network to the input by repeated application of the chain rule. Let δ_j and δ_k be a shortcut notation for the locally evaluated partial derivatives $\partial f/\partial z_j$ and $\partial f/\partial z_k$. In this network, the chain-rule equation for propagating derivatives is

$$\delta_j = 1_{z_j > 0} \cdot \textstyle\sum_k w_{jk} \delta_k \tag{1}$$

in the hidden layers, and $\delta_i = \sum_j w_{ij} \delta_j$ for the first layer. The gradients are propagated until the input variables, where they can be converted to importance scores, e.g. by squaring ($R_i = \delta_i^2$). We refer to this way of setting importance scores as *sensitivity analysis* (SA). Explanation through sensitivity analysis has been used, e.g. by Gevrey et al. (2003), Baehrens et al. (2010), and Simonyan et al. (2013). Sensitivity analysis as well as other methods relying on the gradient assume that the function value is not varying too quickly in the input domain. This assumption usually does not hold for deep networks, where the function becomes steeper and higher-frequency with every added layer, leading to an uninformative gradient (Balduzzi et al. 2017).

To remediate to this problem, alternate propagation rules can be applied, for example the *guided backprop* (GB) technique (Springenberg et al. 2014) applies the modified rule

$$\tilde{\delta}_j = 1_{z_j > 0} \cdot \max\left(0, \textstyle\sum_k w_{jk} \tilde{\delta}_k\right) \tag{2}$$

which rectifies the incoming gradient and therefore prevents inhibitory effects to propagate. The propagated signal is no longer a gradient, but still retains a rough interpretation as a local direction of variation. Like for the gradient, the result of the propagation procedure can be converted to importance scores by squaring, i.e. $R_i = \tilde{\delta}_i^2$.

In general, methods relying solely on the gradient or similar quantities are in essence closer to an explanation of the function's variation than of the function value itself: For example, sensitivity scores relate to the function as: $\sum_i R_i = \sum_i \delta_i^2 = \|\nabla_{\mathbf{x}} f(\mathbf{x})\|^2$, i.e. the scores are a decomposition of the function's local slope (Montavon et al. 2018). Stated otherwise, these methods explain why the function varies strongly locally, however, they do not explain why the function has high value locally.

2.2 Deep Taylor Decomposition

To explain the function's value we aim for an importance score that directly relates to $f(\mathbf{x})$. A number of works have proposed to attribute importance scores subject to the conservation constraint $\sum_i R_i = f(\mathbf{x})$, and where these scores are computed using a specific graph propagation procedure (Landecker et al. 2013; Bach et al. 2015; Zhang et al. 2016; Montavon et al. 2017; Shrikumar et al. 2017). Unlike gradient-based methods, the quantity propagated at each neuron is no longer the partial derivatives δ_j, δ_k or some variant of it, but importance scores R_j, R_k. In the following, we present the *deep Taylor decomposition* (DTD) approach (Montavon et al. 2017) to explaining $f(\mathbf{x})$, for which rules specific to deep networks with ReLU nonlinearities were derived. The DTD propagation rule between two hidden layers is given by

$$R_j = \sum_k \frac{a_j w_{jk}^+}{\sum_j a_j w_{jk}^+} R_k, \tag{3}$$

where $w_{jk}^+ = \max(0, w_{jk})$. The intuition for this rule is to redistribute the function value based on the excitation incurred by neurons in the lower-layer onto neurons of the current layer. This rule also has an interpretation as a Taylor decomposition of relevance R_k in the space of positive activations $(a_j)_j \in \mathbb{R}_+$. Another DTD rule specific to the input layer receiving as input pixel intensities $x_i \in [l_i, h_i]$ is given by

$$R_i = \sum_j \frac{x_i w_{ij} - l_i w_{ij}^+ - h_i w_{ij}^-}{\sum_i x_i w_{ij} - l_i w_{ij}^+ - h_i w_{ij}^-} R_j, \tag{4}$$

with $w_{ij}^+ = \max(0, w_{ij})$ and $w_{ij}^- = \min(0, w_{ij})$. A strict application of the DTD method imposes as additional requirements the absence of positive bias parameters and the ability to represent the concept to explain as a top-layer ReLU neuron. More details and theoretical justification for the DTD propagation rules are given in the original paper (Montavon et al. 2017).

2.3 Theoretical Limitations

The methods presented above, namely sensitivity analysis, guided backprop, and deep Taylor decomposition, are in principle applicable to a broad range of architectures, including shallow or deep ones, fully or locally connected, as well as recurrent architectures.

However, despite this broad applicability, the quality of explanation can differ strongly depending on the structure of the neural network. Unlike predictions made by the deep network, explanations are not what the network is trained for and come as a by-product instead. The fact that the model is not optimized for explanation error implies a possibly strong divergence from a ground truth explanation. We identify two potential sources of divergence:

The first source of divergence is gradient noise, and affects SA: Although a function $f(\mathbf{x})$ may be close to the ground truth $f^\star(\mathbf{x})$ in terms of function value (i.e. $\forall_\mathbf{x} : \|f(\mathbf{x}) - f^\star(\mathbf{x})\| \leq \epsilon$), the gradient of the function, on which sensitivity analysis is based, can still be made uncontrollably large (Snyder et al. 2015, 2012; Montúfar et al. 2014; Balduzzi et al. 2017). As a consequence, the resulting explanations are no longer selective of the target concept to explain. A corollary of this gradient noise in the context of RNN architecture is the exploding gradients problem (Bengio et al. 1994; Pascanu et al. 2013), where a finite variation in the output space can be accompanied by a very large gradient in the space representing the older time steps.

The second source of divergence arises from attempts by explanation techniques such as GB or DTD to reduce gradient noise: For example, the gradient rectification applied by GB makes the procedure more stable than the actual gradient, however, the rectification operation can bias the explanation towards certain types of features in a CNN or certain time steps in a RNN. DTD also strongly departs from the actual gradient by redistributing only based on positive weights and activations in the hidden layers.

In the next two sections, we characterize these sources of divergence in the context of CNNs and RNNs, and propose to modify the neural network architecture specifically for reducing them.

3 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a special category of neural networks that have come to attention in the last years due to their great success in tasks such as image classification (Krizhevsky et al. 2012; Szegedy et al. 2015). The first layers extract simple features at various locations and build some translation invariance, and the last layers map these features to the final concepts (e.g. image categories). The explanation problem can here be defined as finding which pixels are responsible for a certain classification decision produced at the output of the network.

While evidence for some classes originates from the same pixels (e.g. these classes share some of the low-level features), other semantically less related classes correspond to distinct features in the image, and we would like the explanation to better capture these features.

As an example, an image of *trousers* from the FashionMNIST dataset in Fig. 4 has the flared outline of a dress but otherwise resembles *trousers*. We would expect a heatmap for *trousers* to focus more on the gap between the legs and a heatmap for *dress* to focus more on the flared outline.

An explanation method must therefore be able to identify pixels that are truly relevant for a specific class of interest, and that are not simply relevant in general.

Our hypothesis is that all classes share a common salient component of representation, and that discrimination between classes does not occur as the effect of building individual class-specific features but rather as measuring small differences on this salient component. While this strategy is perfectly viable for the purpose of prediction, any explanation technique that deviates too much from the function itself and that relies instead on the graph structure might be biased with respect to this salient component.

In the following, we analyze the quality of explanations with respect to the structure of CNNs, specifically the level of connectivity of the dense layers, which controls how fast the backpropagated signal mixes between classes. Specifically, we want to build a structure that encourages the use of separate features for different classes.

We consider three different levels of connectedness, depicted in Fig. 1:

Structure 1: Unrestricted
No restriction is applied to the dense layers of the neural network. That is, if f is the function implemented by the neural network, we simply solve

$$\min_{w,b} \quad J_{\text{emp}}(f)$$

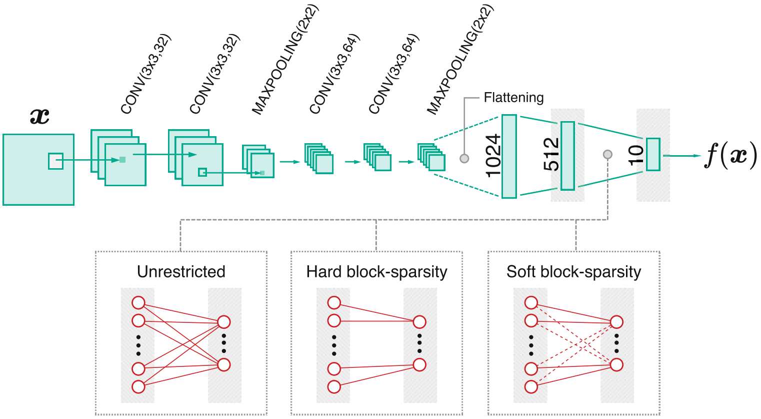


Fig. 1 CNN with various levels of connectedness for the dense layers

that is the standard neural network objective, by minimizing the cost $J_{\text{emp}}(f)$ over the network weights w and biases b. This is our baseline scenario.

Structure 2: Hard Block-Sparsity
Here, we force the weight matrix of the dense layers to have block-diagonal structure so that the classes only recoup near the convolutions layers. That is, we solve the constrained optimization problem

$$\min_{w,b} \; J_{\text{emp}}(f) : \forall_l \forall_{i,j} : w_{ij}^{(l)} = 0 \text{ if } C(i) \neq C(j),$$

where $w_{ij}^{(l)}$ is the weight connecting the ith neuron in layer $l-1$ to the jth neuron in layer l, $\forall_l$ spans the last few dense layers, $\forall_{i,j}$ spans the input and output neurons of the current layer, and $C(i)$ and $C(j)$ are the classes for which neuron i and j are reserved respectively. Practically, the constraint can be enforced at each iteration by multiplying the weight matrix by a mask with block-diagonal structure, or can instead be implemented by splitting the neural network near the output into several pathways, each of which predicts a different class.

Structure 3: Soft Block-Sparsity
In this last setting, the connectivity constraint is replaced by a L1-penalty on all weights that are outside the block-diagonal structure. The optimization problem is rewritten as

$$\min_{w,b} \; J_{\text{emp}}(f) + \lambda \textstyle\sum_{l,i,j} |w_{ij}^{(l)}| \cdot 1_{C(i) \neq C(j)}, \tag{5}$$

with the same definitions as in Structure 2 and additionally λ controlling the level of sparsity. For DTD, because negative weights are not used in the backward propagation, we can further soften the regularization constraint to only penalize positive weights, i.e. we replace $|w_{ij}^{(l)}|$ by $\max(0, w_{ij}^{(l)})$ in the equation above. We call these two variants L1 and L1+.

3.1 Experiments

We trained several CNNs on the MNIST, FashionMNIST, and CIFAR10 datasets (LeCun and Cortes 2010; Xiao et al. 2017; Krizhevsky 2009). The neural network used for CIFAR10 is shown in Fig. 1, and the neural networks used for the two other datasets have similar structure. The networks were pre-trained without regularization until the loss no longer improved for eight concurrent epochs, a heuristically chosen number. Due to the restriction of DTD, we constrained biases in all layers to be zero or negative. The trained network is our baseline. This network is fine-tuned by respectively applying L1 regularization, L1+ regularization or a block constraint and training until loss has again no longer improved for eight epochs. We heuristically chose $\lambda = 1.0$ for the regularization rate. The weight parameters of the last layer, to which the structuring penalty is applied, is visualized in Fig. 2.

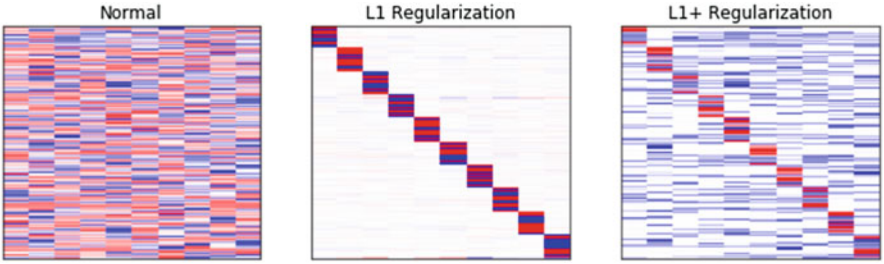


Fig. 2 Visualization of dense layer weights for baseline, L1, and L1+ regularized networks. Positive values are red, negative values are blue

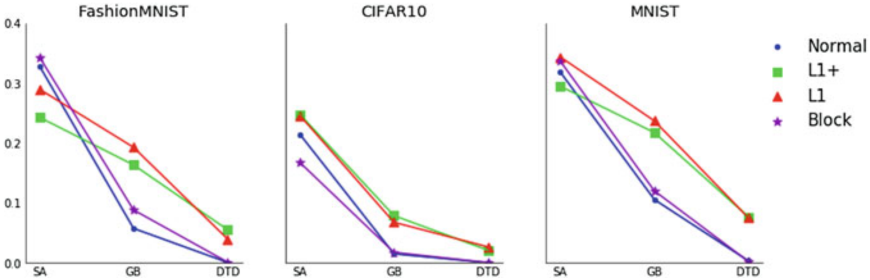


Fig. 3 Explanation separability as measured by the expected cosine distance (ECD), for different models, explanation techniques, and datasets

Denoting by $R_A(\mathbf{x})$ and $R_B(\mathbf{x})$ the heatmaps for the true class and the class with the second highest output, we measure the effectiveness of the architecture at separating classes by the expected cosine distance (ECD):

$$\text{ECD} = \mathbb{E}_{\mathfrak{D}}\left[1 - \frac{\langle R_A(\mathbf{x}), R_B(\mathbf{x})\rangle}{||R_A(\mathbf{x})||_2 \cdot ||R_B(\mathbf{x})||_2}\right], \tag{6}$$

where $\mathbb{E}_{\mathfrak{D}}$ is the expectation over the set of test data points for which the neural networks build evidence for at least two classes. A high ECD reflects a strong ability of the neural network to produce class-specific heatmaps, and accordingly suffer less from the explanation bias.[1]

In Fig. 3, the ECD for regularized and normal networks is shown. We see that structuring the network with L1 regularization consistently helps with the disentanglement of class representations for GB and DTD. It does not have a significant effect for SA. This is likely due to the fact that SA is based on local variations of the prediction function and less dependent on the way the function structures itself in the neural network. The effects were consistently present when we repeated the experiments multiple times with different network configurations.

[1] Other quantitative ways of comparing the interpretability of different models, or different explanation techniques, are given in Bau et al. (2017) and Samek et al. (2017).

Table 1 CNN model accuracy

Structure	MNIST	FashionMNIST	CIFAR10
Unrestricted	99.16%	91.98%	83.02%
Block	99.29%	91.84%	71.75%
L1	99.20%	92.21%	84.20%
L1+	99.25%	92.55%	84.07%

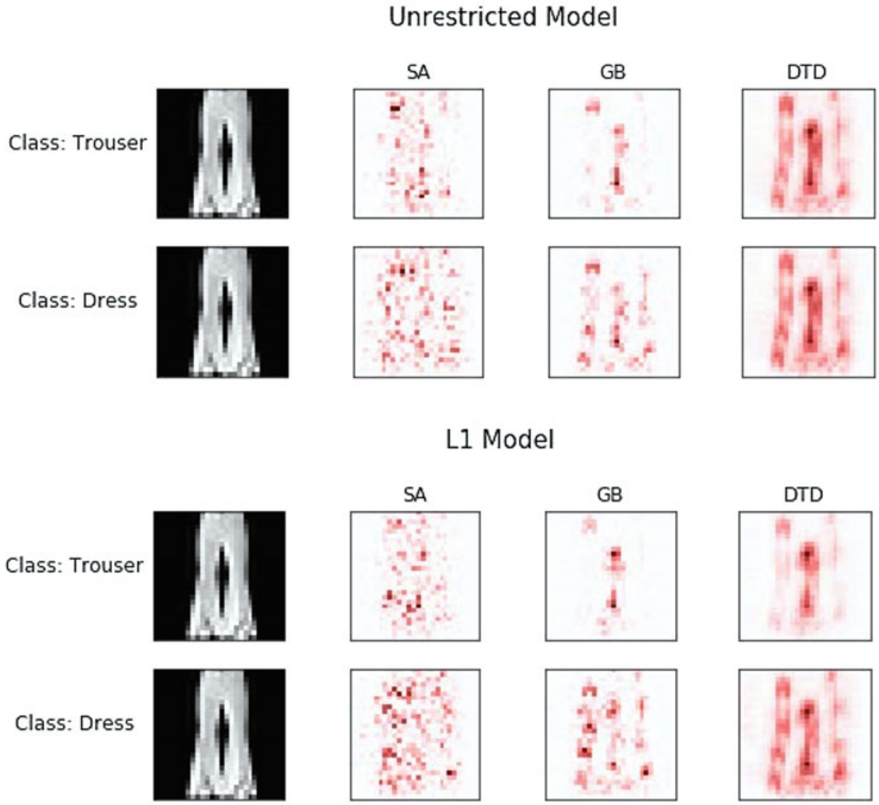


Fig. 4 Heatmaps on FashionMNIST produced by different explanation techniques applied to the basic unrestricted model (top) and the L1/L1+ model with soft block-sparsity (bottom). For the first model, there is nearly no differences between classes. For the second model, the explanations with GB and DTD identify the leg gap as relevant for *trouser* and flared outline for *dress*

As shown in Table 1, the various structuration schemes do not impact the model accuracy with one exception for the block constraint on CIFAR10. They can therefore be considered as viable methods to decrease entanglement of explanations without trading in performance.

We can see in an example in Fig. 4 that the disentanglement of class representations is reflected in sensible differences between heatmaps. The structured model focuses more on the gap between the legs for *trousers* compared to the heatmap for *dress*. The heatmap for *dress* is spread more uniformly over the entire piece of clothing and focuses on the outline, which resembles a dress with a flared bottom. It is visible that the disentanglement of classes also improves the explanations for the correct class, as they now focus more on the relevant feature.

4 Recurrent Neural Networks

Recurrent neural networks (RNNs) are a class of machine learning models that can extract patterns of variable length from sequential data. A longstanding problem with RNN architectures has been the modeling of long-term dependencies. The problem is linked with the difficulty of propagating gradient over many time steps. Architectures, such as LSTM (Hochreiter and Schmidhuber 1997), or hierarchical RNNs (Hihi and Bengio 1995), as well as improved optimization techniques (Sutskever et al. 2013) have been shown to address these difficulties remarkably well so that these techniques can now be applied to complex tasks such as speech recognition or machine translation. Some work has recently focused on explaining recurrent architectures in the context of text analysis (Arras et al. 2017).

In a similar way as for Sect. 3, we will hypothesize that the recurrent structure forms a large salient component of representation and that the classes are predicted based on small variations of that component rather on class-specific features. Thus, explanation techniques that deviate from the prediction function itself might be biased towards that salient component.

To verify this, we consider various RNN architectures with different depths and connectivity. Each of these architectures can be expressed in terms of cells receiving the previous state and the current data, and producing the next state and the prediction. We use ReLU activations for every layer and softmax activation to output the last cell to class probabilities. We consider the following five cell structures (two of them are shown in Fig. 5):

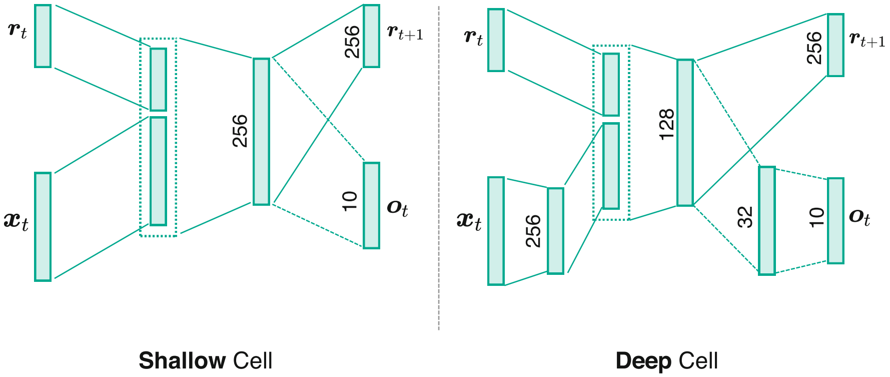


Fig. 5 Examples of RNN cell architectures

Structure 1: Shallow Cell
The shallow cell performs a linear combination of the current state and current data, and computes the next state from it. This is our baseline scenario.

For applicability of deep Taylor decomposition to this architecture, we need an additional propagation rule to redistribute on two different modalities at the same time (hidden state and pixels). Denoting i and j the pixels and ReLU activations respectively forming the two cell modalities and k the hidden layer neuron, the propagation rule is redefined as $R_i = \sum_k (x_i w_{ik} - l_i w_{ik}^+ - h_i w_{ik}^-) \cdot (R_k / z_k)$ and $R_j = \sum_k r_j w_{jk}^+ (R_k / z_k)$, where $z_k = \sum_j r_j w_{jk}^+ + \sum_i x_i w_{ik} - l_i w_{ik}^+ - h_i w_{ik}^-$ is the normalization term.

Structure 2: Deep Cell
The deep cell nonlinearly combines the current state and the current data. This allows to build a data representation that can more meaningfully combine with the hidden state representation. It also makes explanation easier as the two modalities being merged are ReLU activations, and therefore, do not need a special propagation rule for DTD.

Structure 3: Convolutional-Deep Cell
The convolutional-deep (ConvDeep) cell is an extension of the Deep cell in which a sequence of 2 convolution and pooling layers is applied to the input instead of a fully-connected layer. More precisely, we use 24 convolutional filters of size 5×5, followed by *sum* pooling with 2×2 receptive fields. The second convolutional layer has 32 filters of size 3×3, and the setting of the following pooling is the same. We use stride 1 for the two convolution layers, and stride 2 for the pooling layers. This allows to produce well-disentangled features that integrate better with the recurrent states.

Structure 4: R-LSTM Cell
This cell is another variant of the Deep cell. It employs one fully-connected layer with 256 neurons connecting to 75 R-LSTM cells. The R-LSTM cell is a modified version of LSTM whose tanh activations are replaced by ReLU in order to satisfy the constraint of GB and DTD. We treat gate activations in the cell as constants when applying DTD as suggested by Arras et al. (2017), and set their gradients to zero for GB.

Structure 5: ConvR-LSTM Cell
The last cell is an extension of R-LSTM where the first fully-connected layer is replaced by the convolution and pooling layers used in Structure 3.

4.1 Experiments

We construct an artificial problem consisting of three images concatenated horizontally (two of a given class, and one of another class), and where the goal is

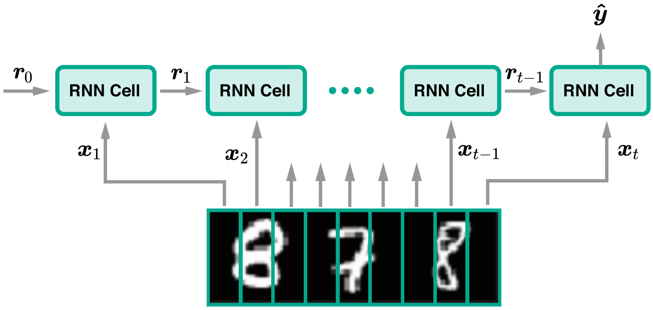


Fig. 6 RNN architecture scanning through a sequence of three digits and predicting the dominating class, here "8"

Table 2 Number of parameters of each RNN structure, and model accuracy

		Accuracy	
Cell architecture	# Parameters	MNIST	FashionMNIST
Shallow	184,330	98.12%	90.00%
Deep	153,578	98.16%	89.81%
ConvDeep	151,802	99.22%	92.87%
R-LSTM	150,570	98.50%	91.35%
ConvR-LSTM	152,125	99.26%	93.33%

to predict the dominating class. We consider MNIST (LeCun and Cortes 2010) or FashionMNIST (Xiao et al. 2017) examples for this experiment. This leads to classification tasks where the input $\mathbf{x}$ is a mosaic of size 28×84, and where the output is a set of 10 possible classes. With this construction, we can easily estimate explainability by measuring which percentage of the explanation falls onto the correct tiles of the mosaic.

The problem above is mapped to the RNN architecture by horizontally splitting $\mathbf{x}$ into non-overlapping segments $\{\mathbf{x}_t \in \mathbb{R}^{28\times 7}\}_{t=1}^{12}$ and sequentially presenting these segments to the RNN. Figure 6 illustrates the setting.

The number of neurons for each layer in each architecture is chosen such that these architectures have a similar number of training parameters. Table 2 summarizes the numbers. All models are trained using the backpropagation through time procedure and using the Adam optimizer (Kingma and Ba 2014). We initialize weights with 2σ-truncated normal distribution with $\mu = 0$ and $\sigma = 1/\sqrt{|\mathbf{a}|}$ where $|\mathbf{a}|$ is the number of neurons from the previous layer as suggested in LeCun et al. (2012). Biases are initialized to zero and constrained to be zero or negative during training. We train for 100 epochs using batch size 50. We apply dropout to every fully-connected layers, except neurons in input and output layers. Dropout probability is set to 0.2.

The learning rate is adjusted for each architecture to achieve good predictive performance. To use an architecture for experiments, we require that accuracy reaches approximately 98% and 90% on MNIST and FashionMNIST respectively.

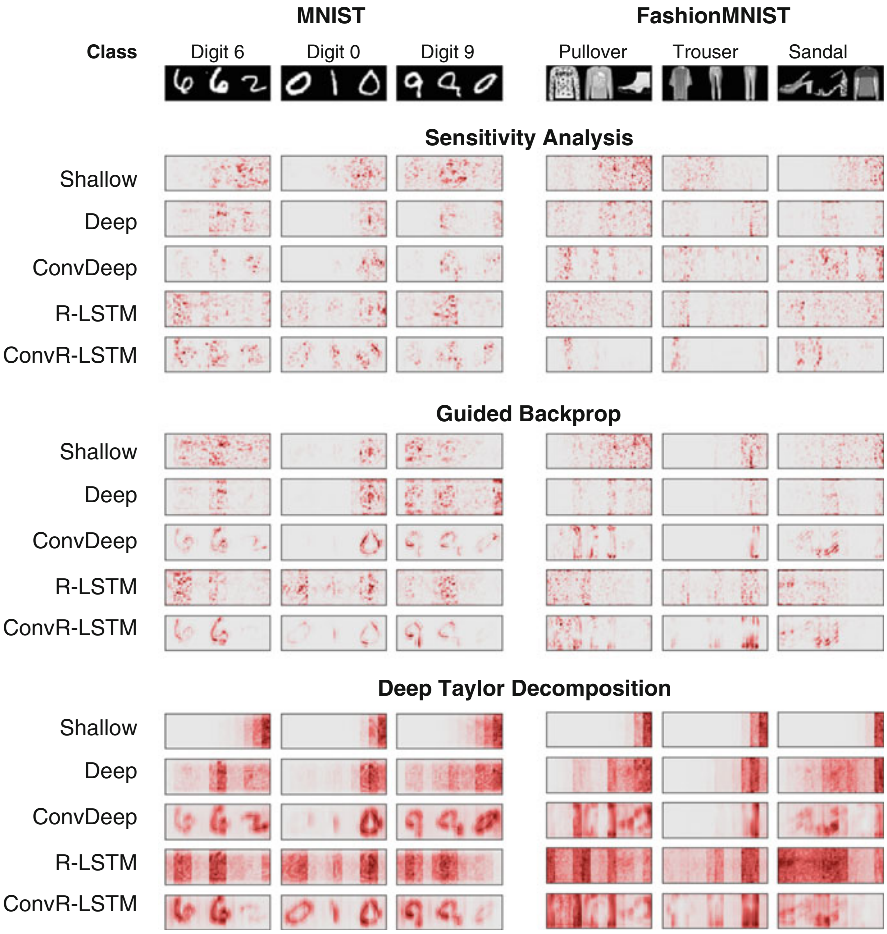


Fig. 7 Heatmaps obtained with each RNN structure, for different explanation techniques and datasets

Lastly, we add one additional input with constant value zero to the softmax layer. This last modification forces the model to build positive evidence for predicting classes rather than relying on building counter-evidence for other classes.

Figure 7 shows relevance heatmaps produced by various methods on the Shallow, Deep, ConvDeep, R-LSTM and ConvR-LSTM architectures. We observe that incorporating structure into the cell leads to a better allocation onto the relevant elements of the sequence. This is particularly noticeable for DTD, where heatmaps of the base model (Shallow) are strongly biased towards a *salient component* constituted of the rightmost pixels, whereas heatmaps for the structured models, especially LSTMs, are more balanced. ConvR-LSTM further improves R-LSTM's heatmaps by providing more resolution at the pixel level. Nevertheless, the presence of features from irrelevant input, such as “1” in Digit 0 example, suggests that cell design can be further improved for the purpose of explanation, beyond the modifications we have proposed here.

In the following, we provide quantitative measures of heatmap quality.[2] By construction, we know that relevance should be assigned to the two dominating items in the sequence (i.e. those that jointly determine the class). The degree to which heatmaps satisfy this property can be quantified by computing the cosine similarity between a binary vector $I(\mathbf{x}) \in \{(1,1,0),(1,0,1),(0,1,1)\}$, indicating what are the two items of the sequence $\mathbf{x}$ having the same class, and a vector of the same dimensions $R(\mathbf{x}) \in \mathbb{R}^3$ containing relevance scores pooled on each item of the sequence. Our metric for quantifying explanation power is the expected cosine similarity:

$$\mathrm{ECS} = \mathbb{E}_{\mathcal{D}}\left[\frac{\langle I(\mathbf{x}), R(\mathbf{x})\rangle}{\|I(\mathbf{x})\|_2 \cdot \|R(\mathbf{x})\|_2}\right], \tag{7}$$

where $\mathbb{E}_{\mathcal{D}}[\cdot]$ computes an average over all sequences in the test set. The higher the ECS the better. Figure 8 shows our ECS metric for various models and explanation techniques. Generally, we can see that more structured cells have higher ECS than the Shallow architecture. In particular, R-LSTM and ConvR-LSTM show significant improvements across all methods. Moreover, the large difference of the cosine similarity between Shallow and Deep architectures also corroborates the strong impact of cell structure on the DTD heatmaps as it was observed in Fig. 7.

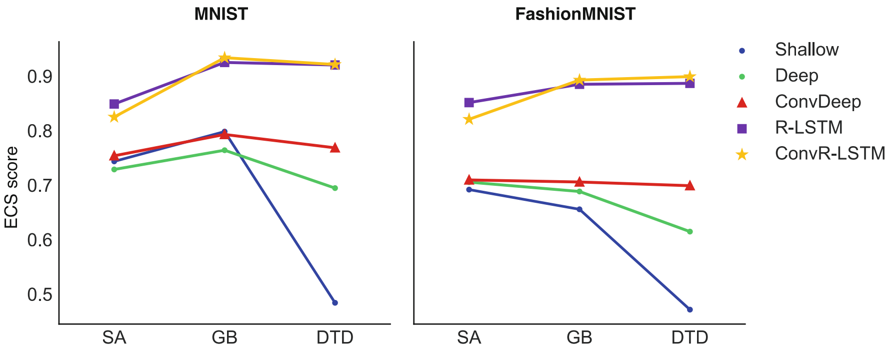


Fig. 8 Explanation quality as measured by the expected cosine similarity (ECS), for different models, explanation techniques, and datasets

[2] See also Bau et al. (2017) and Samek et al. (2017).

5 Conclusion

The success of neural networks at learning functions that accurately predict complex data has fostered the development of techniques that explain how the network decides. While the training objective closely relates to the prediction task, the explanation of these predictions comes as a by-product and little guarantee is offered on their correctness.

In this paper, we have shown that different neural network structures, while offering similar prediction accuracy, can strongly influence the quality of explanations. Both for the baseline CNNs and RNNs, the explanations are biased towards a salient component. This salient component corresponds to general image features for the CNNs or the last time steps for the RNNs.

While the neural network is still able to solve the task based on capturing small variations of that salient component, the explanation technique, which departs from the function to predict, is much more sensitive to it. Therefore, when explanation of the prediction is needed, it is important to pay further attention to the neural network architecture, in particular, by making sure that each class or concept to explain, builds its own features, and that these features are well-disentangled.

Acknowledgements This work was supported by the Brain Korea 21 Plus Program through the National Research Foundation of Korea; the Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government [No. 2017-0-01779]; the Deutsche Forschungsgemeinschaft (DFG) [grant MU 987/17-1]; the German Ministry for Education and Research as Berlin Big Data Center (BBDC) [01IS14013A]; and the Innovation Foundation Denmark through the DABAI project. This publication only reflects the authors views. Funding agencies are not liable for any use that may be made of the information contained herein.

References

Angermueller C, Pärnamaa T, Parts L, Stegle O (2016) Deep learning for computational biology. Molecular Systems Biology 12(7)

Arras L, Montavon G, Müller K, Samek W (2017) Explaining recurrent neural network predictions in sentiment analysis. In: Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@EMNLP 2017, Copenhagen, Denmark, September 8, 2017, pp 159–168

Bach S, Binder A, Montavon G, Klauschen F, Müller KR, Samek W (2015) On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLOS ONE 10(7):e0130,140

Baehrens D, Schroeter T, Harmeling S, Kawanabe M, Hansen K, Müller K (2010) How to explain individual classification decisions. Journal of Machine Learning Research 11:1803–1831

Balduzzi D, Frean M, Leary L, Lewis JP, Ma KW, McWilliams B (2017) The shattered gradients problem: If resnets are the answer, then what is the question? In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017, pp 342–350

Bau D, Zhou B, Khosla A, Oliva A, Torralba A (2017) Network dissection: Quantifying interpretability of deep visual representations. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21–26, 2017, pp 3319–3327

Bengio Y, Simard PY, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Networks 5(2):157–166

Caruana R, Lou Y, Gehrke J, Koch P, Sturm M, Elhadad N (2015) Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10–13, 2015, pp 1721–1730

Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa PP (2011) Natural language processing (almost) from scratch. Journal of Machine Learning Research 12:2493–2537

Gevrey M, Dimopoulos I, Lek S (2003) Review and comparison of methods to study the contribution of variables in artificial neural network models. Ecological Modelling 160(3):249–264

Hihi SE, Bengio Y (1995) Hierarchical recurrent neural networks for long-term dependencies. In: Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, November 27–30, 1995, pp 493–499

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Computation 9(8):1735–1780

Jarrett K, Kavukcuoglu K, Ranzato M, LeCun Y (2009) What is the best multi-stage architecture for object recognition? In: IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009, pp 2146–2153

Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. CoRR abs/1412.6980

Krizhevsky A (2009) Learning Multiple Layers of Features from Tiny Images. Tech. rep., University of Toronto

Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3–6, 2012, Lake Tahoe, Nevada, United States., pp 1106–1114

Landecker W, Thomure MD, Bettencourt LMA, Mitchell M, Kenyon GT, Brumby SP (2013) Interpreting individual classifications of hierarchical networks. In: IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2013, Singapore, 16–19 April, 2013, pp 32–38

LeCun Y (1989) Generalization and network design strategies. In: Pfeifer R, Schreter Z, Fogelman F, Steels L (eds) Connectionism in perspective, Elsevier

LeCun Y, Cortes C (2010) MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/

LeCun Y, Bottou L, Orr GB, Müller KR (2012) Efficient backprop. In: Neural networks: Tricks of the trade, Springer, pp 9–50

Montavon G, Lapuschkin S, Binder A, Samek W, Müller K (2017) Explaining nonlinear classification decisions with deep Taylor decomposition. Pattern Recognition 65:211–222

Montavon G, Samek W, Müller K (2018) Methods for interpreting and understanding deep neural networks. Digital Signal Processing 73:1–15

Montúfar GF, Pascanu R, Cho K, Bengio Y (2014) On the number of linear regions of deep neural networks. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8–13 2014, Montreal, Quebec, Canada, pp 2924–2932

Pascanu R, Mikolov T, Bengio Y (2013) On the difficulty of training recurrent neural networks. In: Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16–21 June 2013, pp 1310–1318

Poulin B, Eisner R, Szafron D, Lu P, Greiner R, Wishart DS, Fyshe A, Pearcy B, Macdonell C, Anvik J (2006) Visual explanation of evidence with additive classifiers. 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, pp 1822–1829

Rasmussen PM, Hansen LK, Madsen KH, Churchill NW, Strother SC (2012) Model sparsity and brain pattern interpretation of classification models in neuroimaging. Pattern Recognition 45(6):2085–2100

Ribeiro MT, Singh S, Guestrin C (2016) “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, San Francisco, CA, USA, August 13–17, 2016, pp 1135–1144

Samek W, Binder A, Montavon G, Lapuschkin S, Müller KR (2017) Evaluating the visualization of what a deep neural network has learned. IEEE transactions on neural networks and learning systems 28(11):2660–2673

Schütt KT, Arbabzadah F, Chmiela S, Müller KR, Tkatchenko A (2017) Quantum-chemical insights from deep tensor neural networks. Nature Communications 8:13,890

Shrikumar A, Greenside P, Kundaje A (2017) Learning important features through propagating activation differences. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017, pp 3145–3153

Simonyan K, Vedaldi A, Zisserman A (2013) Deep inside convolutional networks: Visualising image classification models and saliency maps. CoRR abs/1312.6034

Snyder JC, Rupp M, Hansen K, Müller KR, Burke K (2012) Finding density functionals with machine learning. Physical Review Letters 108(25)

Snyder JC, Rupp M, Müller KR, Burke K (2015) Nonlinear gradient denoising: Finding accurate extrema from inaccurate functional derivatives. International Journal of Quantum Chemistry 115(16):1102–1114

Springenberg JT, Dosovitskiy A, Brox T, Riedmiller MA (2014) Striving for simplicity: The all convolutional net. CoRR abs/1412.6806

Sutskever I, Martens J, Dahl GE, Hinton GE (2013) On the importance of initialization and momentum in deep learning. In: Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16–21 June 2013, pp 1139–1147

Szegedy C, Liu W, Jia Y, Sermanet P, Reed SE, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7–12, 2015, pp 1–9

Xiao H, Rasul K, Vollgraf R (2017) Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. CoRR abs/1708.07747

Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I, pp 818–833

Zhang J, Lin ZL, Brandt J, Shen X, Sclaroff S (2016) Top-down neural attention by excitation backprop. In: Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV, pp 543–559

Zhou B, Khosla A, Lapedriza À, Oliva A, Torralba A (2016) Learning deep features for discriminative localization. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016, pp 2921–2929

Part III
Explainability and Interpretability in Computer Vision

Generating Post-Hoc Rationales of Deep Visual Classification Decisions

Zeynep Akata, Lisa Anne Hendricks, Stephan Alaniz, and Trevor Darrell

Abstract Clearly explaining a rationale for a classification decision to an end-user can be as important as the decision itself. Existing approaches for deep visual recognition are generally opaque and do not output any justification text; contemporary vision-language models can describe image content but fail to take into account class-discriminative image aspects which justify visual predictions. Our model focuses on the discriminating properties of the visible object, jointly predicts a class label, and explains why the predicted label is appropriate for the image. A sampling and reinforcement learning based loss function learns to generate sentences that realize a global sentence property, such as class specificity. Our results on a fine-grained bird species classification dataset show that this model is able to generate explanations which are not only consistent with an image but also more discriminative than descriptions produced by existing captioning methods. In this work, we emphasize the importance of producing an explanation for an observed action, which could be applied to a black-box decision agent, akin to what one human produces when asked to explain the actions of a second human.

Keywords Explainable AI · Rationalizations · Fine-grained classification

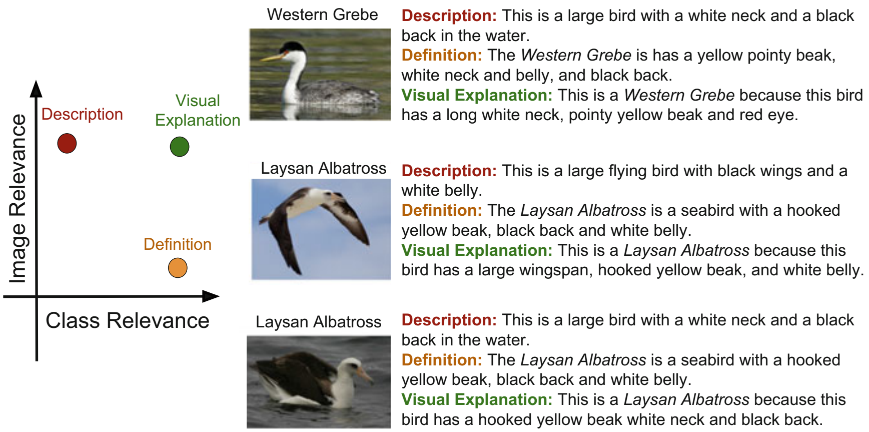
1 Introduction

Explaining why the output of a visual system is compatible with visual evidence is a key component for understanding and interacting with AI systems (Biran and McKeown 2014). Deep image classification frameworks have had tremendous

Z. Akata (✉) · S. Alaniz
AMLAB, University of Amsterdam, Amsterdam, The Netherlands
e-mail: z.akata@uva.nl; s.alaniz@uva.nl

L. A. Hendricks · T. Darrell
EECS, University of California Berkeley, Berkeley, CA, USA
e-mail: lisa_anne@berkeley.edu; trevor@eecs.berkeley.edu

H. J. Escalante et al. (eds.), *Explainable and Interpretable Models in Computer Vision and Machine Learning*, The Springer Series on Challenges in Machine Learning, https://doi.org/10.1007/978-3-319-98131-4_6



Description: This is a large bird with a white neck and a black back in the water.
Definition: The *Western Grebe* is has a yellow pointy beak, white neck and belly, and black back.
Visual Explanation: This is a *Western Grebe* because this bird has a long white neck, pointy yellow beak and red eye.

Description: This is a large flying bird with black wings and a white belly.
Definition: The *Laysan Albatross* is a seabird with a hooked yellow beak, black back and white belly.
Visual Explanation: This is a *Laysan Albatross* because this bird has a large wingspan, hooked yellow beak, and white belly.

Description: This is a large bird with a white neck and a black back in the water.
Definition: The *Laysan Albatross* is a seabird with a hooked yellow beak, black back and white belly.
Visual Explanation: This is a *Laysan Albatross* because this bird has a hooked yellow beak white neck and black back.

Fig. 1 Our proposed model generates *explanations* that are both image relevant and class relevant. In contrast, *descriptions* are image relevant, but not necessarily class relevant, and *definitions* are class relevant but not necessarily image relevant

success in visual recognition (Krizhevsky et al. 2012; Gao et al. 2016; Donahue et al. 2013), but their outputs can be unsatisfactory if the model cannot provide a consistent justification of why it made a certain prediction. In contrast, systems which can justify why a prediction is consistent with visual elements to a user are more likely to be trusted (Teach and Shortliffe 1981). Explanations of visual systems could also aid in understanding network mistakes and provide feedback to improve classifiers.

We argue that visual explanations must satisfy two criteria: they must be *class discriminative* and they must *accurately describe* a specific image instance. As shown in Fig. 1, explanations are distinct from *descriptions*, which are based only on visual information, and *definitions*, which are based only on class information. Indeed, explanations detail why a certain category is appropriate for a given image while only mentioning image relevant properties. For example, consider a classification system that predicts a certain image belongs to the class "western grebe" (Fig. 1, top). A standard captioning system might provide a description such as "This is a large bird with a white neck and black back in the water." However, as this description does not mention *class discriminative* features, it could also be applied to a "laysan albatross" (Fig. 1, bottom). In contrast, we propose to provide *explanations*, such as "This is a western grebe because this bird has a long white neck, pointy yellow beak, and a red eye." The explanation includes the "red eye" property, which is important for distinguishing between "western grebe" and "laysan albatross". As such, our system explains *why* the predicted category is the most appropriate for the image.

In this work, we differentiate between two types of explanations, Type 1, i.e. introspections and Type 2, i.e. post-hoc rationalizations:

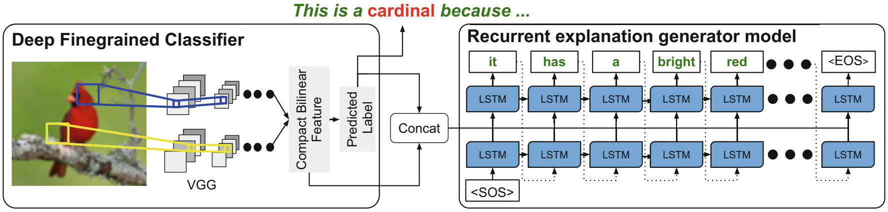

This is a cardinal because ...
Deep Finegrained Classifier
VGG
Compact Bilinear Feature
Predicted Label
Concat
Recurrent explanation generator model
it
has
a
bright
red
<EOS>
LSTM
<SOS>

Fig. 2 Our joint classification and explanation model, aka GVE. We extract visual features using a fine-grained classifier before sentence generation and, unlike other sentence generation models, condition sentence generation on the predicted class label. A novel discriminative loss encourages generated sentences to include class specific attributes

Type 1 *Introspections* focus on explanations as a network transparency process that occurs while making a decision. Introspective explanations argue how a model determines its final output, e.g. "This is a western grebe because filter 2 has a high activation…". The decision agent needs to be fully transparent, hence the decision maker should allow interventions.

Type 2 *Post-Hoc Rationales* explain the decision of another deep network, i.e. the decision maker. They detail by which means the visual evidence is compatible with a decision, e.g. "This is a western grebe because it has red eyes…". In other words, they find patterns that relate the input data and the decision after a black-box agent produces a decision.

We concentrate on Type 2 explanations, i.e. post-hoc rationales, because they may be more useful to the end users who typically do not have knowledge of modern computer vision systems (Biran and McKeown 2014).

Figure 2 outlines our approach. In contrast to models that generate descriptions, we condition our explanation generator on the image and the predicted class label. We also use features extracted from a fine-grained recognition pipeline (Gao et al. 2016). Like many contemporary image captioning models (Vinyals et al. 2015; Donahue et al. 2015; Karpathy and Li 2015; Xu et al. 2015; Kiros et al. 2014), we use an LSTM (Hochreiter and Schmidhuber 1997) to generate word sequences. On the other hand, we design a novel loss function which encourages generated sentences to include class discriminative information, i.e. to be class specific. One challenge is that class specificity is a global sentence property: e.g. a sentence "This is an all black bird with a bright red eye" is class specific to a "bronzed cowbird", whereas words and phrases in this sentence, such as "black" or "red eye" are less class specific on their own. Our final output is a sampled sentence, so we backpropagate the discriminative loss through the sentence sampling mechanism via REINFORCE (Williams 1992), i.e. a technique from the reinforcement learning literature.

In Sect. 2, we review prior works related to ours. To the best of our knowledge, ours is the first framework to produce deep visual explanations using natural language justifications. In Sect. 3 we detail GVE, i.e. our novel joint vision and language explanation model, that combines classification and captioning by incorporating a loss function that operates over sampled sentences. In Sect. 5 we show that this formulation is able to focus generated text to be more discriminative and that our model produces better explanations than a description baseline. Our results also confirm including a discriminative class label loss improves accuracy with respect to traditional sentence generation metrics. Finally, we extend our framework to generating counterfactual explanations, i.e. sentences that explain why an image belongs to a certain class and not to another one. An initial version of our work has been published in Hendricks et al. (2016). In this book chapter, in addition to the material from Hendricks et al. (2016), we also include a discussion on Type1 and Type 2 explanations, further clarifications and more experimental analysis of our model.

2 Related Work

We present related works in explanation, visual description, fine-grained classification and reinforcement learning in computer vision topics.

Automatic Reasoning and Explanation Automatic reasoning and explanation has a long and rich history within the artificial intelligence community (Biran and McKeown 2014; Shortliffe and Buchanan 1975; Lane et al. 2005; Core et al. 2006; Van Lent et al. 2004; Lomas et al. 2012; Lacave and Díez 2002; Johnson 1994). Explanation systems span a variety of applications including explaining medical diagnosis (Shortliffe and Buchanan 1975), simulator actions (Lane et al. 2005; Core et al. 2006; Van Lent et al. 2004; Johnson 1994), and robot movements (Lomas et al. 2012). Many of these systems are rule-based (Shortliffe and Buchanan 1975) or solely reliant on filling in a predetermined template (Van Lent et al. 2004). Methods such as (Shortliffe and Buchanan 1975) require expert-level explanations and decision processes. As expert explanations or decision processes are not available during training, our model learns purely from visual features and fine-grained visual descriptions to fulfill our two proposed visual explanation criteria. In contrast to systems like (Shortliffe and Buchanan 1975; Lane et al. 2005; Core et al. 2006; Van Lent et al. 2004; Lomas et al. 2012; Lacave and Díez 2002) which aim to explain the underlying mechanism behind a decision, Biran and McKeown (2014) concentrate on why a prediction is justifiable to a user. Such systems are advantageous because they do not rely on user familiarity with the design of an intelligent system in order to provide useful information.

Many vision methods focus on discovering visual features which can help "explain" an image classification decision by discovering exemplar visual patches (Berg and Belhumeur 2013; Doersch et al. 2012) or analyzing what individual neurons might "represent" in a deep network (Zeiler and Fergus 2014; Escorcia et al. 2015; Zhou et al. 2015). "Visual" explanations which produce heat/attention maps over important image regions, are also popular (Fong and Vedaldi 2017; Selvaraju et al. 2016; Zeiler and Fergus 2014; Zintgraf et al. 2017). Importantly, these models do not link explanations to natural language expressions. We believe that the methods are complementary to our proposed system. In fact, such explanations could be used as additional inputs to our model to produce better explanations. For example, since the initial release of this work, Park et al. (2018) proposed one such model which produces both visual and textual explanations.

Visual Description Early image description methods rely on detecting visual concepts (e.g., subject, verb, and object) before generating a sentence with either a simple language model or sentence template (Kulkarni et al. 2011; Guadarrama et al. 2013). Recent deep models (Vinyals et al. 2015; Donahue et al. 2015; Karpathy and Li 2015; Xu et al. 2015; Kiros et al. 2014; Fang et al. 2015; Mao et al. 2014) outperform such systems and produce fluent, accurate descriptions. Though most description models condition sentence generation only on image features, Jia et al. (2015) condition generation on auxiliary information, such as words used to describe a similar image in the train set. However, Jia et al. (2015) does not condition sentence generation on category labels.

LSTM sentence generation models are generally trained with a cross-entropy loss between the probability distribution of predicted and ground truth words (Vinyals et al. 2015; Donahue et al. 2015; Karpathy and Li 2015; Xu et al. 2015; Mao et al. 2014). Frequently, however, the cross-entropy loss does not directly optimize for properties desirable at test time. Mao et al. (2016) propose a training scheme for generating unambiguous region descriptions which maximizes the probability of a region description while minimizing the probability of other region descriptions. In this work, we propose a novel loss function for sentence generation which allows us to specify a global constraint on generated sentences.

Fine-Grained Classification Object classification, particularly fine-grained classification, is an attractive setting for explanation systems because describing image content does not suffice as an explanation. Explanation models must focus on aspects that are both class-specific and depicted in the image.

Most fine-grained zero-shot and few-shot image classification systems use attributes (Lampert et al. 2013) as auxiliary information. Attributes discretize a high dimensional feature space into simple and readily interpretable decision statements that can act as an explanation. However, attributes have several disadvantages. They require experts for annotation which is costly and results in attributes which are hard for non-experts to interpret (e.g., "spatulate bill shape"). Attributes are not scalable as the list of attributes needs to be revised to ensure discriminativeness for new classes. Finally, attributes do not provide a natural language explanation like the user expects. We therefore use natural language descriptions (Reed et al. 2016a) which achieve superior performance on zero-shot learning compared to attributes and also are shown to be useful for text to image generation (Reed et al. 2016b).

Reinforcement Learning in Computer Vision Vision models which incorporate algorithms from reinforcement learning, specifically how to backpropagate through a sampling mechanism, have recently been applied to visual question answering (Andreas et al. 2016) and activity detection (Yeung et al. 2016). Additionally, Xu et al. (2015) use a sampling mechanism to attend to specific image regions for caption generation, but use the standard cross-entropy loss during training. Concurrently to our work, Ranzato et al. (2015) proposed employing reinforcement learning to directly optimize sentence metrics. Since then, Liu et al. (2016), Ren et al. (2017), Rennie et al. (2016) have proposed other variants for sentence generation using reinforcement learning.

3 Generating Visual Explanations (GVE) Model

Our framework, referred to as Generating Visual Explanations (GVE), is illustrated in Fig. 3. It starts with a deep fine-grained classifier which takes an image as an input and outputs class discriminative image features as well as a classification decision, i.e. image category. The image category then gets concatenated with the image feature which is used to condition a two layer LSTM model which learns to generate class-consistent and fluent sentences that describe image content. The LSTM which constitutes our GVE model aims to produce an explanation which describes visual content present in a specific image instance, enforced by the *relevance loss* (Fig. 3, bottom right) while containing appropriate information to explain why the image belongs to a specific category, enforced by the *discriminative loss* (Fig. 3, top right). Below we detail these two loss functions that are trained jointly.

3.1 Relevance Loss

Image relevance can be accomplished by training a visual description model. Our model is based on LRCN (Donahue et al. 2015), which consists of a convolutional

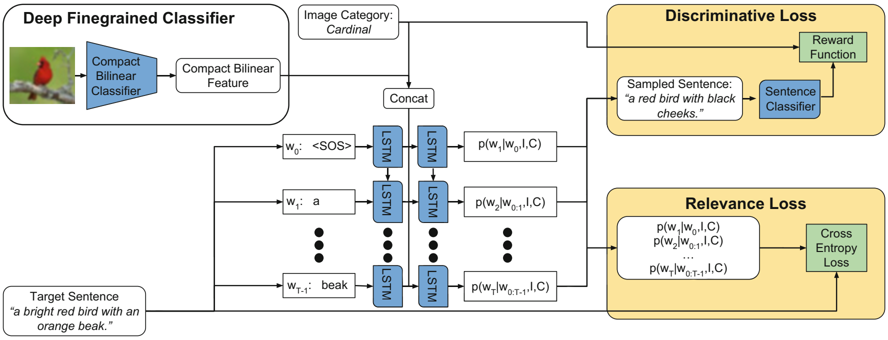


Fig. 3 Training our Generating Visual Explanations (GVE) model. Our GVE model differs from other caption models because it (1) includes the object category as an additional input and (2) incorporates a reinforcement learning based discriminative loss

network, which extracts high level visual features, and two stacked recurrent networks (specifically LSTMs), which generate descriptions conditioned on visual features. During inference, the first LSTM receives the previously generated word w_{t-1} as input and produces an output l_t. The second LSTM, receives the output of the first LSTM l_t and an image feature f and produces a probability distribution $p(w_t)$ over the next word. The word w_t is generated by sampling from the distribution $p(w_t)$. Generation continues until an "end-of-sentence" token is generated.

We propose two modifications to the LRCN framework to increase the image relevance of generated sequences (Fig. 3, top left). First, category predictions are used as an additional input to the second LSTM in the sentence generation model. Intuitively, category information can help inform the caption generation model which words and attributes are more likely to occur in a description. For example, category level information can help the model decide if a red eye or red eyebrow is more likely for a given class. We experimented with a few methods to represent class labels, and found that training a language model, e.g, an LSTM, to generate word sequences conditioned on images, then using the average hidden state of the LSTM across all sequences for all classes in the train set as a vectorial representation of a class works best. Second, we use rich category specific image features (Gao et al. 2016) to generate relevant explanations.

Each training instance consists of an image, category label, and a ground truth sentence. During training, the model receives the ground truth word w_t for each time step $t \in T$. We define the relevance loss for a specific image (I) and caption (C) as:

$$L_R(I, C) = -\frac{1}{N} \sum_{n=0}^{N-1} \sum_{t=0}^{T-1} \log p(w_{t+1} | w_{0:t}, I, C) \tag{1}$$

where w_t is a ground truth word and N is the batch size. By training the model to predict each word in a ground truth sentence, the model produces sentences which reflect the image content. However, this loss does not explicitly encourage generated sentences to discuss discerning visual properties. In order to generate sentences which are both image relevant and category specific, we include a discriminative loss to focus sentence generation on discriminative visual properties of the object.

3.2 Discriminative Loss

Our discriminative loss is based on a reinforcement learning paradigm for learning with layers which require sampling intermediate activations of a network. In our formulation, we first sample a sentence and then use the sampled sentence to compute a discriminative loss. By sampling the sentence before computing the loss, we ensure that sentences sampled from our model are more likely to be class specific. Our reinforcement based loss enables us to backpropagate through the sentence sampling mechanism.

We minimize the following overall loss function with respect to the network weights θ:

$$L_R(I, C) - \lambda \mathbb{E}_{\tilde{w} \sim p(w|I,C)} \left[R_D(\tilde{w}) \right] \tag{2}$$

which is a linear combination of the relevance loss L_R and the expectation of the negative discriminator reward $-R_D(\tilde{w})$ over sentences $\tilde{w} \sim p(w|I, C)$, where $p(w|I, C)$ is the model's estimated conditional distribution over sentences w given the image I and category C. Since $\mathbb{E}_{\tilde{w} \sim p(w|I,C)} \left[R_D(\tilde{w}) \right]$ is intractable, we estimate it at training time using Monte Carlo sampling of sentences from the categorical distribution given by the model's softmax output at each timestep. The sampling operation for the categorical distribution is non-smooth in the distribution's parameters $\{p_i\}$ as it is a discrete distribution. Therefore, $\nabla_\theta R_D(\tilde{w})$ for a given sample $\tilde{w}$ with respect to the weights θ is undefined.

Following the REINFORCE (Williams 1992) algorithm, we make use of the following equivalence property of the expected reward gradient:

$$\nabla_\theta \mathbb{E}_{\tilde{w} \sim p(w|I,C)} \left[R_D(\tilde{w}) \right] = \mathbb{E}_{\tilde{w} \sim p(w|I,C)} \left[R_D(\tilde{w}) \nabla_\theta \log p(\tilde{w}) \right] \tag{3}$$

In this reformulation, the gradient $\nabla_\theta \log p(\tilde{w})$ is well-defined: $\log p(\tilde{w})$ is the log-likelihood of the sampled sentence $\tilde{w}$, just as L_R is the log-likelihood of the ground truth sentence. However, the sampled gradient term is weighted by the reward $R_D(\tilde{w})$, pushing the weights to increase the likelihood assigned to the most highly rewarded (and hence most discriminative) descriptions. Therefore, the final gradient we compute to update the weights θ, given a description $\tilde{w}$ sampled from the model's softmax distribution, is:

$$\nabla_\theta L_R - \lambda R_D(\tilde{w}) \nabla_\theta \log p(\tilde{w}). \tag{4}$$

$R_D(\tilde{w})$ should be high when sampled sentences are discriminative. We define our reward simply as $R_D(\tilde{w}) = p(C|\tilde{w})$, or the probability of the ground truth category C given only the generated sentence $\tilde{w}$. By placing the discriminative loss after the sampled sentence, the sentence acts as an information bottleneck. For the model to produce an output with a large reward, the generated sentence must include enough information to classify the original image properly.

To define a reward function $R_D(\tilde{w})$, we train a single layer LSTM model which takes words as input and predicts a probability distribution over CUB classes. We train this model using ground truth sentences and report that our model assigns the highest probability to the correct class of unseen validation set sentences 22% of the time. Chance level is 0.5% as there are 200 classes in total. This number is possibly low because descriptions in the dataset do not necessarily contain discriminative properties (e.g., "This is a white bird with grey wings." is a valid description but can apply to multiple bird species). Nonetheless, we find that this reward function provides enough information to train our GVE model. Outside text sources (e.g., field guides) could be useful when designing a reward function. However, incorporating outside text can be challenging as this requires aligning our image annotation vocabulary to field-guide vocabulary. When training the GVE model, we do not update weights in the reward function.

4 Experimental Setup

In this section, we first detail our experimental setup and then present the qualitative and quantitative results that our model achieves on several varieties of the fine-grained visual explanation task.

Dataset We employ the Caltech UCSD Birds 200-2011 (CUB) dataset (Wah et al. 2011) which contains 200 classes of bird species and 11,788 images in total. Recently, Reed et al. (2016a) collected five sentences for each of the images which do not only describe the content of the image, e.g., “This is a bird”, but also give a detailed description of the bird, e.g., “with red feathers and has a black face patch”. Unlike other image-sentence datasets, every image in the CUB dataset belongs to a class, and therefore sentences as well as images are associated with a single label. This property makes this dataset unique for the visual explanation task, where our aim is to generate sentences that are both discriminative and class-specific.

Though sentences collected in Reed et al. (2016a) were not originally collected for the visual explanation task, we observe that sentences include detailed and fine-grained category specific information. When ranking human annotations by our learned reward function, we find that sentences with high reward (and thus more discriminative sentences) include rich discriminative details. For example, the sentence “…mostly black all over its body with a small red and yellow portion in its wing” has a score of 0.99 for “red winged blackbird” and includes details specific to this bird variety, such as “red and yellow portion in its wing”. As ground truth annotations are descriptions as opposed to explanations, not all annotations are guaranteed to include discriminative information. For example, though the “bronzed-cowbird” has striking red eyes, not all humans mention this discriminative feature. To generate satisfactory explanations, our model must learn which features are discriminative from descriptions and incorporate discriminative properties into generated explanations.

Implementation For image features, we extract 8192 dimensional features from the penultimate layer of the compact bilinear fine-grained classification model (Gao et al. 2016) which is based on a VGG model that has been pre-trained on the CUB dataset and achieves an accuracy of 84%. This fine-grained classifier combines the VGG16 network (Simonyan and Zisserman 2014) with the compact bilinear layer proposed in Gao et al. (2016). We use one-hot vectors to represent input words at each time step. One-hot vectors are embedded into a 1000 dimensional space then input into an LSTM with 1000 hidden units. We train our models using

Caffe (Jia et al. 2014), and determine model hyperparameters using the standard CUB validation set before evaluating on the test set. Before co-training with the relevance and discriminative loss, we find it is important to pre-train models with only the relevance loss. All models are trained using stochastic gradient descent, and the number of training iterations is determined by performance on the CUB validation set. When training with only the relevance loss, we use a learning rate of 0.01 which is decreased every 2000 iterations with a decay of 0.5. After pre-training models with the relevance loss, we train with both the relevance and discriminative losses using a learning rate of 0.001 which is also decreased every 2000 iterations with a decay of 0.5. All reported results are on the standard CUB test set.

Baseline and Ablation Models We propose two baseline models: a *description* model and a *definition* model. The description baseline generates sentences conditioned only on images and is equivalent to LRCN (Donahue et al. 2015) except we use image features from a fine-grained classifier (Gao et al. 2016). The definition baseline generates sentences using only an image label as input. Consequently, this model outputs the same sentence for every image of the same class. Our model is both more image and class relevant than either of these baselines and thus superior for the explanation task.

Our GVE model differs from description models in two key ways. First, in addition to an image, generated sentences are conditioned on class predictions. Second, explanations are trained with a discriminative loss which enforces that generated sentences contain class specific information (see Eq. (2)). To demonstrate that both class information and the discriminative loss are important, we compare our GVE model to an *GVE-image* model which is not trained with the discriminative loss, and to an *GVE-class* model which is not conditioned on the predicted class.

Evaluation Metrics To evaluate our explanation model, we use automatic metrics and two human evaluations. Our automatic metrics rely on the common sentence evaluation metrics, METEOR (Banerjee and Lavie 2005) and CIDEr (Vedantam et al. 2015), and are used to evaluate the quality of our explanatory text. METEOR is computed by matching words in generated and reference sentences, but unlike other common metrics such as BLEU (Papineni et al. 2002), it uses WordNet (Miller et al. 1990) to also match synonyms. CIDEr measures the similarity of a generated sentence to a reference sentence by counting common n-grams which are TF-IDF weighted. Consequently, CIDEr rewards sentences for correctly including n-grams which are uncommon in the dataset.

A generated sentence is *image relevant* if it mentions concepts which are mentioned in ground truth reference sentences for the image. Thus, to measure image relevance we simply report METEOR and CIDEr scores, with more relevant sentences producing higher METEOR and CIDEr scores.

Measuring *class relevance* is considerably more difficult. We could use the reward function used to train our discriminative loss, but this is an unfair metric because some models were trained to directly increase the reward. Instead, we measure class relevance by considering how similar generated sentences for a class are to ground truth sentences for that class. Sentences which describe a certain bird class, e.g., “cardinal”, should contain similar words and phrases to ground truth “cardinal” sentences, but not ground truth “black bird” sentences. We compute CIDEr scores for images from each bird class, but instead of using ground truth image descriptions as reference sentences, we pool all reference sentences which correspond to a particular class. We call this metric the *class similarity* metric.

Though class relevant sentences should have high class similarity scores, a model could achieve a better class similarity score by producing better overall sentences (e.g., better grammar) without producing more class relevant descriptions. To further demonstrate that our sentences are class relevant, we compute a *class rank* metric. Intuitively, class similarity scores computed for generated sentences about *cardinals* should be higher when compared to *cardinal* reference sentences than when compared to reference sentences from other classes. Consequently, more class relevant models should yield higher rank for ground truth classes. We compute the class similarity for each generated sentence with respect to each bird category and rank bird categories by class similarity. We report the mean rank of the ground truth class as the *class rank* metric. We emphasize the CIDEr metric because of the TF-IDF weighting over n-grams. If a bird has a unique feature, such as “red eyes”, generated sentences which mention this attribute should be rewarded more than sentences which just mention attributes common across all bird classes. We apply our metrics to images for which we predict the correct label as it is unclear if the best explanatory text should be more similar to the correct class or the predicted class. However, the same trends hold if we apply our metrics to all generated sentences.

5 Results

We demonstrate that our model generates superior visual explanations and produces image and class relevant text. Additionally, generating visual explanations results in higher quality sentences based on common sentence generation metrics.

5.1 Quantitative Results

Here, we first measure image and class relevance of our explanations through automatic metrics and also user studies.

Measuring Image Relevance Table 1, columns 2 and 3, record METEOR and CIDEr scores for our generated sentences. Importantly, our GVE model produces sentences with higher METEOR and CIDEr scores than our baselines. The GVE also outperforms the GVE-image and GVE-class model suggesting that both label conditioning and the discriminative loss are key to producing better sentences. Furthermore, METEOR and CIDEr are substantially higher when including a discriminative loss during training (compare rows 2 and 4 and rows 3 and 5)

Table 1 Comparing our explanation model, i.e. GVE, with the definition and description baselines, as well as the GVE-image and GVE-class models ablated from ours

	Image relevance		Class relevance		Best explanation
Model	METEOR	CIDEr	Similarity	Rank (1–200)	Bird expert rank (1–5)
Definition	27.9	43.8	42.60	15.82	2.92
Description	27.7	42.0	35.30	24.43	3.11
GVE-image	28.1	44.7	40.86	17.69	2.97
GVE-class	28.8	51.9	43.61	19.80	3.22
GVE	29.2	56.7	52.25	13.12	2.78

Our GVE explanations are image relevant, as measured by METEOR and CIDEr scores (higher is better). They are also class relevant, as measured by class similarity metric (higher is better) and class rank metric (lower is better) (see Sect. 4 for details). Finally, our GVE explanations are ranked better by experienced bird watchers

demonstrating that including this additional loss leads to better generated sentences. Moreover, the baseline definition model produces more image relevant sentences than the baseline description model suggesting that category information is important for fine-grained description. On the other hand, our GVE-image results, i.e. model not trained with a discriminative loss, are better than both the definition and description results showing that the image and label contain complementary information.

Measuring Class Relevance Table 1, columns 4 and 5, report class similarity and class rank metrics (see Sect. 4 for details). Our GVE model produces a higher class similarity score than other models by a substantial margin. The class rank for our GVE model is also lower than for any other model suggesting that sentences generated by our GVE model more closely resemble the correct class than other classes in the dataset. According to our ranking metric, sentences must include enough information to differentiate between very similar bird classes without looking at an image, and our results clearly show that our GVE model performs best at this difficult task. Observing the reward assigned to sentences produced by different models exhibits the same general trend: the reward function assigns the highest probability to the ground truth class for 59.13% of the sentences generated by the GVE model, in contrast to 22.32% for the description model.

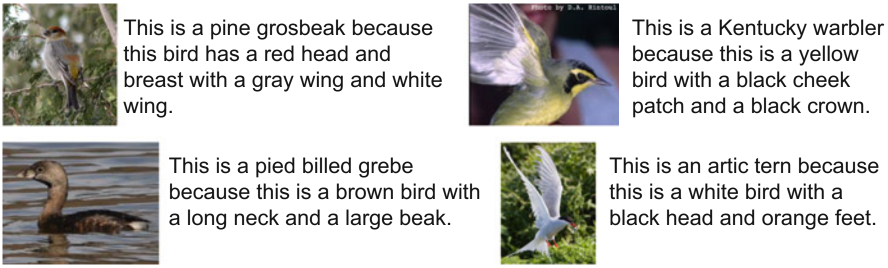
User Studies The ultimate goal of our explanation system is to provide useful information about an unknown object to a user. We therefore also consulted experienced bird watchers to rate our GVE explanations against our baseline and ablation models. Consulting experienced bird watchers is important because some sentences may provide correct, but non-discriminative properties, which an average person may not be able to properly identify. For example, *This is a bronzed cowbird because this bird is nearly all black with a short pointy bill.* is correct, but is a poor explanation as it does not mention unique attributes of a *bronzed cowbird* such as *red eye*. Two experienced bird watchers evaluated 91 randomly selected images and answered which sentence provided the best explanation for the bird class (Table 1, column 6). Our GVE model has the best mean rank (lower is better), followed by the definition model. This trend resembles the trend seen when evaluating class relevance.

We also demonstrate that explanations are more effective than descriptions at helping humans identify different bird species. We ask five Amazon Mechanical Turk workers to choose between two images given a generated description and a GVE explanation. We evaluate 200 images (one for each bird category) and find that our GVE explanations are more helpful to humans. When provided with the sentence that GVE model generates, the correct image is chosen (with an image considered to be chosen correctly if 4 out of 5 workers select the correct image) 56% of the time, whereas when provided with a sentence that the description model generates, the correct image is chosen 52% of the time.

5.2 Qualitative Results

Figure 4 shows sample GVE explanations which first declare a predicted class label ("This is a *Kentucky Warbler* because") followed by the explanatory text produced by the model described in Sect. 3. Qualitatively, our GVE model performs quite well. Note that our model accurately describes fine detail such as *black cheek patch* for *Kentucky Warbler* and *long neck* for *Pied Billed Grebe*.

Comparing Explanations, Baselines, and Ablations Figure 5 compares sentences generated by our GVE, GVE-image, GVE-class, as well as the baseline definition and description models. Each model produces reasonable sentences, however, we expect our GVE model to produce sentences which discuss class relevant properties. For many images, the GVE model uniquely mentions some relevant properties. In Fig. 5, row 1, the GVE model specifies that the *Bronzed Cowbird* has *red eyes* which is rarer than properties mentioned correctly by the definition and description models (*black*, *pointy bill*). For *White Necked Raven* (Fig. 5 row 3), the GVE model identifies the *white nape*, which is a unique attribute of that bird. The GVE sentences



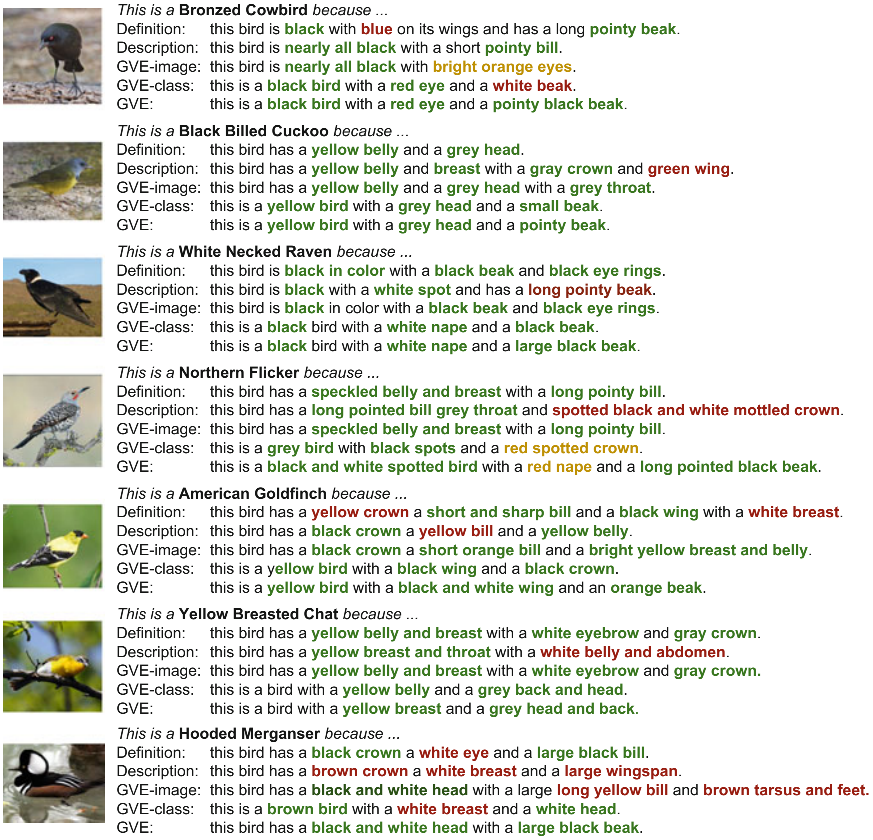
This is a pine grosbeak because this bird has a red head and breast with a gray wing and white wing.

This is a Kentucky warbler because this is a yellow bird with a black cheek patch and a black crown.

This is a pied billed grebe because this is a brown bird with a long neck and a large beak.

This is an artic tern because this is a white bird with a black head and orange feet.

Fig. 4 Visual explanations generated by our system. Our GVE model produces image relevant sentences that also discuss class discriminative attributes



This is a **Bronzed Cowbird** *because ...*
Definition: this bird is **black** with **blue** on its wings and has a long **pointy beak**.
Description: this bird is **nearly all black** with a short **pointy bill**.
GVE-image: this bird is **nearly all black** with **bright orange eyes**.
GVE-class: this is a **black bird** with a **red eye** and a **white beak**.
GVE: this is a **black bird** with a **red eye** and a **pointy black beak**.

This is a **Black Billed Cuckoo** *because ...*
Definition: this bird has a **yellow belly** and a **grey head**.
Description: this bird has a **yellow belly** and **breast** with a **gray crown** and **green wing**.
GVE-image: this bird has a **yellow belly** and a **grey head** with a **grey throat**.
GVE-class: this is a **yellow bird** with a **grey head** and a **small beak**.
GVE: this is a **yellow bird** with a **grey head** and a **pointy beak**.

This is a **White Necked Raven** *because ...*
Definition: this bird is **black in color** with a **black beak** and **black eye rings**.
Description: this bird is **black** with a **white spot** and has a **long pointy beak**.
GVE-image: this bird is **black** in color with a **black beak** and **black eye rings**.
GVE-class: this is a **black** bird with a **white nape** and a **black beak**.
GVE: this is a **black** bird with a **white nape** and a **large black beak**.

This is a **Northern Flicker** *because ...*
Definition: this bird has a **speckled belly and breast** with a **long pointy bill**.
Description: this bird has a **long pointed bill grey throat** and **spotted black and white mottled crown**.
GVE-image: this bird has a **speckled belly and breast** with a **long pointy bill**.
GVE-class: this is a **grey bird** with **black spots** and a **red spotted crown**.
GVE: this is a **black and white spotted bird** with a **red nape** and a **long pointed black beak**.

This is a **American Goldfinch** *because ...*
Definition: this bird has a **yellow crown** a **short and sharp bill** and a **black wing** with a **white breast**.
Description: this bird has a **black crown** a **yellow bill** and a **yellow belly**.
GVE-image: this bird has a **black crown** a **short orange bill** and a **bright yellow breast and belly**.
GVE-class: this is a **yellow bird** with a **black wing** and a **black crown**.
GVE: this is a **yellow bird** with a **black and white wing** and an **orange beak**.

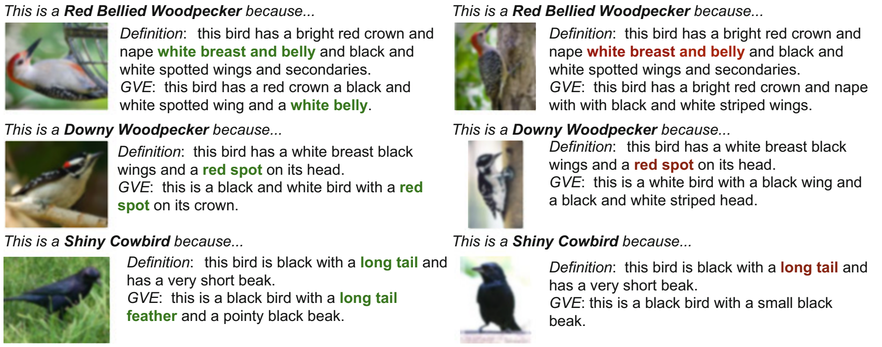
This is a **Yellow Breasted Chat** *because ...*
Definition: this bird has a **yellow belly and breast** with a **white eyebrow** and **gray crown**.
Description: this bird has a **yellow breast and throat** with a **white belly and abdomen**.
GVE-image: this bird has a **yellow belly and breast** with a **white eyebrow** and **gray crown.**
GVE-class: this is a bird with a **yellow belly** and a **grey back and head**.
GVE: this is a bird with a **yellow breast** and a **grey head and back**.

This is a **Hooded Merganser** *because ...*
Definition: this bird has a **black crown** a **white eye** and a **large black bill**.
Description: this bird has a **brown crown** a **white breast** and a **large wingspan**.
GVE-image: this bird has a **black and white head** with a large **long yellow bill** and **brown tarsus and feet.**
GVE-class: this is a **brown bird** with a **white breast** and a **white head**.
GVE: this bird has a **black and white head** with a **large black beak**.

Fig. 5 Example sentences generated by our GVE, GVE-image, GVE-class, as well as the baseline definition and description models. Correct properties are highlighted in green, mostly correct ones are highlighted in yellow, and incorrect ones are highlighted in red. Our GVE model correctly mentions image relevant and class relevant properties

are also more image relevant. For example, in Fig. 5 row 7 our GVE model correctly mentions visible properties of the *Hooded Merganser*, but other models fail in at least one property.

Comparing Definitions and Visual Explanations Figure 6 directly compares GVE explanations to definitions for three bird categories. Images on the left include a visual property of the bird species which is not present in the image on the right. Because the definition is the same for all image instances of a bird class, it can produce sentences which are not image relevant. For example, in the second row, the definition model says the bird has a *red spot on its head* which is true for the image on the left but not for the image on the right. In contrast, our GVE model mentions *red spot* only when it is present in the image.



This is a ***Red Bellied Woodpecker*** *because...*

Definition: this bird has a bright red crown and nape **white breast and belly** and black and white spotted wings and secondaries.
GVE: this bird has a red crown a black and white spotted wing and a **white belly**.

This is a ***Red Bellied Woodpecker*** *because...*

Definition: this bird has a bright red crown and nape **white breast and belly** and black and white spotted wings and secondaries.
GVE: this bird has a bright red crown and nape with with black and white striped wings.

This is a ***Downy Woodpecker*** *because...*

Definition: this bird has a white breast black wings and a **red spot** on its head.
GVE: this is a black and white bird with a **red spot** on its crown.

This is a ***Downy Woodpecker*** *because...*

Definition: this bird has a white breast black wings and a **red spot** on its head.
GVE: this is a white bird with a black wing and a black and white striped head.

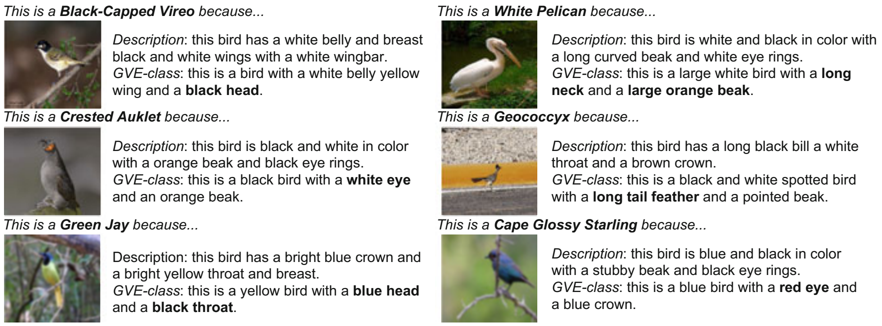
This is a ***Shiny Cowbird*** *because...*

Definition: this bird is black with a **long tail** and has a very short beak.
GVE: this is a black bird with a **long tail feather** and a pointy black beak.

This is a ***Shiny Cowbird*** *because...*

Definition: this bird is black with a **long tail** and has a very short beak.
GVE: this is a black bird with a small black beak.

Fig. 6 We compare generated explanations and definitions. All explanations on the left include an attribute which is not present on the image on the right. In contrast to definitions, our GVE model can adjust its output based on visual evidence



This is a ***Black-Capped Vireo*** *because...*

Description: this bird has a white belly and breast black and white wings with a white wingbar.
GVE-class: this is a bird with a white belly yellow wing and a **black head**.

This is a ***White Pelican*** *because...*

Description: this bird is white and black in color with a long curved beak and white eye rings.
GVE-class: this is a large white bird with a **long neck** and a **large orange beak**.

This is a ***Crested Auklet*** *because...*

Description: this bird is black and white in color with a orange beak and black eye rings.
GVE-class: this is a black bird with a **white eye** and an orange beak.

This is a ***Geococcyx*** *because...*

Description: this bird has a long black bill a white throat and a brown crown.
GVE-class: this is a black and white spotted bird with a **long tail feather** and a pointed beak.

This is a ***Green Jay*** *because...*

Description: this bird has a bright blue crown and a bright yellow throat and breast.
GVE-class: this is a yellow bird with a **blue head** and a **black throat**.

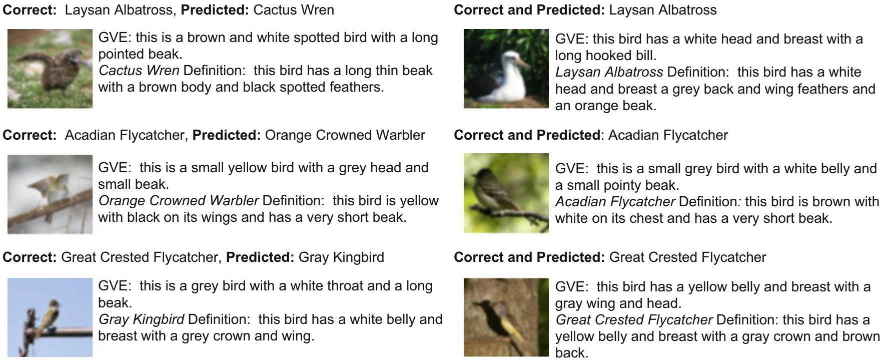
This is a ***Cape Glossy Starling*** *because...*

Description: this bird is blue and black in color with a stubby beak and black eye rings.
GVE-class: this is a blue bird with a **red eye** and a blue crown.

Fig. 7 Comparing sentences generated by the baseline description and GVE-class models. Though both are capable of accurately describing visual attributes, the GVE-class model captures more "class-specific" attributes

Effect of Discriminative Loss To determine how the discriminative loss impacts sentence generation, we compare the description and GVE-class models in Fig. 7. Neither model receives class information at test time, though the GVE-class model is explicitly trained to produced class specific sentences. Both models generate visually relevant sentences. However, the model trained with our discriminative loss contains properties specific to a class more often than the ones generated using the description model. For instance, for the class *black-capped vireo*, the GVE-class model mentions *black head* which is one of the most prominent distinguishing properties of this vireo type. For the *white pelican* image, the GVE-class model mentions highly discriminative features like *long neck* and *orange beak*.

Effect of Incorrect Class Prediction We qualitatively examine explanations for instances where the incorrect label is predicted (Fig. 8). In these scenarios, GVE explanations are frequently image relevant and mention features common in both



Correct: Laysan Albatross, **Predicted:** Cactus Wren

GVE: this is a brown and white spotted bird with a long pointed beak.
Cactus Wren Definition: this bird has a long thin beak with a brown body and black spotted feathers.

Correct and Predicted: Laysan Albatross

GVE: this bird has a white head and breast with a long hooked bill.
Laysan Albatross Definition: this bird has a white head and breast a grey back and wing feathers and an orange beak.

Correct: Acadian Flycatcher, **Predicted:** Orange Crowned Warbler

GVE: this is a small yellow bird with a grey head and small beak.
Orange Crowned Warbler Definition: this bird is yellow with black on its wings and has a very short beak.

Correct and Predicted: Acadian Flycatcher

GVE: this is a small grey bird with a white belly and a small pointy beak.
Acadian Flycatcher Definition*:* this bird is brown with white on its chest and has a very short beak.

Correct: Great Crested Flycatcher, **Predicted:** Gray Kingbird

GVE: this is a grey bird with a white throat and a long beak.
Gray Kingbird Definition: this bird has a white belly and breast with a grey crown and wing.

Correct and Predicted: Great Crested Flycatcher

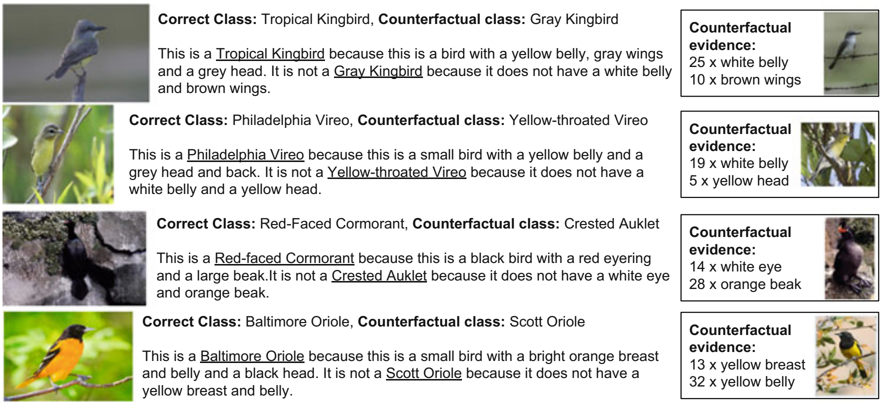
GVE: this bird has a yellow belly and breast with a gray wing and head.
Great Crested Flycatcher Definition: this bird has a yellow belly and breast with a gray crown and brown back.

Fig. 8 When the model predicts the wrong class, the sentence generated by our GVE model is image relevant and it frequently discusses attributes that are common between the image and the predicted class

the image instance and the predicted class. For example, in the first row of Fig. 8 the model mistakes the *Laysan Albatross* for the *Cactus Wren*. The explanation text includes many features also mentioned in the *Cactus Wren* definition (for example color and the spotted feathers) and is relevant to the image.

Counterfactual Explanations Another way of explaining a visual concept is through generating *counterfactual* explanations that indicate why the classifier does not predict another class label. To construct counterfactual explanations, we posit that if a visual property is discriminative for another class, i.e. a class that is different from the class that the query image belongs to, but not relevant for the query image, then this visual property is a *counterfactual* evidence. To discuss counterfactual evidence for a classification decision, we first hypothesize which visual evidence might indicate that the bird belongs to another class. We do so by searching for the most similar image that belongs to another class, i.e. counterfactual class. We then determine the counterfactual properties that are class specific, i.e. the visual properties that are mentioned the most frequently in the ground-truth sentences of the counterfactual class. Our counterfactual explanation mentions those visual properties that are not shared by the correct class, i.e. it mentions only the counterfactual evidence. For instance, "bird has a long flat bill" is negated to "bird does not have a long flat bill" where the counterfactual evidence is the "long flat bill".

In Fig. 9 we show our counterfactual explanation results. The top row shows an image of a *Tropical Kingbird* with a "yellow belly, grey wings and a grey head" whereas the class of the most similar image, i.e. *Gray Kingbird*, has the "white belly and brown wings" which are the counterfactual evidence that those should be mentioned in a counterfactual explanation. Similarly, for the last row, the selected *Baltimore Oriole* image is not a *Scott Oriole* because it "does not have a yellow breast and belly" as the bird actually has a "bright orange breast and belly". With these results we show that our GVE model is capable of reasoning about both factual



Correct Class: Tropical Kingbird, **Counterfactual class:** Gray Kingbird

This is a Tropical Kingbird because this is a bird with a yellow belly, gray wings and a grey head. It is not a Gray Kingbird because it does not have a white belly and brown wings.

Counterfactual evidence:
25 x white belly
10 x brown wings

Correct Class: Philadelphia Vireo, **Counterfactual class:** Yellow-throated Vireo

This is a Philadelphia Vireo because this is a small bird with a yellow belly and a grey head and back. It is not a Yellow-throated Vireo because it does not have a white belly and a yellow head.

Counterfactual evidence:
19 x white belly
5 x yellow head

Correct Class: Red-Faced Cormorant, **Counterfactual class:** Crested Auklet

This is a Red-faced Cormorant because this is a black bird with a red eyering and a large beak.It is not a Crested Auklet because it does not have a white eye and orange beak.

Counterfactual evidence:
14 x white eye
28 x orange beak

Correct Class: Baltimore Oriole, **Counterfactual class:** Scott Oriole

This is a Baltimore Oriole because this is a small bird with a bright orange breast and belly and a black head. It is not a Scott Oriole because it does not have a yellow breast and belly.

Counterfactual evidence:
13 x yellow breast
32 x yellow belly

Fig. 9 Counterfactual explanations. For the query image on the left, correct class effects the generation of the first sentence, i.e. "This is a …because …", the counterfactual class effects the generation of the second sentence, i.e. "It is not a …because …". On the right, we point to the counterfactual evidence. It lists the most frequent phrases found in the ground truth sentences of the counterfactual class which relate to the same nouns mentioned in the correct-class GVE explanation (the number indicates the occurence count). We also show the most similar image from the counterfactual class to the query image

and counterfactual evidence. We argue and emphasize that visual explanations that talk about the counterfactual evidence helps build a stronger cognitive model of the object at hand. Hence, a visual explanation system should preferably be fashioned with the capability of counterfactual reasoning.

6 Conclusion

Our work is an important step towards explaining deep visual models, a crucial capability required from intelligent systems. Visual explanation is a rich research direction, especially as the field of computer vision continues to employ and improve deep models which are not easily interpretable. We anticipate that future models will look "deeper" into networks to produce explanations and perhaps begin to explain the internal mechanism of deep models.

In summary, we have presented a novel image explanation framework which justifies the class prediction of a visual classifier. We proposed a novel reinforcement learning based loss which allows us to influence the kinds of sentences generated with a sentence level loss function. Though we focused on a discriminative loss in this work, we believe the general principle of a loss which operates on a sampled sentence and optimizes for a global sentence property is potentially beneficial in other applications. Our quantitative and qualitative evaluations demonstrate the potential of our proposed model and effectiveness of our novel loss function. Our GVE model goes beyond the capabilities of current captioning systems and effectively incorporates classification information to produce convincing factual and counterfactual explanations, potentially a key advance for adoption of many sophisticated AI systems.

References

Andreas J, Rohrbach M, Darrell T, Klein D (2016) Learning to compose neural networks for question answering. In: NAACL

Banerjee S, Lavie A (2005) Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, vol 29, pp 65–72

Berg T, Belhumeur P (2013) How do you tell a blackbird from a crow? In: ICCV, pp 9–16

Biran O, McKeown K (2014) Justification narratives for individual classifications. In: Proceedings of the AutoML workshop at ICML 2014

Core MG, Lane HC, Van Lent M, Gomboc D, Solomon S, Rosenberg M (2006) Building explainable artificial intelligence systems. In: Proceedings of the national conference on artificial intelligence, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, vol 21, p 1766

Doersch C, Singh S, Gupta A, Sivic J, Efros A (2012) What makes paris look like paris? ACM Transactions on Graphics 31(4)

Donahue J, Jia Y, Vinyals O, Hoffman J, Zhang N, Tzeng E, Darrell T (2013) Decaf: A deep convolutional activation feature for generic visual recognition. ICML

Donahue J, Hendricks LA, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K, Darrell T (2015) Long-term recurrent convolutional networks for visual recognition and description. In: CVPR

Escorcia V, Niebles JC, Ghanem B (2015) On the relationship between visual attributes and convolutional networks. In: CVPR

Fang H, Gupta S, Iandola F, Srivastava RK, Deng L, Dollár P, Gao J, He X, Mitchell M, Platt JC, et al (2015) From captions to visual concepts and back. In: CVPR, pp 1473–1482

Fong RC, Vedaldi A (2017) Interpretable explanations of black boxes by meaningful perturbation. arXiv preprint arXiv:170403296

Gao Y, Beijbom O, Zhang N, Darrell T (2016) Compact bilinear pooling. In: CVPR

Guadarrama S, Krishnamoorthy N, Malkarnenkar G, Venugopalan S, Mooney R, Darrell T, Saenko K (2013) Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In: ICCV, pp 2712–2719

Hendricks LA, Akata Z, Rohrbach M, Donahue J, Schiele B, Darrell T (2016) Generating visual explanations. In: ECCV

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780

Jia X, Gavves E, Fernando B, Tuytelaars T (2015) Guiding long-short term memory for image caption generation. ICCV

Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia, ACM, pp 675–678

Johnson WL (1994) Agents that learn to explain themselves. In: AAAI, pp 1257–1263

Karpathy A, Li F (2015) Deep visual-semantic alignments for generating image descriptions. In: CVPR

Kiros R, Salakhutdinov R, Zemel R (2014) Multimodal neural language models. In: Proceedings of the 31st International Conference on Machine Learning (ICML-14), pp 595–603

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: NIPS, pp 1097–1105

Kulkarni G, Premraj V, Dhar S, Li S, choi Y, Berg A, Berg T (2011) Baby talk: understanding and generating simple image descriptions. In: CVPR

Lacave C, Díez FJ (2002) A review of explanation methods for bayesian networks. The Knowledge Engineering Review 17(02):107–127

Lampert C, Nickisch H, Harmeling S (2013) Attribute-based classification for zero-shot visual object categorization. In: TPAMI

Lane HC, Core MG, Van Lent M, Solomon S, Gomboc D (2005) Explainable artificial intelligence for training and tutoring. Tech. rep., DTIC Document

Liu S, Zhu Z, Ye N, Guadarrama S, Murphy K (2016) Improved image captioning via policy gradient optimization of spider. arXiv preprint arXiv:161200370

Lomas M, Chevalier R, Cross II EV, Garrett RC, Hoare J, Kopack M (2012) Explaining robot actions. In: Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction, ACM, pp 187–188

Mao J, Xu W, Yang Y, Wang J, Yuille AL (2014) Explain images with multimodal recurrent neural networks. NIPS Deep Learning Workshop

Mao J, Huang J, Toshev A, Camburu O, Yuille A, Murphy K (2016) Generation and comprehension of unambiguous object descriptions. In: CVPR

Miller GA, Beckwith R, Fellbaum C, Gross D, Miller KJ (1990) Introduction to wordnet: An on-line lexical database*. International journal of lexicography 3(4):235–244

Papineni K, Roukos S, Ward T, Zhu WJ (2002) Bleu: a method for automatic evaluation of machine translation. In: ACL, pp 311–318

Park DH, Hendricks LA, Akata Z, Rohrbach A, Schiele B, Darrell T, Rohrbach M (2018) Multimodal explanations: Justifying decisions and pointing to the evidence. arXiv preprint arXiv:180208129

Ranzato M, Chopra S, Auli M, Zaremba W (2015) Sequence level training with recurrent neural networks. arXiv preprint arXiv:151106732

Reed S, Akata Z, Lee H, Schiele B (2016a) Learning deep representations of fine-grained visual descriptions. In: CVPR

Reed S, Akata Z, Yan X, Logeswaran L, Schiele B, Lee H (2016b) Generative adversarial text to image synthesis. ICML

Ren Z, Wang X, Zhang N, Lv X, Li LJ (2017) Deep reinforcement learning-based image captioning with embedding reward. arXiv preprint arXiv:170403899

Rennie SJ, Marcheret E, Mroueh Y, Ross J, Goel V (2016) Self-critical sequence training for image captioning. arXiv preprint arXiv:161200563

Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2016) Grad-cam: Visual explanations from deep networks via gradient-based localization. See https://arxivorg/abs/161002391v37(8)

Shortliffe EH, Buchanan BG (1975) A model of inexact reasoning in medicine. Mathematical biosciences 23(3):351–379

Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556

Teach RL, Shortliffe EH (1981) An analysis of physician attitudes regarding computer-based clinical consultation systems. In: Use and impact of computers in clinical medicine, Springer, pp 68–85

Van Lent M, Fisher W, Mancuso M (2004) An explainable artificial intelligence system for small-unit tactical behavior. In: Proceedings of the National Conference on Artificial Intelligence, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, pp 900–907

Vedantam R, Lawrence Zitnick C, Parikh D (2015) Cider: Consensus-based image description evaluation. In: CVPR, pp 4566–4575

Vinyals O, Toshev A, Bengio S, Erhan D (2015) Show and tell: A neural image caption generator. In: CVPR

Wah C, Branson S, Welinder P, Perona P, Belongie S (2011) The Caltech-UCSD Birds-200-2011 Dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology

Williams RJ (1992) Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning

Xu K, Ba J, Kiros R, Courville A, Salakhutdinov R, Zemel R, Bengio Y (2015) Show, attend and tell: Neural image caption generation with visual attention. ICML

Yeung S, Russakovsky O, Jin N, Andriluka M, Mori G, Fei-Fei L (2016) Every moment counts: Dense detailed labeling of actions in complex videos. In: CVPR

Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: ECCV

Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2015) Object detectors emerge in deep scene cnns

Zintgraf LM, Cohen TS, Adel T, Welling M (2017) Visualizing deep neural network decisions: Prediction difference analysis. arXiv preprint arXiv:170204595

Ensembling Visual Explanations

Nazneen Fatema Rajani and Raymond J. Mooney

Abstract Many machine learning systems deployed for real-world applications such as recommender systems, image captioning, object detection, etc. are ensembles of multiple models. Also, the top-ranked systems in many data-mining and computer vision competitions use ensembles. Although ensembles are popular, they are opaque and hard to interpret. Explanations make AI systems more transparent and also justify their predictions. However, there has been little work on generating explanations for ensembles. In this chapter, we propose two new methods for ensembling visual explanations for VQA using the localization maps for the component systems. Our novel approach is scalable with the number of component models in the ensemble. Evaluating explanations is also a challenging research problem. We introduce two new approaches to evaluate explanations—the comparison metric and the uncovering metric. Our crowd-sourced human evaluation indicates that our ensemble visual explanation is significantly qualitatively outperform each of the individual system's visual explanation. Overall, our ensemble explanation is better 61% of the time when compared to any individual system's explanation and is also sufficient for humans to arrive at the correct answer, just based on the explanation, at least 64% of the time.

Keywords Stacking with auxiliary features · Visual question answering · Visualization techniques for deep networks

1 Introduction

Visual Question Answering (VQA) (Malinowski and Fritz 2014; Antol et al. 2015), answering natural-language questions about images, requires both language and image understanding, language grounding capabilities, as well as common-sense

N. F. Rajani (✉) · R. J. Mooney
Department of Computer Science, The University of Texas at Austin, Austin, TX, USA
e-mail: nrajani@cs.utexas.edu; mooney@cs.utexas.edu

H. J. Escalante et al. (eds.), *Explainable and Interpretable Models in Computer Vision and Machine Learning*, The Springer Series on Challenges in Machine Learning, https://doi.org/10.1007/978-3-319-98131-4_7

knowledge. A variety of methods to address these challenges have been developed in recent years (Andreas et al. 2016a; 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. The top-performing VQA systems are ensembles of neural networks that perform significantly better than any of the underlying individual models (Fukui et al. 2016).

Although there have been several innovative and ground-breaking ideas deployed to solve VQA problems, the current state-of-the-art on real-world images is still approximately 12 points behind human accuracy.[1] One way to reduce this gap in performance is to analyze how various neural architectures arrive at their predicted answers, and then design heuristics or loss functions that overcome the shortcomings of current networks. Also, systems that can explain their decisions make them more trustworthy, transparent, and user-friendly (Aha et al. 2017; Gunning 2016). This has led to some work in generating explanations that help interpret the decisions made by CNNs (Goyal et al. 2016; Hendricks et al. 2016; Park et al. 2016; Ross et al. 2017). However, previous work focuses on generating explanations for individual models even though the top performing systems on various computer vision and language tasks are ensembles of multiple models. This motivated us to explore the problem of generating explanations for an ensemble using explanations from underlying individual models as input. In this chapter, we focus on ensembling *visual* explanations for VQA.

VQA systems have been shown to attend to relevant parts of the image when answering a question (Goyal et al. 2016). The regions of an image on which a model focuses can be thought of as a visual explanation for that image-question (IQ) pair. The localization map of a system is in the form of an intensity map and from this point on we will refer to it as the *explanation map*. The Guided Grad-CAM algorithm highlights the regions in an image that the model focuses on by generating a heat-map with intensity gradients (Selvaraju et al. 2017). We adapt the Guided Grad-CAM approach to generate heat-map visualizations for three different VQA systems—LSTM (Antol et al. 2015), HieCoAtt (Lu et al. 2016) and MCB (Fukui et al. 2016). By manually analyzing some of the visualizations from each of these systems, we found that all of them had some degree of noise with high variance depending on the IQ pair under consideration. We also observed that there was high variance across visualizations for different models even when they agreed on the answer for a given IQ pair. This motivated us to ensemble visualizations of the individual models such that the ensembled visual explanation: (1) aggregates visualizations from appropriate regions of the image, (2) discounts visualizations

[1] Based on the performance reported on the CodaLab Leader-board and human performance reported on the task in Antol et al. (2015).

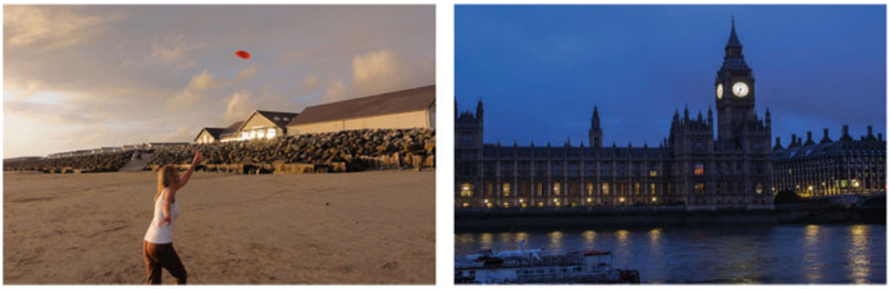
from regions that are not relevant, (3) reduces noise and (4) is superior to any individual system's visualization on a human evaluation.

Evaluating AI-system explanations is a challenging problem that has attracted attention in recent years (Samek et al. 2017; Ribeiro et al. 2016; Park et al. 2016; Das et al. 2017). The work in this area uses crowd-sourcing to evaluate explanations. However, most of these evaluations measure the extent to which a machine-generated explanation overlaps with a human-generated explanation, considering human explanation as the ground truth. This has several disadvantages. Research shows that human and deep-learning models do not attend to the same input evidence even when they produce the same output (Das et al. 2017). To aid interpretability and trust, machine explanations should accurately reflect the system's reasoning rather than try to produce "post-hoc rationalizations" that mimic human explanations and might convince users to *mistakenly* trust its results. Consequently, we propose two novel evaluation methods for judging the quality of explanations. Our first approach evaluates explanations by asking human subjects to compare and score two machine generated explanations side-by-side. Our second approach measures how accurately a human subject can arrive at the same decision as the system using only the information from a system-generated explanation. Results of the first evaluation indicate that, on average, our visual explanation ensemble is superior to any of the individual system's explanation 61% of the time, while an individual system's explanation is better only 32% of the time. Results on the second evaluation indicate that, on average, our visual explanation ensemble is sufficient to independently arrive at the correct answer at least 64% of the time, while an individual system's explanation is sufficient to arrive at the right answer at most 46% of the time.

2 Background and Related Work

Visual Question Answering (VQA) VQA is the task of answering a natural language question about an image with an appropriate word or phrase. Several datasets have been released in recent years for the VQA task. The DAtaset for QUestion Answering on Real-world images (DAQUAR) was the first dataset and benchmark for this task (Malinowski and Fritz 2014). The Visual Question Answering (VQA) dataset is the most well-known and widely used dataset for VQA. The dataset consists of images taken from the MS COCO dataset (Lin et al. 2014) with three questions and answers per image obtained through Mechanical Turk (Antol et al. 2015). Figure 1 shows a sample of images and questions from the VQA 2016 challenge.

Several deep learning models have been developed that combine a computer vision component with a linguistic component in order to solve the VQA challenge. Some of these models also use data-augmentation for pre-training. A non-deep learning approach to VQA uses a Bayesian framework to predict the form of the answer from the question (Kafle and Kanan 2016). Some of the deep learning



Q. Is that a frisbee?
A. Yes
Q. Is this a man or a woman?
A. Woman
Q. What color is the frisbee?
A. Red

Q. Is this a romantic spot that couples would like to go?
A. Yes
Q. What time of day is it?
A. Night
Q. How many spires below big ben's clock?
A. 10

Fig. 1 Random sample of images with questions and ground truth answers taken from the VQA dataset

models that attempt to solve VQA are iBowIMG (Zhou et al. 2015b), the DPPNet (Noh et al. 2016), Neural Module Networks (NMNs) (Andreas et al. 2016b), an LSTM (Antol et al. 2015), HieCoAtt (Lu et al. 2016) and MCB (Fukui et al. 2016). iBowIMG concatenates the image features with the bag-of-word question embedding and feeds them into a softmax classifier to predict the answer, resulting in performance comparable to other models that use deep or recurrent neural networks. iBowIMG beats most VQA models considered in their paper. DPPNet, on the other hand, learns a CNN with some parameters predicted from a separate parameter prediction network, which uses a Gated Recurrent Unit (GRU) to generate a question representation and maps the predicted weights to a CNN via hashing. DPPNet uses external data (data-augmentation) in addition to the VQA dataset to pre-train the GRU. Another well-known VQA model is the Neural Module Network (NMN) that generates a neural network on the fly for each individual image and question. This is done through choosing from various sub-modules based on the question and composing these to generate the neural network, e.g., the `find[x]` module outputs an attention map for detecting `x`. The question is first parsed into a symbolic expression and based on this expression, modules are composed to answer the query. The whole system is trained end-to-end through backpropagation. The LSTM model uses a VGGNet (Simonyan and Zisserman 2015) to obtain embeddings for the image and combines it with an LSTM (Hochreiter and Schmidhuber 1997) embedding of each question via element-wise multiplication. The HieCoAtt model jointly reasons about the visual and language components using two types of "co-attention"—parallel and alternating. The MCB model combines the vision and language vector representations using an outer product. Multimodal Compact Bilinear pooling (MCB) (Gao et al. 2016) is used to efficiently and expressively combine the image representation formed using the 152-layer Residual Network (He et al. 2016) with the question embedding formed using an LSTM (Hochreiter and Schmidhuber 1997).

Stacking with Auxiliary Features (SWAF) Ensembling multiple systems is a well known standard approach to improving accuracy in machine learning (Dietterich 2000). Stacking with Auxiliary Features (Rajani and Mooney 2017) is a recent ensembling algorithm that learns to combine outputs of multiple systems using features of the current problem as context. 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. It has previously been applied effectively to information extraction (Viswanathan et al. 2015), entity linking (Rajani and Mooney 2016), ImageNet object detection (Rajani and Mooney 2017) as well as VQA (Rajani and Mooney 2018). The SWAF approach extracts features from the Image-Question pair under consideration, as well as the component models and provides this information to the classifier. The meta-classifier then learns to predict whether a specific generated answer is correct or not. Figure 2 shows an overview of the SWAF approach.

Generating Explanations Ensembles of deep learning models have been widely used on several real-world vision and language problems. Despite their success, ensembles lack transparency and are unable to explain their decisions. On the other hand, humans can justify their decisions in natural language as well as point

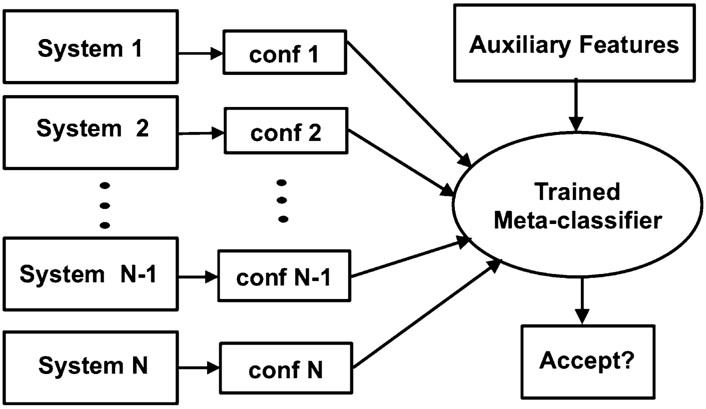


Fig. 2 Ensemble architecture using stacking with auxiliary features. Given an input, the ensemble judges every possible question-answer pair produced by the component systems and determines the final output answer

to the visual evidence that supports their decision. AI systems that can generate explanations supporting their predictions have several advantages (Johns et al. 2015; Agrawal et al. 2016). This has motivated recent work on explainable AI systems, particularly in computer vision (Antol et al. 2015; Goyal et al. 2016; Park et al. 2016). Hendricks et al. (2016) developed a deep network to generate natural language justifications for a fine-grained object classifier. A variety of work has proposed methods to visually explain decisions. Berg and Belhumeur (2013) use discriminative visual patches, whereas Zhou et al. (2015a) aim to understand intermediate features which are important for end decisions by naming hidden neurons that detect specific concepts. However, there has been no prior work on generating explanations for *ensembles* of multiple AI systems. In this chapter, we propose algorithms for ensembling visual explanations of deep learning models that can be used to explain the decision of the ensemble. We demonstrate the success of our approach on the challenging task of Visual Question Answering (VQA).

Evaluating Explanations Evaluating explanations generated by AI systems is a challenging problem and has attracted some attention in recent years. Although crowd-sourced human evaluation has been typically used to evaluate explanations, the actual metrics and approaches have differed widely across tasks and domains. Hendricks et al. (2016) used human experts on bird watching to evaluate explanations for fine-grained bird classification and asked them to rank the image-explanation pairs. On the other hand, Das et al. (2017) collect human attention maps for VQA by instructing human subjects on Mechanical Turk (MTurk) to sharpen parts of a blurred image that are important for answering the questions accurately. Typical explanation evaluation metrics rely on annotated ground truth explanations (Park et al. 2016; Goyal et al. 2016; Das et al. 2017). Selvaraju et al. (2017) evaluated explanations for image captioning by instructing human subjects on MTurk to select if a machine generated explanation is reasonable or not based on the predicted output. In this chapter, we propose two new evaluation approaches that are not dependent on ground truth explanations. Our work is the first to evaluate explanations for VQA that does not rely on human-generated explanation. This is important because research shows that machines and humans do not have the same "view" of visual explanations (Das et al. 2017).

3 Algorithms for Ensembling Visual Explanations

Our goal is to generate visual explanations for ensemble VQA systems. We do this by ensembling explanations of the component models, using heuristics to constrain the ensemble explanation such that it is faithful to and supports the ensemble's prediction. Our strategy depends on the individual component models' answer and visual explanation for a given IQ pair. We first build an ensemble model that uses Stacking With Auxiliary Features (SWAF) (Rajani and Mooney 2017) to combine outputs of three component systems. We then generate an explanation for the ensemble by combining the visual explanations of the component systems.

We generate model-specific explanation maps for each IQ pair using Grad-CAM (Selvaraju et al. 2017). First, the gradient of the score y^c for the predicted class c is computed before the softmax layer with respect to the explanation maps A^k of a convolutional layer. Then, the gradients flowing back are global average pooled to obtain the neuron importance weights.

$$w_k^c = \overbrace{\frac{1}{Z}\sum_i \sum_j}^{\text{global average pooling}} \qquad \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{backprop gradients}}$$

The above weights capture the importance of a convolutional explanation map k for the output class c. Z is the total number of pixels and i,j are the indices in the explanation map. A ReLU over the weighted combination of the explanation maps results in the required explanation-map for the output class as follows:

$$H^c = ReLU\left(\sum_k w_k^c A^k\right)$$

We generate such an explanation-map for each of the component VQA models in the ensemble. Thereafter, we combine these explanation-maps to create an explanation for the ensemble. We propose two novel methods for generating an ensembled visual-explanation that reflects the behavior of the ensemble: Weighted Average (WA) and Penalized Weighted Average (PWA).

3.1 Weighted Average Ensemble Explanation

As the name suggests, this approach averages the explanations of the component models proportional to their weights. The Weighted Average (WA) ensemble explanation is calculated as follows:

$$E_{i,j} = \begin{cases} \frac{1}{|K|}\sum_{k\in K} w_k A_{i,j}^k, & \text{if } A_{i,j}^k \geq t \\ 0, & \text{otherwise} \end{cases}$$
$$\text{subject to } \sum_{k\in K} w_k = 1 \tag{1}$$

Here, E is the explanation map of the ensemble, i and j are used to index into the explanation map entries, K is the set of component systems, w_k and A^k are the weights and explanation maps respectively for each of the component systems, and t is a thresholding value.

Thresholding the pixel values for the maps before or after averaging worked well for reducing noise as well as eliminating several low-intensity regions that arose as a result of combining multiple noisy maps. A weighted combination of the component feature maps worked better than using equal weights across all component systems. We weight the maps of the component systems proportional to their performance on the validation set, subject to the constraint that the weights sum to one. The ensemble explanation only combines maps of individual systems that agree with the ensemble on the answer. If some component systems do not agree with the ensemble, this approach ignores them. However, information from the explanation maps of such disagreeing systems can be used to adjust the ensemble explanation, as in the following approach.

3.2 *Penalized Weighted Average Ensemble Explanation*

Component VQA systems that agree with the ensemble on the answer for an IQ pair have relevant explanation maps that reflect how the model arrived at its prediction. On the other hand, component systems that do not agree with the ensemble's output answer have explanation maps that are potentially irrelevant to the ensemble's answer and can be discounted from the ensemble's explanation. The Penalized Weighted Average (PWA) ensemble explanation is calculated as follows:

$$E_{i,j} = \begin{cases} \frac{1}{|K||M|} \sum_{k \in K} \sum_{m \in M} \overbrace{w_k A^k_{i,j} - w_m I^m_{i,j}}^{\text{p}}, & \text{if } p \geq t \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$\text{subject to} \sum_{k \in K} w_k + \sum_{m \in M} w_m = 1$$

Here, I^m is an explanation map of a component system that does not agree with the ensemble and M is the total number of such systems. This assumes that a system that does not agree with the ensemble's answer is highlighting regions of the image that are *not* relevant, so we down-weight those regions in the explanation map for the ensemble. Another variation we explored is forcing a component model that does not agree with the ensemble to produce an explanation map for the alternate answer picked by the ensemble. We then calculate the ensemble explanation as in the previous section, where all systems agree on the output.

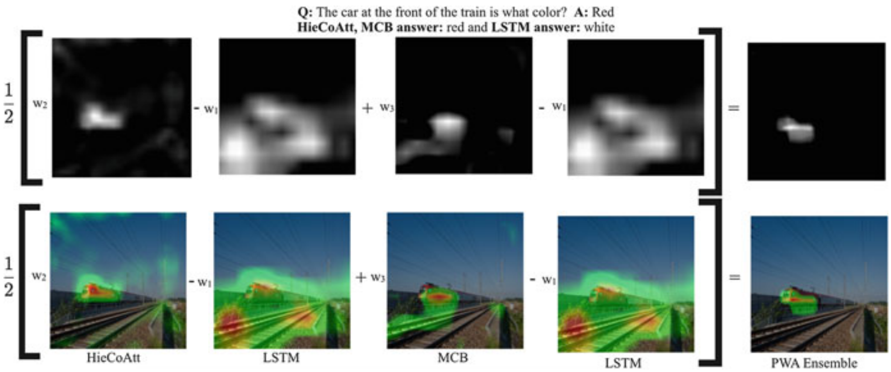

Q: The car at the front of the train is what color? A: Red
HieCoAtt, MCB answer: red and LSTM answer: white
1/2
w2
- w1
+ w3
- w1
=
1/2
w2
- w1
+ w3
- w1
=
HieCoAtt
LSTM
MCB
LSTM
PWA Ensemble

Fig. 3 The top row shows the process of ensembling visual explanation for an IQ pair when the ensemble model agrees with the MCB and HieCoAtt models (ans: “red”) and disagrees with the LSTM model (ans: “white”). The bottom row shows the reference IQ pair along with the individual systems and the ensemble visual explanation as a heat-map

3.3 Agreement with N Systems

A visual explanation ensemble can be generated for N component models using Eqs. (1) and (2) and it scales with N. We consider three component VQA systems and there are three scenarios that arise depending on whether the ensemble model agrees with all three, any two, or only one of the component systems. Figure 3 shows the process of ensembling visual explanations for an IQ pair for the second scenario. For all the scenarios, we first generate a gray-scale GradCam visualization for each of the component systems. Thereafter, we generate the ensemble explanation using the aforementioned approaches depending on the scenario under consideration.

We observed that our ensemble model agreed with all *three* systems on approximately half of the VQA test set. In this case, we use the WA approach described in Sect. 3.1 to generate the ensemble explanation map. The MCB component model (Fukui et al. 2016), which uses the 152-layer ResNet network had the highest weight followed by the HieCoAtt (Lu et al. 2016) and the LSTM (Antol et al. 2015) models, that use VGGNet. For approximately one-fourth of the test set, our ensemble model agreed with exactly *two* component systems. For this scenario, we combine the explanation maps using both the WA and PWA approaches by, respectively, ignoring or down-weighting the system that does not agree with the ensemble’s answer. When the ensemble model agreed with only *one* component system’s output, we generated the ensemble explanation map in two ways. First, the ensemble explanation was set equal to the explanation of the system it agrees with, minus the explanation of the systems it does not agree with, as in Eq. (2). Second, we force the systems that do not agree with the ensemble to produce explanation maps for the answer produced by the ensemble and then use those maps to calculate the ensemble explanation map using Eq. (1).

4 Evaluating Explanations

A good explanation evaluation metric tests how well an explanation supports the decision made by the system. In light of this, we propose two new crowd-sourced human evaluation metrics and use them to assess our ensemble explanation. The first metric asks human judges to compare two machine-generated explanations and is called the *comparison* metric, while the second metric determines if the input evidence highlighted by an explanation is sufficient to allow a human judge to independently arrive at the same answer and is called the *uncovering* metric.

4.1 Comparison Metric

For the comparison metric, we showed two visual explanations side-by-side to workers on Amazon Mechanical Turk (AMT) along with the image-question pair as well as the ensemble model's answer and ask them "Which picture highlights the part of the image that best supports the answer to the question?". One image is the explanation map of the ensemble while the other is the explanation map of one of the systems that the ensemble agrees with. We provide detailed instructions along with an example to show what a good visualization looks like. Apart from picking one of the two images as superior, we also give two more options—"cannot decide" and "wrong answer" (for when the judge believes the given answer is incorrect).

Since both visual explanations are machine generated, there is no question of judging explanations based on their similarity to human ones. The evaluation simply compares explanations based on whether they highlight regions of the image that support the answer. To produce the final visualizations, all explanation maps are converted from gray-scale to heat-maps based on pixel intensity ranges. We note that the ensemble explanation is compared to an individual system's explanation map only if that system produced the same output answer as the ensemble. When multiple individual systems agree with the ensemble on an output for an image-question pair, then the ensemble explanation is compared to one of the individual system's explanation chosen randomly with equal probability.

4.2 Uncovering Metric

The uncovering metric tests whether the input evidence highlighted by a visual explanation is sufficient to allow a human judge to independently arrive at the same prediction as the model that produced it. A judge is shown the question and a partially uncovered image and asked to pick an answer from a multiple choice list (taken from the multiple-choice version of the standard VQA evaluation (Antol et al. 2015)). A fraction of the most heavily weighted pixels in the visual explanation are "uncovered" and the judge is asked to pick the answer, or choose "cannot decide" when the partially uncovered image does not support an answer.

First, only the top one-third most intense pixels are uncovered, followed by the top two-thirds and finally, the entire explanation map is uncovered. The regions of the image that are not part of the explanation (zero-weighted pixels) are never exposed. Turkers have to select the first partially uncovered image that is sufficient to pick the answer from the available choices. The Turkers were asked to complete two parts for each instance and given the following instruction, "Part I: Select an answer based on the question and set of partially visible images; Part II: Select the first image from the set that was sufficient to arrive at the answer." In this way, we evaluate both the ensemble's explanation as well those of the individual component systems and compare the percentage of the explanation map that was uncovered versus the accuracy of the answers selected by the human subjects. An explanation is better to the extent that it allows humans to pick the correct answer from a partially uncovered image showing just the most highly-weighted evidence used by the system. Figure 4 shows the step-by-step uncovering of the explanation maps for the ensemble and LSTM models along with the question and answer options provided to the Turkers. As evident, the ensemble explanation maps are more precise than that of the LSTM model's explanation maps even though it highlights a bigger region of the image.

A drawback of the aforementioned approach is that if a system tends to highlight a larger proportion of the overall image, then it would have an undue advantage over other systems since it would tend to cause a larger fraction of the overall

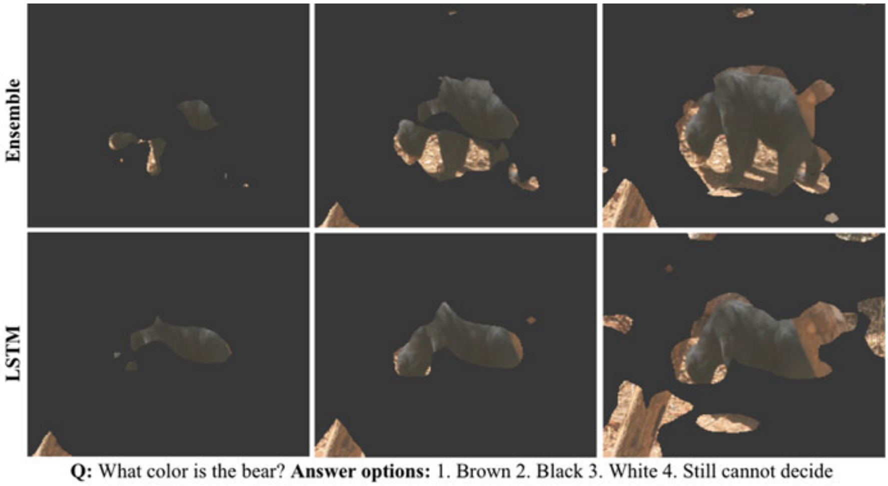
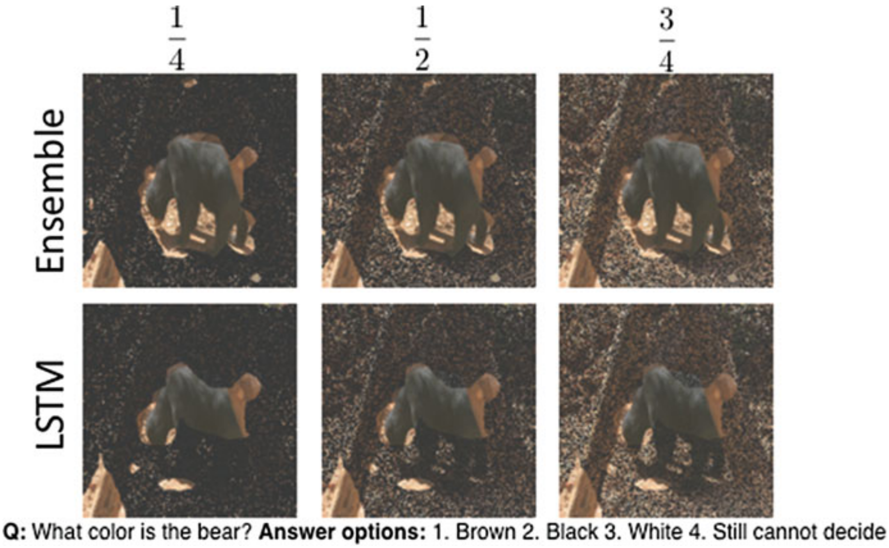

Ensemble
LSTM
Q: What color is the bear? Answer options: 1. Brown 2. Black 3. White 4. Still cannot decide

Fig. 4 Top and bottom rows show the gradual uncovering of the explanation map from left-to-right for the ensemble and LSTM models respectively. The second image in the top row is sufficient to arrive at the answer but even the third image in the bottom row is barely enough to decide on an answer, since it is not clear if that object is a bear



Q: What color is the bear? **Answer options:** 1. Brown 2. Black 3. White 4. Still cannot decide

Fig. 5 Top and bottom rows show the gradual normalized uncovering of the entire image from left-to-right for the ensemble and LSTM models respectively

image to be revealed. This is because the evaluation metric discussed above is based on uncovering some fraction of the *non-zero-weighted region* of the image. To overcome this drawback, we also measured an alternate *normalized* version of the uncovering metric that revealed a fraction of the *entire image* as opposed to just the non-zero portion highlighted in the explanation map. So, at each step, we showed Turkers images that uncovered one-fourth, one-half, and three-fourths of the highest-weighted pixels in the entire image. In order maintain the ratio, we frequently had to uncover a number of zero-intensity pixels. The zero intensity pixels uncovered are randomly chosen from the entire image, giving rise to a "snow like" effect as shown in Fig. 5. Arguably, this approach gives a more fair comparison between explanation maps of various systems. Our choice for the fractions of the uncovered image at each step was decided so that the total number of images to be judged is neither too many (if a very small fraction is revealed) nor too less (if a very big fraction is revealed). The optimal number of fractions improves the effectiveness of the uncovering metric.

4.3 Crowd-Sourced Hyper-Parameter Tuning

The weighted average and the penalized weighted average methods for ensembling explanation maps depend on the parameter t which thresholds the pixel values for the maps. We also use a threshold parameter for eliminating noise in the explanation maps of individual systems. We used crowd-sourcing to determine the optimal value of t for each approach. The idea is to optimize the explanation map generation for the actual evaluation metric, which as discussed above, uses human judgment. In a similar manner, crowd-sourcing was recently used to tune the hyper-parameters of a policy network for a reinforcement learning agent (Fridman et al. 2018).

We used Mechanical Turk to search for a good value of the parameter t for each of the individual systems as well as the ensemble system. We chose 50 random instances from the VQA validation set for judging the value of the threshold parameter. The Turkers had to select the image that highlighted just the right amount of the appropriate regions to answer the given question. They were shown images with explanation maps thresholded in steps of $0.1, 0.15, 0.2, 0.25$. The human judges also had the option of stating that the highlighted region was not appropriate for answering the question. We found that a threshold less than 0.1 or more than 0.25 generated maps that were either too noisy or not sufficiently highlighted, respectively. The result of crowd-sourcing the choice of threshold parameter was that a threshold of 0.2 worked well for the ensemble and all individual systems except HieCoAtt. The optimal threshold for HieCoAtt was 0.15. The pixel intensities greater than t were normalized to lie between zero and one. We used the threshold parameters obtained from crowd-sourcing for all our evaluations. Results improved slightly by searching for the right threshold value for each system compared to using a uniform threshold for all systems.

5 Experimental Results and Discussion

We evaluated our visual explanation maps for the VQA ensemble using the aforementioned comparison and uncovering metrics. The image-question pairs used for generating and evaluating explanations were taken from the test-set of the VQA challenge. Three workers evaluated each of 200 random test IQ pairs for each of the different explanation ensembling methods discussed in Sect. 3 for each of the metrics. We then aggregated the Turkers decisions using voting, and when there is no agreement among workers, we classified those instances under a "no agreement" category and we ignored the instances for which the majority of Turkers thought the ensemble's answer was incorrect. For the comparison and uncovering metrics, we obtained inter-annotator agreement of 88% and 79% respectively.

Comparison Metric Table 1 shows the results obtained when comparing the ensemble explanation using the weighted average (WA) and the penalized weighted average (PWA) approaches with the individual systems' explanation. The results are averaged across instances of image-question pairs for each individual system. The rows in Table 1 show the percentage of time the Turkers found the single system vs. the ensemble explanation map to be superior. For a small percentage of cases, the Turkers were not able to decide if either the single system's or the ensemble's explanation was superior, displayed in the third column and for the

Table 1 Results obtained using the comparison metric for evaluating the ensemble explanation map in terms of the percentage of cases a system's explanation was preferred, averaged for each ensembling approach

Approach	Single system	Ensemble	Cannot decide
Ensemble (WA)			
LSTM	36	**58**	3
HieCoAtt	27	**62**	6
MCB	41	52	2
Ensemble (PWA)			
LSTM	28	**64**	3
HieCoAtt	26	**69**	1
MCB	35	**61**	1

The remaining percentage of the time there was no majority agreement among human subjects. The bold figures imply statistical significance (p-value < 0.05)

remaining percentage of time there was no majority agreement among the Turkers. We found that, on an average, the Turkers considered our ensemble's explanation superior to an individual model's explanation 61% of the time.

We used the WA approach for generating the ensemble explanation when more than one system agreed with the ensemble's output prediction. We observed that the ensemble's explanation for an image-question (IQ) pair was better than the LSTM, the HieCoAtt and the MCB models 58%, 62% and 52% of the time respectively. We performed a pairwise t-test with a significance level of 0.05 and found that the ensemble explanation using the weighted average approach was significantly better (p-value < 0.05) than the LSTM and the HieCoAtt systems' explanation.

When there was at least one individual system that did not agree with the ensemble on an output, we used the PWA approach for generating the explanation map. We observed that the ensemble's explanation for an IQ pair was better than the LSTM, the HieCoAtt and the MCB models 64%, 69% and 61% of the time respectively. We found that the ensemble explanation using the penalized weighted average approach was significantly better (p-value < 0.05) than all the three individual system's explanation on a pairwise t-test with significance level 0.05. We note that there were scenarios, like when two systems agreed with the ensemble, when we compared both the WA and PWA ensemble explanation maps to an individual system's explanation map. In such scenarios, we observed that PWA performed better than WA.

Uncovering Metric After each step of uncovering either 1/3, 2/3, or all of the explanation map for an image, we measured for what percentage of the test cases a human judge both decided they were able to answer the question and picked the correct answer. We found that, on an average, the penalized weighted average (PWA) ensemble explanation was sufficient 69% of the time and the weighted average (WA) ensemble explanation was sufficient 64% of the time to arrive at the correct answer from a set of answers for a given image-question pair. For the same image-question pairs and answer choices, the LSTM, the HieCoAtt, and the MCB models had explanation maps that were sufficient to arrive at the right answer

Table 2 Results obtained using the uncovering metric averaged over image-question pairs

System	One-third	Two-thirds	Entire map
Ensemble (PWA)	29	35	69
Ensemble (WA)	17	28	64
LSTM	10	22	42
HieCoAtt	9	19	38
MCB	11	20	46

Shows the percentage of cases for which a partially revealed image was sufficient to arrive at the correct answer

Table 3 Results obtained using the uncovering metric averaged over image-question pairs

System	One-fourth	One-half	Three-fourths
Ensemble (PWA)	23	38	76
Ensemble (WA)	21	34	71
LSTM	10	24	65
HieCoAtt	10	23	57
MCB	12	25	64

Shows the percentage of cases for which a partially revealed image was sufficient to arrive at the correct answer based on the normalized uncovered pixel ratio

only 42%, 38% and 46% of the time respectively. Table 2 shows the breakup of these percentages across the three partially uncovered images ranging from the least visible to entirely uncovered explanation maps for the ensemble and each of the individual models.

We observed that the Turkers were unable to decide on an answer based on just the PWA and WA ensemble explanation and required the entire image for about 14% and 17% of the questions respectively. However, for the individual models, the same fraction was 43% for the LSTM, 51% for the HieCoAtt and 42% for the MCB models. There was no agreement among the Turkers for the remaining percentage of cases for each system.

For the results in Table 2, the ensemble explanation generated using PWA was significantly better (p-value < 0.05) than all the three individual systems' explanation on a pairwise t-test with significance level 0.05. On the other hand, the ensemble explanation generated using WA was significantly better (p-value $<$ 0.1) than all the three individual systems' explanation on a pairwise t-test with significance level 0.1. Also, the ensemble explanations generated by neither WA or PWA were significantly better than the other.

We also experimented with uncovering a fraction of the *entire image* and not just the explanation map. The human judges were shown images that had 1/4, 1/2 and 3/4 of the uncovered regions as shown in Fig. 5. Table 3 shows the results obtained when uncovering with respect to the entire image and not just the explanation map. We found that, on an average, the PWA and the WA ensemble explanation were sufficient 76% and 71% of the time to arrive at the correct answer for a given image-question pair. For the same image-question pairs and answer choices, the LSTM, the HieCoAtt, and the MCB models had explanation maps that were sufficient to arrive at the right answer only 65%, 57% and 64% of the time respectively. We observed that the difference between the performance of the ensemble and individual systems when uncovering with respect to the entire image is not as pronounced as uncovering with respect to the explanation map; however, the overall trends in the results are very similar. We found that the ensemble explanation using the PWA approach was significantly better (p-value < 0.1) than all the three individual system's explanation on a pairwise t-test with significance level 0.1. On the other hand, the ensemble explanation using the WA approach was significantly better (p-value < 0.1) than the HieCoAtt system's explanation. Also, the ensemble explanations generated by neither WA or PWA were significantly better than the other.

6 Conclusions and Future Directions

Visual explanations, which highlight the parts of an image that support a vision system's conclusion, can help us understand the decisions made by Visual Question Answering (VQA) systems and thereby aid error analysis, increase transparency, and help build trust with human users. We have presented the first approaches to ensembling such visual explanations. We proposed two novel methods for combining the explanation maps of multiple systems to produce improved ensemble explanations. Crowd-sourced human evaluations indicated that our ensemble visual explanation is superior to each of the component system's visual explanation.

Research on explainable AI systems is incomplete without a good evaluation metric to measure their effectiveness and usefulness. In this chapter, we proposed two such metrics that measure explanation quality without the need for a ground-truth human-generated explanation. Our evaluation metrics rely on crowd-sourced human judgments on simple tasks involving comparing visual explanations, or making decisions from partially revealed images. We demonstrated our metrics by evaluating explanation maps generated by our ensemble system as well as three component VQA models. On average, our ensemble's explanation was more interpretable than the individual component models' explanation 61% of the time using the comparison metric and was sufficient to allow humans to arrive at the correct answer at least 64% of the time as indicated by the uncovering metric.

In the future, we plan to explore using *textual* explanations along with the visual explanations for VQA, as done by Park et al. (2016). We believe that the words in the question to which a system attends to are also as important as the regions in an image. One way of finding natural-language concepts in sub-regions of the image that contributed to the system's answer is by using *network dissection* (Bau et al. 2017). In this method, the semantics of the hidden units in a CNN are scored on how well they detect a number of visual concepts including objects, parts, scenes, textures, materials, and colors. These natural-language concepts can then be used to generate a coherent explanatory sentence using either a template-based approach or a trained LSTM. Ensembling natural-language explanations obtained from each of the individual component systems is another interesting and challenging future direction. Finally, the ensembled visual explanation generated using the approach described in this chapter can be combined with the ensembled textual explanation so that the natural-language concepts in the textual explanation directly point to corresponding visual-explanation regions in the image.

References

Agrawal A, Batra D, Parikh D (2016) Analyzing the behavior of visual question answering models. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP2016)

Aha DW, Darrell T, Pazzani M, Reid D, Sammut C, (Eds) PS (2017) Explainable Artificial Intelligence (XAI) Workshop at IJCAI. URL http://home.earthlink.net/~dwaha/research/meetings/ijcai17-xai/

Andreas J, Rohrbach M, Darrell T, Klein D (2016a) Learning to compose neural networks for question answering. In: Proceedings of NAACL2016

Andreas J, Rohrbach M, Darrell T, Klein D (2016b) Neural module networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2016), pp 39–48

Antol S, Agrawal A, Lu J, Mitchell M, Batra D, Lawrence Zitnick C, Parikh D (2015) VQA: Visual Question Answering. In: Proceedings of ICCV2015

Bau D, Zhou B, Khosla A, Oliva A, Torralba A (2017) Network dissection: Quantifying interpretability of deep visual representations. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 3319–3327

Berg T, Belhumeur PN (2013) How do you tell a blackbird from a crow? In: Proceedings of ICCV2013

Chen K, Wang J, Chen LC, Gao H, Xu W, Nevatia R (2015) ABC-CNN: An attention based convolutional neural network for Visual Question Answering. arXiv preprint arXiv:151105960

Das A, Agrawal H, Zitnick L, Parikh D, Batra D (2017) Human attention in visual question answering: Do humans and deep networks look at the same regions? Computer Vision and Image Understanding 163:90–100

Dietterich T (2000) Ensemble methods in machine learning. In: Kittler J, Roli F (eds) First International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science, Springer-Verlag, pp 1–15

Fridman L, Jenik B, Terwilliger J (2018) DeepTraffic: Driving Fast through Dense Traffic with Deep Reinforcement Learning. arXiv preprint arXiv:180102805

Fukui A, Park DH, Yang D, Rohrbach A, Darrell T, Rohrbach M (2016) Multimodal Compact Bilinear pooling for Visual Question Answering and Visual Grounding. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP2016)

Gao Y, Beijbom O, Zhang N, Darrell T (2016) Compact bilinear pooling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2016), pp 317–326

Goyal Y, Mohapatra A, Parikh D, Batra D (2016) Towards Transparent AI Systems: Interpreting Visual Question Answering Models. In: International Conference on Machine Learning (ICML) Workshop on Visualization for Deep Learning

Gunning D (2016) Explainable Artificial Intelligence (XAI), DARPA Broad Agency Announcement, URL https://www.darpa.mil/attachments/DARPA-BAA-16-53.pdf

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778

Hendricks LA, Akata Z, Rohrbach M, Donahue J, Schiele B, Darrell T (2016) Generating Visual Explanations. In: Proceedings of the European Conference on Computer Vision (ECCV2016)

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural computation 9(8):1735–1780

Johns E, Mac Aodha O, Brostow GJ (2015) Becoming the expert-interactive multi-class machine teaching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2015)

Kafle K, Kanan C (2016) Answer-type prediction for visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2016)

Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: Common objects in context. In: European Conference on Computer Vision (ECCV2014), Springer, pp 740–755

Lu J, Yang J, Batra D, Parikh D (2016) Hierarchical question-image co-attention for visual question answering. In: Advances In Neural Information Processing Systems (NIPS2016), pp 289–297

Malinowski M, Fritz M (2014) A multi-world approach to question answering about real-world scenes based on uncertain input. In: Advances in Neural Information Processing Systems (NIPS2014), pp 1682–1690

Noh H, Hongsuck Seo P, Han B (2016) Image question answering using convolutional neural network with dynamic parameter prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2016), pp 30–38

Park DH, Hendricks LA, Akata Z, Schiele B, Darrell T, Rohrbach M (2016) Attentive explanations: Justifying decisions and pointing to the evidence. arXiv preprint arXiv:161204757

Rajani NF, Mooney RJ (2016) Combining Supervised and Unsupervised Ensembles for Knowledge Base Population. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP2016), URL http://www.cs.utexas.edu/users/ai-lab/pub-view.php?PubID=127566

Rajani NF, Mooney RJ (2017) Stacking With Auxiliary Features. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI2017), Melbourne, Australia

Rajani NF, Mooney RJ (2018) Stacking With Auxiliary Features for Visual Question Answering. In: Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies

Ribeiro MT, Singh S, Guestrin C (2016) 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 (KDD2016)

Ross AS, Hughes MC, Doshi-Velez F (2017) Right for the right reasons: Training differentiable models by constraining their explanations. In: Proceedings of IJCAI2017

Samek W, Binder A, Montavon G, Lapuschkin S, Müller KR (2017) Evaluating the visualization of what a deep neural network has learned. IEEE Transactions on Neural Networks and Learning Systems

Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-cam: Visual explanations from deep networks via gradient-based localization. In: The IEEE International Conference on Computer Vision (ICCV2017)

Simonyan K, Zisserman A (2015) Very Deep Convolutional Networks for Large-scale Image Recognition. In: Proceedings of ICLR2015

Viswanathan V, Rajani NF, Bentor Y, Mooney RJ (2015) Stacked Ensembles of Information Extractors for Knowledge-Base Population. In: Association for Computational Linguistics (ACL2015), Beijing, China, pp 177–187

Wolpert DH (1992) Stacked Generalization. Neural Networks 5:241–259

Xu H, Saenko K (2016) Ask, Attend and Answer: Exploring question-guided spatial attention for visual question answering. In: Proceedings of ECCV2016

Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2015a) Object detectors emerge in deep scene CNNs. In: Proceedings of the International Conference on Learning Representations (ICLR2015)

Zhou B, Tian Y, Sukhbaatar S, Szlam A, Fergus R (2015b) Simple baseline for visual question answering. arXiv preprint arXiv:151202167

Explainable Deep Driving by Visualizing Causal Attention

Jinkyu Kim and John Canny

Abstract Deep neural perception and control networks are likely to be a key component of self-driving vehicles. These models need to be explainable—they should provide easy-to-interpret rationales for their behavior—so that passengers, insurance companies, law enforcement, developers etc., can understand what triggered a particular behavior. Here, we explore the use of visual explanations. These explanations take the form of real-time highlighted regions of an image that causally influence the network's output (steering control). Our approach is two-stage. In the first stage, we use a visual attention model to train a convolutional network end-to-end from images to steering angle. The attention model highlights image regions that potentially influence the network's output. Some of these are true influences, but some are spurious. We then apply a causal filtering step to determine which input regions actually influence the output. This produces more succinct visual explanations and more accurately exposes the network's behavior. We demonstrate the effectiveness of our model on three datasets totaling 16 h of driving. We first show that training with attention does not degrade the performance of the end-to-end network. Then we show that the network highlights interpretable features that are used by humans while driving, and causal filtering achieves a useful reduction in explanation complexity by removing features which do not significantly affect the output.

Keywords Explainable AI · Self-driving vehicles · Visual attention

J. Kim (✉) · J. Canny
Department of Electrical Engineering and Computer Sciences, UC Berkeley, Berkeley, CA, USA
e-mail: jinkyu.kim@berkeley.edu; canny@berkeley.edu

H. J. Escalante et al. (eds.), *Explainable and Interpretable Models in Computer Vision and Machine Learning*, The Springer Series on Challenges in Machine Learning, https://doi.org/10.1007/978-3-319-98131-4_8

1 Introduction

Self-driving vehicle control has made dramatic progress in the last several years, and many auto vendors have pledged large-scale commercialization in a 2–3 year time frame. These controllers use a variety of approaches but recent successes (Bojarski et al. 2016b) suggest that neural networks will be widely used in self-driving vehicles. Deep neural networks have been shown to be an effective tool (Bojarski et al. 2016b; Xu et al. 2017) to learn vehicle controls for self-driving cars in an end-to-end manner. Despite their effectiveness as a function estimator, DNNs operate as a black-box—both network architecture and hidden layer activations may have no obvious relation to the function being estimated by the network.

To allow end-users understand what has triggered a particular behavior, hence to increase trust, these models need to be self-explanatory. There exist two main types of philosophical argument for explanations (Akata et al. 2018):

(i) **Introspective explanations:** A system is introspective through a series of understandable ways (i.e., Bob explains Bob's actions).
(ii) **Rationalizations:** We want to justify or rationalize the system through a series of logically consistent and understandable choices that can correlate model response with physical observations (i.e., Alice watching a video of Bob, and then asking Alice to justify Bob's actions).

One way of achieving introspection is via visual attention mechanisms (Xu et al. 2015; Kim and Canny 2017). Visual attention filters out non-salient image regions, hence the model visually fixates on important image content that is relevant to the decision. These networks provide spatial attention maps—areas of the image that the network attends to—that can be displayed in a way that is easy for users to interpret. They provide their attention maps instantly on images that are input to the network, and in this case on the stream of images from automobile video. Providing visual attention to the user as a justification of a decision increases trust. As we show from our examples later, visual attention maps lie over image areas that have an intuitive influence on the vehicle's control signal. Further, we show that state-of-the-art driving models can be made interpretable without sacrificing accuracy, that attention models provide more robust image annotation, and causal analysis further improves explanation saliency.

But attention maps are only part of the story. Attention is a mechanism for filtering out non-salient image content. But attention networks need to find all *potentially* salient image areas and pass them to the main recognition network (a CNN here) for a final verdict. For instance, the attention network will attend to trees and bushes in areas of an image where road signs commonly occur. Just as a human will use peripheral vision to determine that "there is something there", and then visually fixate on the item to determine what it actually is. We, therefore, post-process the attention network's output, clustering it into attention "blobs" and then

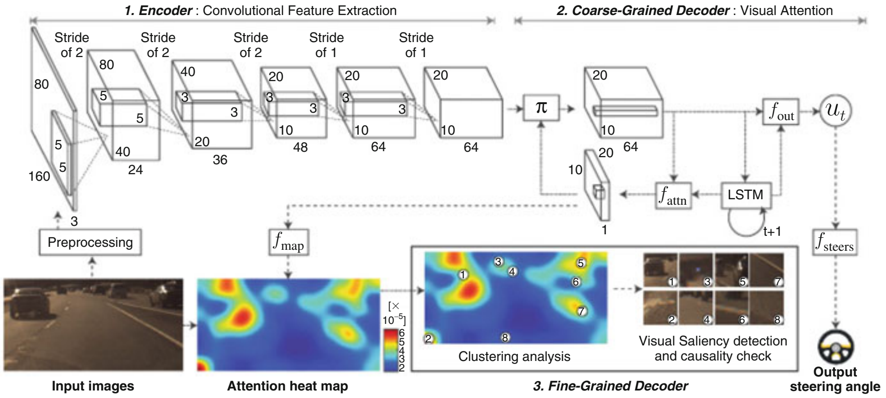


Fig. 1 Our model predicts steering angle commands from an input raw image stream in an end-to-end manner. In addition, our model generates a heat map of attention, which can visualize where and what the model sees. To this end, we first encode images with a CNN and decode this feature into a heat map of attention, which is also used to control a vehicle. We test its causality by scrutinizing each cluster of attention blobs and produce a refined attention heat map of causal visual saliency. Copyright ©2017 IEEE

mask (set the attention weights to zero) each blob to determine the effect on the end-to-end network output. Blobs that have a causal effect on network output are retained while those that do not are removed from the visual map presented to the user.

Figure 1 shows an overview of our model. Our approach can be divided into three steps: (1) Encoder: convolutional feature extraction (Sect. 3.2), (2) Coarse-grained decoder by visual attention mechanism (Sect. 3.3), and (3) Fine-grained decoder: causal visual saliency detection and refinement of attention map (Sect. 3.4).

Our contributions are summarized as follows:

- We show that visual attention heat maps are suitable “explanations” for the behavior of a deep neural vehicle controller, and do not degrade control accuracy.
- We show that attention maps comprise “blobs” that can be segmented and filtered to produce simpler and more accurate maps of visual saliency.
- We demonstrate the effectiveness of using our model with three large real-world driving datasets that contain over 1,200,000 video frames (*approx.* 16 h).
- We illustrate typical spurious attention sources in driving video and quantify the reduction in explanation complexity from causal filtering.

2 Related Work

2.1 End-to-End Learning for Self-Driving Cars

Self-driving vehicle control has made notable progress in the last several years. These controllers use a variety of approaches, which can mainly be divided in the following two types: (1) a mediated perception-based approach and (2) an end-to-end learning approach. The mediated perception-based approach depends on recognizing human-designated features, such as lane markings, pedestrians, or cars. These approaches mainly use a controller with if-then-else rules, which generally require demanding parameter tuning for achieving a balanced performance. Notable examples may include Urmson et al. (2008), Buehler et al. (2009), and Levinson et al. (2011).

ALVINN (Autonomous Land Vehicle In a Neural Network) (Pomerleau 1989) was the first attempt to use neural network for directly mapping images to navigate the direction of the vehicle. Recent success (Bojarski et al. 2016b) suggests that neural networks can be successfully applied to self-driving vehicle control in an end-to-end manner. Most of these approaches use a behavioral cloning model to learn a vehicle controller by supervised regression to demonstrations by human drivers. The training data comprise a stream of dash-cam images from one or more vehicle cameras, and the control outputs (i.e., steering, acceleration, and braking) from the driver.

Bojarski et al. (2016b) used a deep neural network to directly map a stream of front-view dashcam images to steering controls. Xu et al. (2017) collected a large crowd-sourced driving dataset, and predicted a sequence of discretized vehicle's future ego-motion (i.e., go straight, stop, left-turn, and right turn) given a series of dashcam images and prior vehicle states, i.e., speed. Fernando et al. (2017) presented a driving model that uses images and the steering wheel trajectory so that the model gains a long-term planning capacity via neural memory networks. Similarly, Chi and Mu (2017) proposed a Conv-LSTM framework to efficiently utilize the spatio-temporal information to model a stateful process. These models show good performance but their behavior is opaque and uninterpretable.

Chen et al. (2015) explored an intermediate approach by defining human interpretable features (i.e., the curvature of lane, distances to neighboring lane markings, and distance from the front-located vehicle) and training a CNN to predict these features. A simple vehicle controller is then used to map these features to steering angle commands. They also generated deconvolution maps to show image areas that affected network output. However, there were several difficulties with that work: (1) use of the intermediate layer may cause significant degradation of control accuracy and (2) the intermediate feature descriptors provide a limited and ad-hoc vocabulary for explanations.

2.2 Visual Explanations

Zeiler and Fergus (2014) proposed a landmark work by utilizing a deconvolution-style approach to visualize layer activations of convolutional neural networks. LeCun et al. (2015) trained a language model to automatically generate captions as textual explanations of images. Building on this work, Bojarski et al. (2016a) developed a richer notion of the contribution of a pixel to the output. However, there still remains a difficulty: the lack of formal measures of how the network output is affected by spatially-extended features rather than pixels. For an image classification problem, Hendricks et al. (2016) trains a deep network to generate specific explanation without explicitly identifying semantic features. Johnson et al. (2016) proposes DenseCap which uses fully convolutional localization networks for dense captioning, their paper achieves both localizing objects and describing salient regions in images using natural language. In reinforcement learning, Zahavy et al. (2016) proposes a visualization method to interpret the agent's action by describing Markov Decision Process model as a directed graph on a t-SNE map.

3 Attention-Based Explainable Deep Driving Model

As we depicted in Fig. 1, our model predicts continuous steering angle commands from input raw images end-to-end. Our model can be divided into three steps: (1) Encoder: convolutional feature extraction (Sect. 3.2) (2) Coarse-grained decoder by visual attention mechanism (Sect. 3.3), and (3) Fine-grained decoder: causal visual saliency detection and refinement of attention maps (Sect. 3.4).

3.1 Preprocessing

As discussed by Bojarski et al. (2016b), steering angle commands depend on the vehicle's specific steering geometry, and thus our model predicts the curvature $\hat{u}_t$ (= r_t^{-1}), which is defined to be the reciprocal of the turning radius r_t. The relationship between the inverse turning radius u_t and the steering angle command θ_t can be approximated by Ackermann steering geometry (Rajamani 2011) as follows:

$$\theta_t = f_{steers}(u_t) = u_t d_w K_s (1 + {v_t}^2 K_{slip}) \tag{1}$$

where θ_t in degrees and v_t (m/s) is a steering angle and a velocity at time t, respectively. K_s, K_{slip}, and d_w are vehicle-specific parameters. K_s is a steering ratio between the turn of the steering and the turn of the wheels. K_{slip} represents the relative motion between a wheel and the surface of road. d_w is the length between the front and rear wheels. Our model therefore needs two measurements for training: timestamped vehicle's speed and steering angle commands.

To reduce computational cost, each raw input image is down-sampled and resized to 80×160×3 with nearest-neighbor scaling algorithm. For images with different raw aspect ratios, we cropped the height to match the ratio before down-sampling. A common practice in image classification is to subtract the mean RGB value computed on the training set from each pixel (Simonyan and Zisserman 2015). This is effective to achieve zero-centered inputs which are originally in different scales. Driving datasets, however, do not show that various scales. For instance, the camera gains are (automatically or in advance) calibrated to capture such high-quality images in a certain dynamic range. In our experiment, we could not obtain significant improvement by the use of mean subtraction. Instead, we change the range of pixel intensity values and convert to HSV colorspace, which is commonly used for its robustness in problems where color description plays an integral role.

We utilize a single exponential smoothing method (Hyndman et al. 2008) to reduce the effect of human factors-related performance variation and the effect of measurement noise. Formally, given a smoothing factor $0 \leq \alpha_\mathrm{s} \leq 1$, the simple exponential smoothing method is defined as follows:

$$\begin{pmatrix} \hat{\theta}_t \\ \hat{v}_t \end{pmatrix} = \alpha_s \begin{pmatrix} \theta_t \\ v_t \end{pmatrix} + (1 - \alpha_s) \begin{pmatrix} \hat{\theta}_{t-1} \\ \hat{v}_{t-1} \end{pmatrix} \tag{2}$$

where $\hat{\theta}_t$ and $\hat{v}_t$ are the smoothed time-series of θ_t and v_t, respectively. Note that they are same as the original time-series when $\alpha_\mathrm{s} = 1$, while values of α_s closer to zero have a greater smoothing effect and are less responsive to recent changes. The effect of applying smoothing methods is summarized in Sect. 4.4.

3.2 Encoder: Convolutional Feature Extraction

We use a 5-layered convolutional neural network to extract a set of encoded visual feature vector, which we refer to as a convolutional feature cube $\mathbf{X}_t$. Each feature vectors may contain high-level object descriptions that allow the attention model to selectively pay attention to certain parts of an input image by choosing a subset of feature vectors.

As depicted in Fig. 1, we use a 5-layered convolution network that is utilized by Bojarski et al. (2016b) to learn a model for self-driving cars. As discussed by Lee et al. (2009), we omit max-pooling layers to prevent spatial locational information loss as the strongest activation propagates through the model. We collect a three-dimensional convolutional feature cube $\mathbf{X}_t$ from the last layer by pushing the preprocessed image through the model, and the output feature cube will be used as an input of the LSTM layers, which we will explain in Sect. 3.3. Using this convolutional feature cube from the last layer has advantages in generating high-level object descriptions, thus increasing interpretability and reducing computational burdens for a real-time system.

Formally, a convolutional feature cube of size $w \times h \times d$ is created at each timestep t from the last convolutional layer. We then collect $\mathbf{X}_t$, a set of $l = w \times h$

vectors, each of which is a d-dimensional feature slice for different spatial parts of the given input.

$$\mathbf{X}_t = \{\mathbf{x}_{t,1}, \mathbf{x}_{t,2}, \ldots, \mathbf{x}_{t,l}\} \tag{3}$$

where $\mathbf{x}_{t,i} \in \mathbb{R}^d$ for $i \in \{1, 2, \ldots, l\}$. This allows us to focus selectively on different spatial parts of the given image by choosing a subset of these l feature vectors.

3.3 Coarse-Grained Decoder: Visual (Spatial) Attention

The goal of soft deterministic attention mechanism $\pi(\{\mathbf{x}_{t,1}, \mathbf{x}_{t,2}, \ldots, \mathbf{x}_{t,l}\})$ is to search for a good context vector $\mathbf{y}_t$, which is defined as a combination of convolutional feature vectors $\mathbf{x}_{t,i}$ for $i \in \{1, 2, \ldots, l\}$, while producing better prediction accuracy. We utilize a deterministic soft attention mechanism that is trainable by standard back-propagation methods, which thus has advantages over a hard stochastic attention mechanism that requires reinforcement learning approaches. Our model feeds α weighted context $\mathbf{y}_t$ to the system as discuss by several works (Sharma et al. 2015; Xu et al. 2015):

$$\begin{aligned} \mathbf{y}_t &= f_{\text{flatten}}(\pi(\{\alpha_{t,i}\}, \{\mathbf{x}_{t,i}\})) \\ &= f_{\text{flatten}}(\{\alpha_{t,i}\mathbf{x}_{t,i}\}) \end{aligned} \tag{4}$$

where $i = \{1, 2, \ldots, l\}$. $\alpha_{t,i}$ is a scalar attention weight value associated with a certain grid of input image in such that $\sum_i \alpha_{t,i} = 1$. These attention weights can be interpreted as the probability over l convolutional feature vectors that the location i is the important part to produce better prediction accuracy. f_{flatten} is a flattening function, which converts the input feature matrix into a 1-D feature vector to be used by the dense layer for LSTM. $\mathbf{y}_t$ is thus $d \times l$-dimensional vector that contains convolutional feature vectors weighted by attention weights. Note that our attention mechanism $\pi(\{\alpha_{t,i}\}, \{\mathbf{x}_{t,i}\})$ is different from the previous works (Sharma et al. 2015; Xu et al. 2015), which use the α weighted average context $\mathbf{y}_t = \sum_{i=1}^{l} \alpha_{t,i}\mathbf{x}_{t,i}$. We observed that this change significantly improves overall prediction accuracy. The performance comparison is explained in Sect. 4.5.

Long Short-Term Memory (LSTM) As we summarize in Fig. 1, we use a long short-term memory (LSTM) network (Hochreiter and Schmidhuber 1997) that predicts the inverse turning radius $\hat{u}_t$ and generates attention weights $\{\alpha_{t,i}\}$ at each timestep t conditioned on the previous hidden state $\mathbf{h}_{t-1}$ and a current convolutional feature cube $\mathbf{x}_t$. The LSTM is defined as follows:

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} sigm \\ sigm \\ sigm \\ tanh \end{pmatrix} \mathbf{A} \begin{pmatrix} \mathbf{h}_{t-1} \\ \mathbf{y}_t \end{pmatrix} \tag{5}$$

where $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$, and $\mathbf{c}_t \in \mathbb{R}^{\mathrm{M}}$ are the M-dimensional input, forget, output, memory state of the LSTM at time t, respectively. Internal states of the LSTM are computed conditioned on the hidden state $\mathbf{h}_t \in \mathbb{R}^{\mathrm{M}}$ and an α-weighted context vector $\mathbf{y}_t \in \mathbb{R}^d$. We use an affine transformation $\mathbf{A} : \mathbb{R}^{d+\mathrm{M}} \to \mathbb{R}^{4\mathrm{M}}$. The logistic sigmoid activation function and the hyperbolic tangent activation function are represented as *sigm* and *tanh*, respectively. The hidden state $\mathbf{h}_t$ and the cell state $\mathbf{c}_t$ of the LSTM are defined as:

$$\begin{aligned} \mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \\ \mathbf{h}_t &= \mathbf{o}_t \odot \mathit{tanh}(\mathbf{c}_t) \end{aligned} \tag{6}$$

where $\odot$ is element-wise multiplication.

Attention We use an additional hidden layer, denoted by $f_{\mathrm{attn}}(\mathbf{x}_{t,i}, h_{t-1})$, which is conditioned on the previous LSTM state $\mathbf{h}_{t-1}$, and the current feature vectors $\mathbf{x}_{t,i}$. Then, we use multinomial logistic regression (i.e., softmax regression) function to obtain the attention weight $\{\alpha_{t,i}\}$ as follows:

$$f_{\mathrm{attn}}(\mathbf{x}_{t,i}, \mathbf{h}_{t-1}) = \mathbf{W}_{\mathrm{a}}(\mathbf{W}_{\mathrm{x}}\mathbf{x}_{t,i} + \mathbf{W}_{\mathrm{h}}\mathbf{h}_{t-1} + \mathbf{b}_{\mathrm{a}}) \tag{7}$$

where $\mathbf{W}_{\mathrm{a}} \in \mathbb{R}^d$, $\mathbf{W}_{\mathrm{x}} \in \mathbb{R}^{d\times d}$, and $\mathbf{W}_{\mathrm{h}} \in \mathbb{R}^{d\times M}$, which are learned parameters. The attention weight $\alpha_{t,i}$ for each spatial location i is then computed by multinomial logistic regression (i.e., softmax regression) function as follows:

$$\alpha_{t,i} = \frac{\exp(f_{\mathrm{attn}}(\mathbf{x}_{t,i}, \mathbf{h}_{t-1}))}{\sum_{j=1}^{l} \exp(f_{\mathrm{attn}}(\mathbf{x}_{t,j}, \mathbf{h}_{t-1}))} \tag{8}$$

Initialization To initialize memory state $\mathbf{c}_t$ and hidden state $\mathbf{h}_t$ of the LSTM, we use average of the convolutional feature slices $\mathbf{x}_{0,i} \in \mathbb{R}^d$ for $i \in \{0, 1, \ldots, l\}$ and feed through two additional hidden layers: $f_{\mathrm{init},c}$ and $f_{\mathrm{init},h}$.

$$\mathbf{c}_0 = f_{\mathrm{init},c}\left(\frac{1}{l}\sum_{i=1}^{l}\mathbf{x}_{0,i}\right), \quad \mathbf{h}_0 = f_{\mathrm{init},h}\left(\frac{1}{l}\sum_{i=1}^{l}\mathbf{x}_{0,i}\right) \tag{9}$$

Output The output of the vehicle controller is vehicle's inverse turning radius $\hat{u}_t$. We use additional hidden layer, denoted by $f_{\mathrm{out}}(\mathbf{y}_t, \mathbf{h}_t)$, which are conditioned on the current hidden state $\mathbf{h}_t$ and the spatially-attended context $\mathbf{y}_t$.

$$\begin{aligned} \hat{u}_t &= f_{\mathrm{out}}(\mathbf{y}_t, \mathbf{h}_t) \\ &= \mathbf{W}_{\mathrm{U}}(\mathbf{W}_{\mathrm{Y}}\mathbf{y}_t + \mathbf{W}_{\mathrm{h}}\mathbf{h}_t) \end{aligned} \tag{10}$$

where $\mathbf{W}_{\mathrm{U}} \in \mathbb{R}^d$, $\mathbf{W}_{\mathrm{Y}} \in \mathbb{R}^{d\times d}$, $\mathbf{W}_{\mathrm{h}} \in \mathbb{R}^{d\times M}$, which are learned parameters.

Loss Function and Regularization As discussed by Xu et al. (2015), doubly stochastic regularization can encourage the attention model to at different parts of the image. At each timestep t, our attention model predicts a scalar β_t=$sigm(f_\beta(\mathbf{h}_{t-1}))$ with an additional hidden layer f_β conditioned on the previous hidden state $\mathbf{h}_{t-1}$ such that

$$\mathbf{y}_t = sigm(f_\beta(\mathbf{h}_{t-1})) f_{\text{flatten}}(\{\alpha_{t,i}\mathbf{x}_{t,i}\}) \tag{11}$$

Concretely, we use the following penalized loss function $\mathscr{L}_1$:

$$\mathscr{L}_1(u_t, \hat{u}_t) = \sum_{t=1}^{T} |u_t - \hat{u}_t| + \lambda \sum_{i=1}^{L} \left(1 - \sum_{t=1}^{T} \alpha_{t,i}\right) \tag{12}$$

where T is the length of time steps, and λ is a penalty coefficient that encourages the attention model to see different parts of the image at each time frame. Section 4.3 describes the effect of using regularization.

3.4 Fine-Grained Decoder: Causality Test

Image-Level Masking Approach The last step of our pipeline is a fine-grained decoder, in which we refine a map of attention and detect local visual saliencies. Though an attention map from our coarse-grained decoder provides probability of importance over a 2D image space, our model needs to determine specific regions that cause a causal effect on prediction performance. To this end, we assess a decrease in performance when a local visual saliency on an input raw image is masked out.

We first collect a consecutive set of attention weights $\{\alpha_{t,i}\}$ and input raw images $\{I_t\}$ for a user-specified T timesteps. We then create a map of attention, which we refer M_t as defined: $M_t = f_{\text{map}}(\{\alpha_{t,i}\})$. Our 5-layer convolutional neural network uses a stack of 5 × 5 and 3 × 3 filters without any pooling layer, and therefore the input image of size 80 × 160 is processed to produce the output feature cube of size 10 × 20 × 64, while preserving its aspect ratio. Thus, we use $f_{\text{map}}(\{\alpha_{t,i}\})$ as up-sampling function by the factor of eight followed by Gaussian filtering (Burt and Adelson 1983) as discussed in Xu et al. (2015) (see Fig. 2a, b).

To extract a local visual saliency, we first randomly sample 2D N particles with replacement over an input raw image conditioned on the attention map M_t. Note that, we also use time-axis as the third dimension to consider temporal features of visual saliencies. We thus store spatio-temporal 3D particles $P \leftarrow P \cup \{P_t, t\}$ (see Fig. 2c).

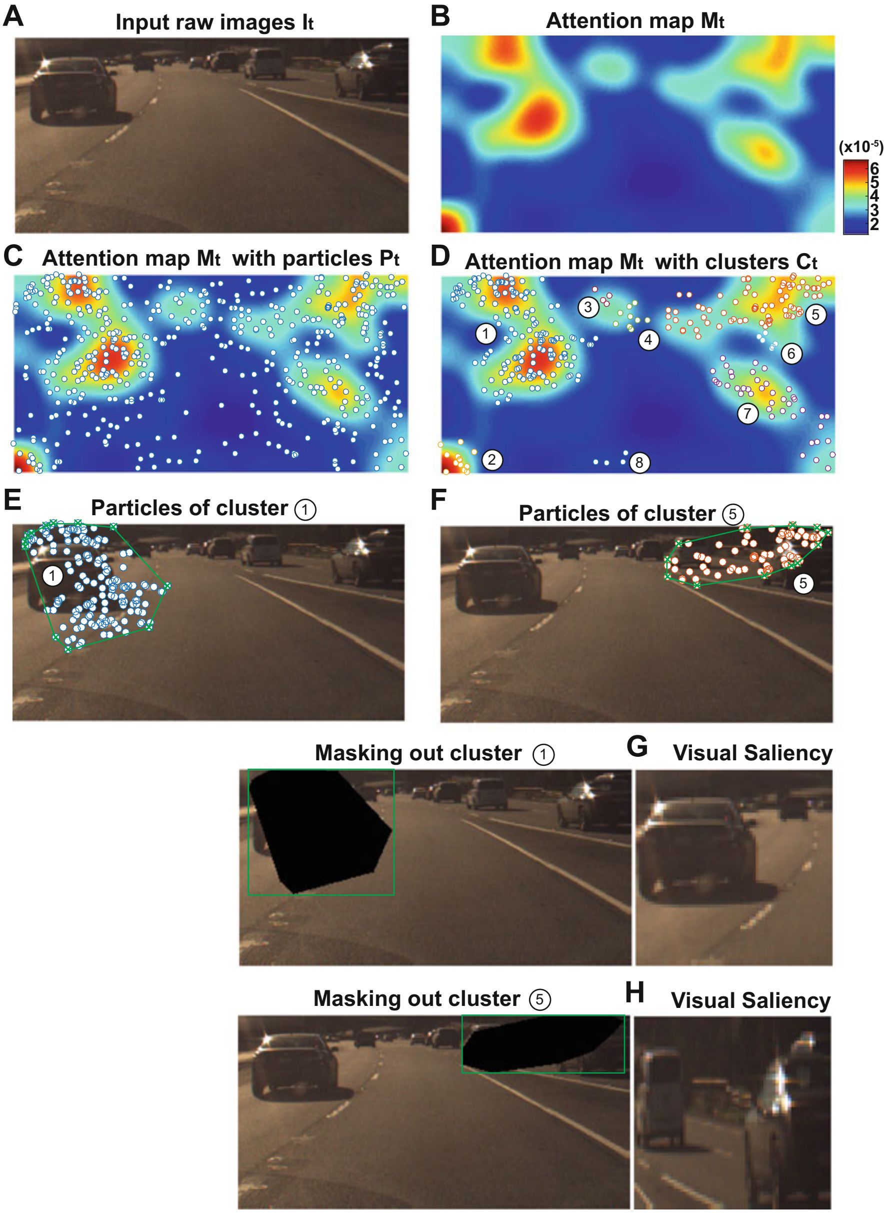


Fig. 2 Overview of our fine-grained decoder. Given an input raw pixels I_t (**a**), we compute an attention map M_t with a function f_{map} (**b**). (**c**) We randomly sample 3D $N = 500$ particles over the attention map, and (**d**) we apply a density-based clustering algorithm (DBSCAN (Ester et al. 1996)) to find a local visual saliency by grouping particles into clusters. (**e**, **f**) For each cluster $c \in C$, we compute a convex hull $H(c)$ to define its region, and mask out the visual saliency to see causal effects on prediction accuracy (see **e**, **f** for clusters 1 and 5, respectively). (**g**, **h**) Warped visual saliencies for clusters 1 and 5, respectively. Copyright ©2017 IEEE

```
Algorithm 1: Fine-grained decoder: causality check
  Data: A consecutive set of $\{u_t\}$ and images $\{I_t\}$ and
  Result: A set of visual saliencies $S$
  Get a set of $\{\alpha_{t,i}\}$ and prediction errors $\{\varepsilon_t\}$ by running Encoder and Decoder for all images $\{I_t\}$;
  $P \leftarrow \phi$, $S \leftarrow \phi$;
  for t = 0 to T-1 do
      Get a 2D attention map $M_t = f_{\text{map}}(\{\alpha_{t,i}\})$;
      Get a set $P_t$ of randomly sampled 2D $N$ points conditioned on $M_t$;
      Aggregate datasets: $P \leftarrow P \cup \{P_t, \text{t}\}$;
  end
  Run clustering algorithm on $P$ and get clusters $\{C_t\}$;
  for t = 0 to T-1 do
      for each cluster c $\in C_t$ do
          Get a convex hull $H(c)$;
          Masking out pixels on $H(c)$ from $I_t$;
          Get a new prediction error $\hat{\varepsilon}_t$;
          if $|\hat{\varepsilon}_t - \varepsilon_t| > \delta$ then
              Aggregate saliency $S \leftarrow S \cup H(c)$;
          end
      end
  end
```

We then apply a clustering algorithm to find a local visual saliency by grouping 3D particles P into clusters $\{C\}$ (see Fig. 2d). In our experiment, we use DBSCAN (Ester et al. 1996), a density-based clustering algorithm that has advantages to deal with a noisy dataset because they group particles together that are closely packed, while marking particles as outliers that lie alone in low-density regions. For points of each cluster c and each time frame t, we compute a convex hull $H(c)$ to find a local region of each visual saliency detected (see Fig. 2e, f).

For points of each cluster c and each time frame t, we iteratively measure a decrease of prediction performance with an input image which we mask out a local visual saliency. We compute a convex hull $H(c)$ to find a local, and mask out each visual saliency by assigning zero values for all pixels lying inside each convex hull. Each causal visual saliency is generated by warping into a fixed spatial resolution (=64×64) as shown in Fig. 2g, h. Algorithm 1 explains a pseudo-code for this step.

Feature-Level Masking Approach Along with devising the fine-grained decoder, we may consider using feature-level masking approach. Masking out convolutional features of attended region can provide a computationally efficient way by removing multiple forward passes of the convolutional network. This approach, however, may suffer from low deconvolutional spatial resolution, which makes challenge to view as a unit apart and divide the whole attention map into distinct attended objects, such as cars or lane markings.

4 Result

4.1 Datasets

As explained in Table 1, we obtain two large-scale datasets that contain over 1,200,000 frames (≈16 h) collected from Comma.ai (2017), Udacity (2017), and Hyundai Center of Excellence in Integrated Vehicle Safety Systems and Control (HCE) under a research contract. These three datasets acquired contain video clips captured by a single front-view camera mounted behind the windshield of the vehicle. Alongside the video data, a set of time-stamped sensor measurement is contained, such as vehicle's velocity, acceleration, steering angle, GPS location and gyroscope angles. Thus, these datasets are ideal for self-driving studies. Note that, for sensor logs unsynced with the time-stamps of video data, we use the estimates of the interpolated measurements. Videos are mostly captured during highway driving in clear weather on daytime, and there included driving on other road types, such as residential roads (with and without lane markings), and contains the whole driver's activities such as staying in a lane and switching lanes. Note also that, we exclude frames when the vehicle stops which happens when $\hat{v}_t < 1$ m/s.

4.2 Training and Evaluation Details

To obtain a convolutional feature cube $\mathbf{x}_t$, we train the 5-layer CNNs explained in Sect. 3.2 by using additional 5-layer fully connected layers (i.e., # hidden variables: 1164, 100, 50, and 10, respectively), of which output predicts the measured inverse turning radius u_t. Incidentally, instead of using addition fully-connected layers, we could also obtain a convolutional feature cube $\mathbf{x}_t$ by training from scratch with the whole network. In our experiment, we obtain the 10×20×64-dimensional

Table 1 Dataset details

	Datasets		
	Comma.ai (2017)	HCE	Udacity (2017)
# Frame	522,434	80,180	650,690
FPS	20 Hz	20 Hz/30 Hz	20 Hz
Hours	≈7 h	≈1 h	≈8 h
Condition	Highway/Urban	Highway	Urban
Location	Bay area, USA	Bay area, USA	Bay area, USA
Lighting	Day/Night	Day	Day

Over 16 h (>1,200,000 video frames) of driving dataset that contains a front-view video frames and corresponding time-stamped measurements of vehicle dynamics. The data is collected from two public data sources and one private data source, Comma.ai (2017) and Udacity (2017), and Hyundai Center of Excellence in Vehicle Dynamic Systems and Control (HCE). Copyright ©2017 IEEE convolutional feature cube, which is then flattened to 200×64 and is fed through the coarse-grained decoder. Other recent types of more recent expressive networks may give a performance boost over our CNN configuration. However, exploration of other convolutional architectures would be out of our scope.

We experiment with various numbers of LSTM layers (1–5) of the soft deterministic visual attention model but did not observe any significant improvements in model performance. Unless otherwise stated, we use a single LSTM layer in this experiment. For training, we use Adam optimization algorithm (Kinga and Adam 2015) and also use dropout (Srivastava et al. 2014) of 0.5 at hidden state connections and Xavier initialization (Glorot and Bengio 2010). We randomly sample a minibatch of size 128, each of batch contains a set Consecutive frames of length $T = 20$. Our model took less than 24 h to train on a single NVIDIA Titan X Pascal GPU. Our implementation is based on Tensorflow (Abadi et al. 2015) and code will be publicly available upon publication.

Two datasets (Comma.ai 2017 and HCE) we used were available with images captured by a single front-view camera. This makes it hard to use the data augmentation technique proposed by Bojarski et al. (2016b), which generated images with artificial shifts and rotations by using two additional off-center images (left-view and right-view) captured by the same vehicle. Data augmentation may give a performance boost, but we report performance without data augmentation.

4.3 Effect of Choosing Penalty Coefficient λ

Our model provides a better way to understand the rationale of the model's decision by visualizing where and what the model sees to control a vehicle. Sample attention maps are shown in Fig. 3. Figure 3 shows consecutive input raw images (with a sampling period of 5 s) and their corresponding attention maps. A tunable parameter λ controls closely the heat map matches the average value at that pixel. Setting λ to 0 encourages the model to pay attention to narrower parts of the image, while the model is encouraged to pay attention to wider parts of the image as we have larger λ. We experiment with three different penalty coefficients $\lambda \in \{0, 10, 20\}$ (see differences between the right three columns in Fig. 3). For better visualization, an attention map is overlaid by an input raw image and color-coded; for example, red parts represent where the model pays attention. We observe that our model is indeed able to pay attention to road elements, such as lane markings, guardrails, and vehicles ahead, which are essential for driving.

4.4 Effect of Varying Smoothing Factors

Recall from Sect. 3.1 that the single exponential smoothing method (Hyndman et al. 2008) is used to reduce the effect of human factors variation and the effect of

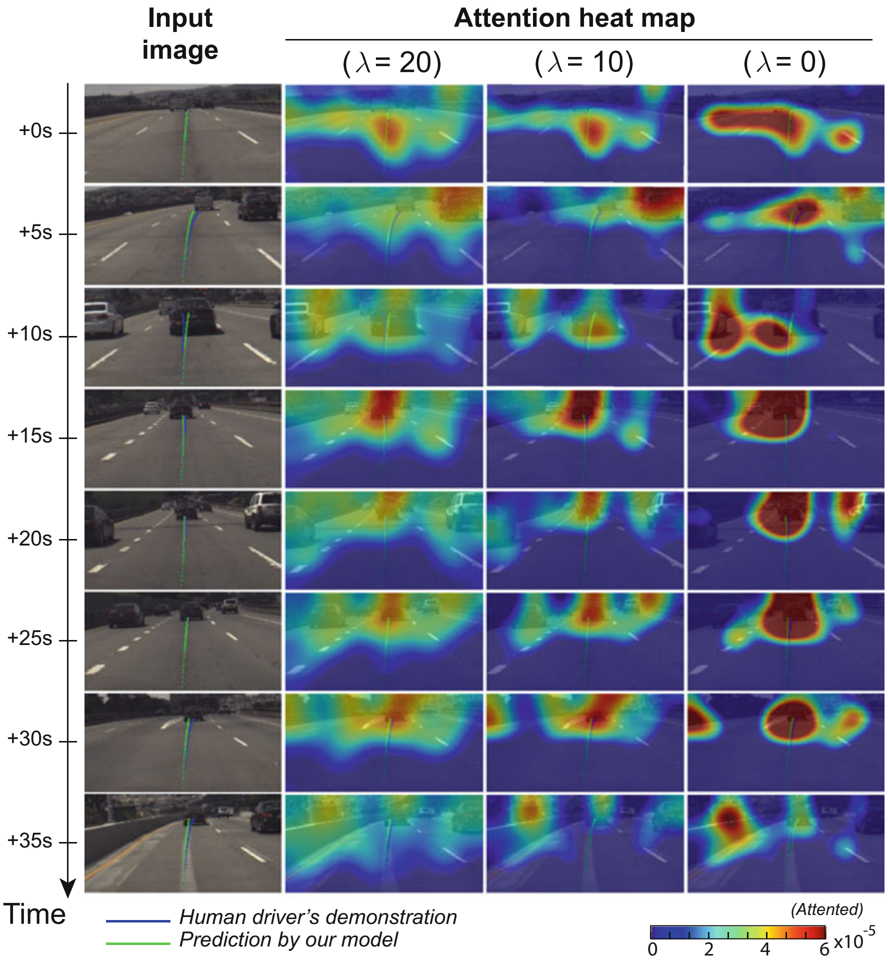


Fig. 3 Attention maps over time. Unseen consecutive input image frames are sampled at every 5 s (see from top to bottom). (first column) Input raw images with human driver's demonstrated curvature of path (blue line) and predicted curvature of path (green line). (From right) We illustrate attention maps with three different regularization penalty coefficients $\lambda \in \{0, 10, 20\}$. Each attention map is overlaid by an input raw image and color-coded. Red parts indicate where the model pays attention. *Data*: Comma.ai (2017). Adapted from Kim and Canny (2017)

measurement noise for two sensor inputs: steering angle and velocity. A robust model for autonomous vehicles would yield consistent performance, even when some measurements are noisy. Figure 4 shows the prediction performance in terms of mean absolute error (MAE) on a comma.ai testing data set. Various smoothing factors $\alpha_s \in \{0.01, 0.05, 0.1, 0.3, 0.5, 1.0\}$ are used to assess the effect of using smoothing methods. With setting α_s=0.05, our model for the task of steering estimation performs the best. Unless otherwise stated, we will use α_s as 0.05.

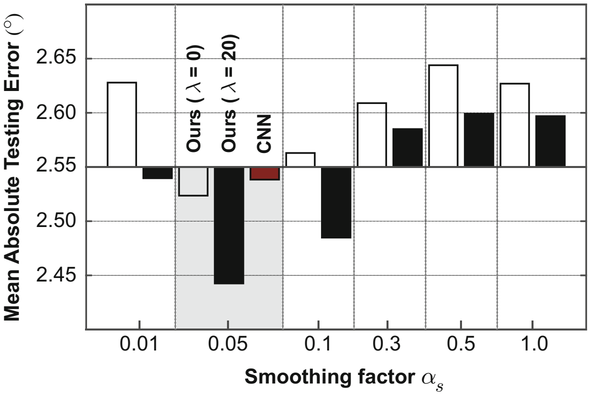


Fig. 4 Effect of applying a single exponential smoothing method over various smoothing factors from 0.1 to 1.0. We use two different penalty coefficients λ ∈ {0, 20}. With setting $\alpha_s = 0.05$, our model performs the best. *Data*: Comma.ai (2017). Copyright ©2017 IEEE

4.5 Quantitative Analysis

In Table 2, we compare the prediction performance with alternatives in terms of MAE. We implement alternatives that include the work by Bojarski et al. (2016b), which used an identical base CNN and a fully-connected network (FCN) without attention. To see the contribution of LSTMs, we also test a CNN and LSTM, which is identical to ours but does not use the attention mechanism. For our model, we test with three different values of penalty coefficients λ ∈ {0, 10, 20}.

Our model shows competitive prediction performance than alternatives. Our model shows 1.18–4.15 in terms of mean absolute error (MAE) on testing dataset. This confirms that incorporation of attention does not degrade control accuracy. While our model shows comparable prediction performance, it also provides an additional layer of interpretability by visualizing where and what the model sees. The average run-time for our model and alternatives took less than a day to train each dataset.

4.6 Effect of Causal Visual Saliencies

Recall from Sect. 3.4, we post-process the attention network's output by clustering it into attention blobs and filtering if they have an causal effect on network output. Figure 5a shows typical examples of an input raw image (leftmost column), an attention networks' outputs with spurious attention sources, and our refined attention heat maps (rightmost column). We observe our model can produce a simpler and more accurate map of visual saliency by filtering out spurious attention blobs. In our experiment, 62% and 58% out of all attention blobs are indeed spurious

Table 2 Control performance comparison in terms of mean absolute error (MAE) in degree and its standard deviation

		MAE in degree [SD]	
Dataset	Model	Training	Testing
Comma.ai (2017)	CNN+FCN (Bojarski et al. 2016b)	0.421 [0.82]	2.54 [3.19]
	CNN+LSTM	0.488 [1.29]	2.58 [3.44]
	Ours (λ=0)	0.497 [1.32]	2.52 [3.25]
	Ours (λ=10)	0.464 [1.29]	2.56 [3.51]
	Ours (λ=20)	0.463 [1.24]	2.44 [3.20]
HCE	CNN+FCN (Bojarski et al. 2016b)	0.246 [0.400]	1.27 [1.57]
	CNN+LSTM	0.568 [0.977]	1.57 [2.27]
	Ours (λ=0)	0.334 [0.766]	1.18 [1.66]
	Ours (λ=10)	0.358 [0.728]	1.25 [1.79]
	Ours (λ=20)	0.373 [0.724]	1.20 [1.66]
Udacity (2017)	CNN+FCN (Bojarski et al. 2016b)	0.457 [0.870]	4.12 [4.83]
	CNN+LSTM	0.481 [1.24]	4.15 [4.93]
	Ours (λ=0)	0.491 [1.20]	4.15 [4.93]
	Ours (λ=10)	0.489 [1.19]	4.17 [4.96]
	Ours (λ=20)	0.489 [1.26]	4.19 [4.93]

Control accuracy is not degraded by incorporation of attention compared to an identical base CNN without attention. *Abbreviation:* SD (standard deviation). Copyright ©2017 IEEE

attention sources on Comma.ai (2017) and HCE datasets (see Fig. 5b). We provide additional example sets in Fig. 6 and examples of detected causal visual saliencies in Fig. 7.

5 Discussion

The proposed method highlights regions that causally influence deep neural perception and control networks for self-driving cars. Thus, it would be worth exploring a potential overlap between the causally salient image areas and what and where human drivers is really paying their attention while driving.

Human driver's gaze may provide important visual cues for driving, including salient objects or regions that should attract human driver's attention. Driver's gaze behavior has been studied, and recently neural networks are trained end-to-end to estimate what and where a human pays attention while driving. Alletto et al. (2016) and Palazzi et al. (2017) collected a large human gaze annotation dataset while driving, called DR(eye)VE, and successfully showed a neural network can be applied to learn human gaze behavior. A number of approaches have been

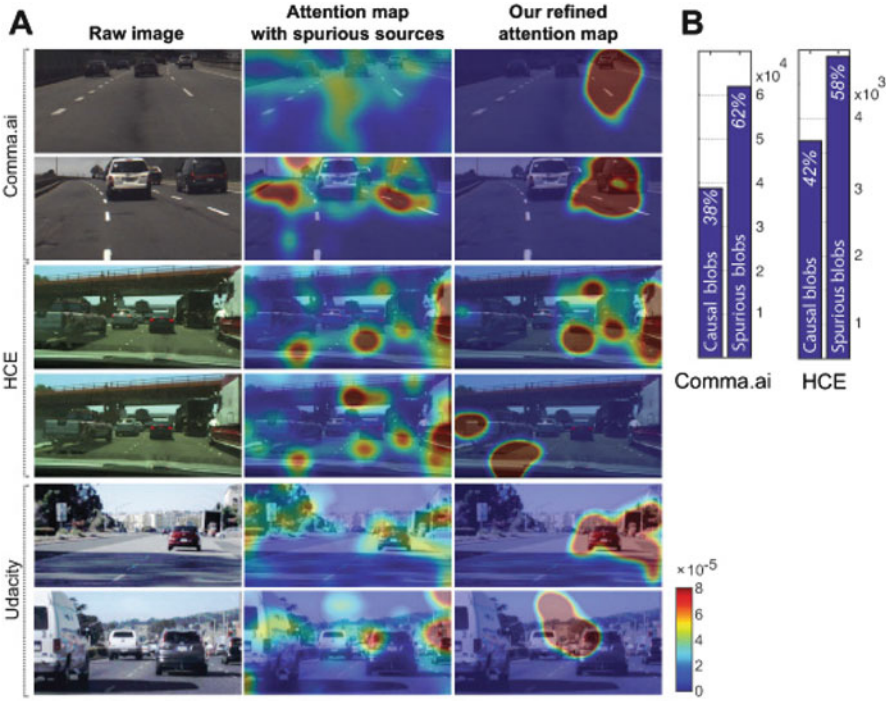


Fig. 5 (**a**) We illustrate examples of (left) raw input images, their (middle) visual attention heat maps with spurious attention sources, and (right) our attention heat maps by filtering out spurious blobs to produce simpler and more accurate attention maps. (**b**) To measure how much the causal filtering is simplifying attention clusters, we quantify the number of attention blobs before and after causal filtering. Copyright ©2017 IEEE

proposed for human gaze prediction in other applications. Most of these methods try to directly mimic the human gaze behavior by applying a supervised learning algorithm to learn a direct mapping from the images to the gazes. This literature is too wide to survey here, but some examples include Kümmerer et al. (2014), Liu et al. (2015), Bazzani et al. (2016), and Cornia et al. (2016).

Further, there have been few attempts to explore explicit incorporation of human gaze behavior for various applications, while humans gazes may provide important visual cues from human demonstration. Yu et al. (2017) achieved improved performance in the task of video captioning by explicitly using a predicted gaze heat map as an attention weight, i.e., supervising the attention model by human gaze prediction. This approach, however, inherently lacks an ability to implicitly explore other image regions—excluded from human gaze—during training, which may make the model sub-optimal. We leave this to a future work.

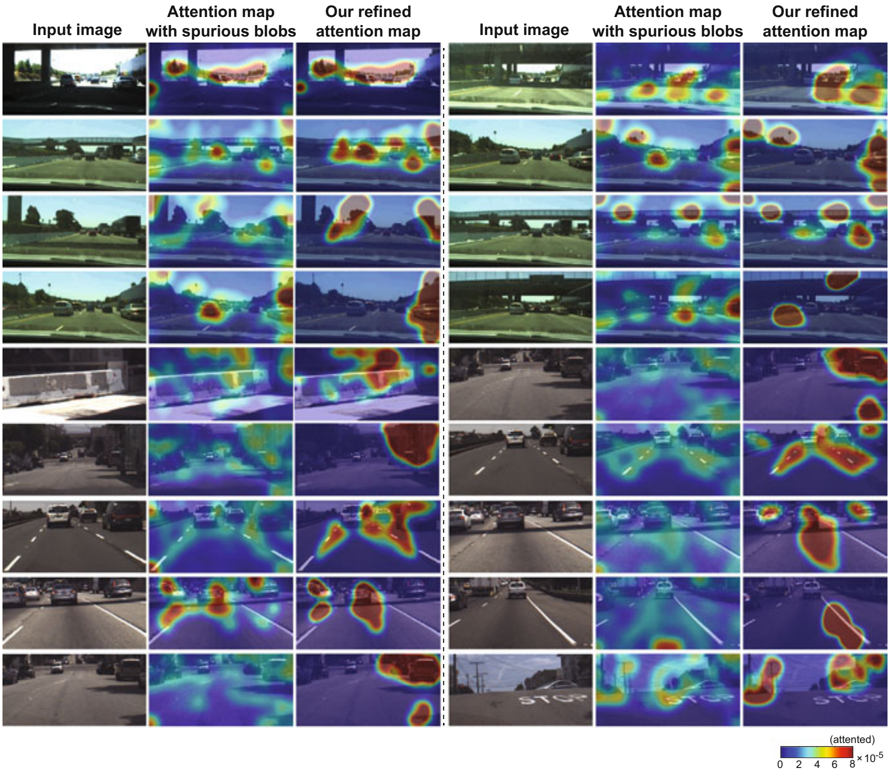
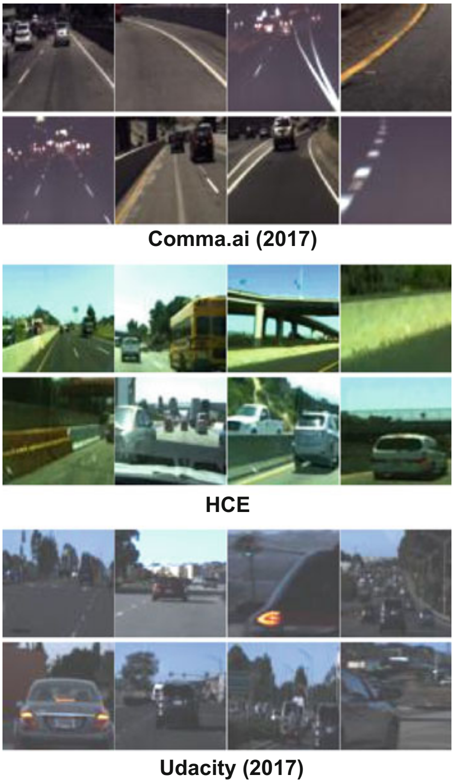

Input image
Attention map with spurious blobs
Our refined attention map
Input image
Attention map with spurious blobs
Our refined attention map
(attented)
0 2 4 6 8
×10-5

Fig. 6 We illustrate additional examples of (left) raw input images, their (middle) visual attention heat maps with spurious attention sources, and (right) our attention heat maps by filtering out spurious blobs to produce simpler and more accurate attention maps

6 Conclusion

We described an interpretable visualization for deep self-driving vehicle controllers. It uses a visual attention model augmented with an additional layer of causal filtering. We tested with three large-scale real driving datasets that contain over 16 h of video frames. We showed that (1) incorporation of attention does not degrade control accuracy compared to an identical base CNN without attention (2) raw attention highlights interpretable features in the image and (3) causal filtering achieves a useful reduction in explanation complexity by removing features which do not significantly affect the output.



Comma.ai (2017)

HCE

Udacity (2017)

Fig. 7 Examples of causal visual saliencies. For visualization, each image is resized to have a fixed size, i.e., 64×64×3

Acknowledgements This work was supported by Berkeley DeepDrive and Samsung Scholarship.

References

Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, et al (2015) Tensorflow: Large-scale machine learning on heterogeneous systems, 2015

Akata Z, Hendricks LA, Alaniz S, Darrell T (2018) Generating post-hoc rationales of deep visual classification decisions. Chapter in Explainable and Interpretable Models in Computer Vision and Machine Learning (The Springer Series on Challenges in Machine Learning)

Alletto S, Palazzi A, Solera F, Calderara S, Cucchiara R (2016) Dr (eye) ve: A dataset for attention-based tasks with applications to autonomous and assisted driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp 54–60

Bazzani L, Larochelle H, Torresani L (2016) Recurrent mixture density network for spatiotemporal visual attention. arXiv preprint arXiv:160308199

Bojarski M, Choromanska A, Choromanski K, Firner B, Jackel L, Muller U, Zieba K (2016a) Visualbackprop: visualizing cnns for autonomous driving. CoRR, vol abs/161105418

Bojarski M, Del Testa D, Dworakowski D, Firner B, Flepp B, Goyal P, Jackel LD, Monfort M, Muller U, Zhang J, et al (2016b) End to end learning for self-driving cars. CoRR abs/160407316

Buehler M, Iagnemma K, Singh S (2009) The DARPA urban challenge: autonomous vehicles in city traffic, vol 56. Springer

Burt P, Adelson E (1983) The laplacian pyramid as a compact image code. IEEE Transactions on communications 31(4):532–540

Chen C, Seff A, Kornhauser A, Xiao J (2015) Deepdriving: Learning affordance for direct perception in autonomous driving. In: Proceedings of the IEEE International Conference on Computer Vision, pp 2722–2730

Chi L, Mu Y (2017) Learning end-to-end driving model from spatial and temporal visual cues. Proceedings of the Workshop on Visual Analysis in Smart and Connected Communities

Commaai (2017) Public driving dataset. https://github.com/commaai/research, [Online; accessed 07-Mar-2017]

Cornia M, Baraldi L, Serra G, Cucchiara R (2016) Predicting human eye fixations via an lstm-based saliency attentive model. arXiv preprint arXiv:161109571

Ester M, Kriegel HP, Sander J, Xu X, et al (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd, vol 96, pp 226–231

Fernando T, Denman S, Sridharan S, Fookes C (2017) Going deeper: Autonomous steering with neural memory networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 214–221

Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Aistats, vol 9, pp 249–256

Hendricks LA, Akata Z, Rohrbach M, Donahue J, Schiele B, Darrell T (2016) Generating visual explanations. In: European Conference on Computer Vision, Springer, pp 3–19

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural computation 9(8): 1735–1780

Hyndman R, Koehler AB, Ord JK, Snyder RD (2008) Forecasting with exponential smoothing: the state space approach. Springer Science & Business Media

Johnson J, Karpathy A, Fei-Fei L (2016) Densecap: Fully convolutional localization networks for dense captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4565–4574

Kim J, Canny J (2017) Interpretable learning for self-driving cars by visualizing causal attention. Proceedings of the IEEE International Conference on Computer Vision

Kinga D, Adam JB (2015) A method for stochastic optimization. In: International Conference on Learning Representations

Kümmerer M, Theis L, Bethge M (2014) Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet. arXiv preprint arXiv:14111045

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444

Lee H, Grosse R, Ranganath R, Ng AY (2009) Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the 26th annual international conference on machine learning, ACM, pp 609–616

Levinson J, Askeland J, Becker J, Dolson J, Held D, Kammel S, Kolter JZ, Langer D, Pink O, Pratt V, et al (2011) Towards fully autonomous driving: Systems and algorithms. In: Intelligent Vehicles Symposium, IEEE, pp 163–168

Liu N, Han J, Zhang D, Wen S, Liu T (2015) Predicting eye fixations using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 362–370

Palazzi A, Abati D, Calderara S, Solera F, Cucchiara R (2017) Predicting the driver's focus of attention: the dr (eye) ve project. arXiv preprint arXiv:170503854

Pomerleau DA (1989) Alvinn, an autonomous land vehicle in a neural network. Tech. rep., Carnegie Mellon University, Computer Science Department

Rajamani R (2011) Vehicle dynamics and control. Springer Science & Business Media

Sharma S, Kiros R, Salakhutdinov R (2015) Action recognition using visual attention. arXiv preprint arXiv:151104119

Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations

Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15(1):1929–1958

Udacity (2017) Public driving dataset. https://www.udacity.com/self-driving-car, [Online; accessed 07-Mar-2017]

Urmson C, Anhalt J, Bagnell D, Baker C, Bittner R, Clark M, Dolan J, Duggins D, Galatali T, Geyer C, et al (2008) Autonomous driving in urban environments: Boss and the urban challenge. Journal of Field Robotics 25(8):425–466

Xu H, Gao Y, Yu F, Darrell T (2017) End-to-end learning of driving models from large-scale video datasets. Proceedings of the IEEE International Conference on Computer Vision

Xu K, Ba J, Kiros R, Cho K, Courville AC, Salakhutdinov R, Zemel RS, Bengio Y (2015) Show, attend and tell: Neural image caption generation with visual attention. In: ICML, vol 14, pp 77–81

Yu Y, Choi J, Kim Y, Yoo K, Lee SH, Kim G (2017) Supervising neural attention models for video captioning by human gaze data. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017). Honolulu, Hawaii, pp 2680–8

Zahavy T, Zrihem NB, Mannor S (2016) Graying the black box: Understanding dqns. arXiv preprint arXiv:160202658

Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: European Conference on Computer Vision, Springer, pp 818–833

Part IV
Explainability and Interpretability in First Impressions Analysis

Psychology Meets Machine Learning: Interdisciplinary Perspectives on Algorithmic Job Candidate Screening

Cynthia C. S. Liem, Markus Langer, Andrew Demetriou, Annemarie M. F. Hiemstra, Achmadnoer Sukma Wicaksana, Marise Ph. Born, and Cornelius J. König

Abstract In a rapidly digitizing world, machine learning algorithms are increasingly employed in scenarios that directly impact humans. This also is seen in job candidate screening. Data-driven candidate assessment is gaining interest, due to high scalability and more systematic assessment mechanisms. However, it will only be truly accepted and trusted if explainability and transparency can be guaranteed. The current chapter emerged from ongoing discussions between psychologists and computer scientists with machine learning interests, and discusses the job candidate screening problem from an interdisciplinary viewpoint. After introducing the general problem, we present a tutorial on common important methodological focus points in psychological and machine learning research. Following this, we both contrast and combine psychological and machine learning approaches, and present a use case example of a data-driven job candidate assessment system, intended to be explainable towards non-technical hiring specialists. In connection to this, we also give an overview of more traditional job candidate assessment approaches, and discuss considerations for optimizing the acceptability of technology-supported hiring solutions by relevant stakeholders. Finally, we present several recommendations on how interdisciplinary collaboration on the topic may be fostered.

C. C. S. Liem (✉) · A. Demetriou
Multimedia Computing Group, Delft University of Technology, Delft, The Netherlands
e-mail: c.c.s.liem@tudelft.nl; a.m.demetriou@tudelft.nl

M. Langer · C. J. König
Universität des Saarlandes, Saarbrücken, Germany
e-mail: markus.langer@uni-saarland.de; ckoenig@mx.uni-saarland.de

A. M. F. Hiemstra · M. Ph. Born
Erasmus School of Social and Behavioral Sciences, Erasmus University, Rotterdam, The Netherlands
e-mail: hiemstra@essb.eur.nl; m.ph.born@essb.eur.nl

Achmadnoer Sukma Wicaksana
Datasintesa Teknologi Nusantara, Jakarta, Indonesia

H. J. Escalante et al. (eds.), *Explainable and Interpretable Models in Computer Vision and Machine Learning*, The Springer Series on Challenges in Machine Learning, https://doi.org/10.1007/978-3-319-98131-4_9

Keywords Psychology · Machine learning · Job candidate screening · Methodology · Explainability · Multimodal analysis · Interdisciplinarity

1 Introduction: Algorithmic Opportunities for Job Candidate Screening

In a rapidly digitizing world, machine learning algorithms are increasingly employed to infer relevant patterns from data surrounding us as human beings. As a consequence, in many domains, information organization, process optimizations and predictions that formerly required human labor can now be systematically performed at higher efficiency and scalability.

The promise of computer-assisted decision-making has also entered the area of *personnel selection*: one of the oldest research areas within applied psychology. As early as in 1917, the problem of assessing whether candidates would be suitable for a job was recognized as:

> the Supreme Problem of diagnosing each individual, and steering him towards his fittest place, which is really the culminating problem of efficiency, because human capacities are after all the chief national resources. (Hall 1917)

This *job candidate screening* problem has been of interest to researchers and practitioners ever since (Ployhart et al. 2017). 100 years later, richer, multimodal and digital means of candidate presentation have become available, such as video resumes. Such presentation forms may offer more nuanced insight into a candidate; in comparison to paper resumes, ethnic minority applicants perceived digital video resumes as a fairer way of presentation (Hiemstra et al. 2012).

Digitization has not only influenced job candidate presentation forms, but also analysis techniques of candidate pools, through the inclusion of algorithmic methods in screening and selection procedures. This especially becomes necessary in case of large applicant pools, but is an actively debated practice. Proponents of automated digital selection methods argue that using algorithmic methods could lead to more diversity and empathetic workplaces, because they help to sidestep pitfalls typically associated with human decision-making. At the same time, caution is warranted because algorithms may be susceptible to bias in data and data labeling. Paradoxically, this bias may especially be harmful to applicants whose attributes are underrepresented in historical data (e.g., ethnic minorities).

1.1 The Need for Explainability

In technologically-assisted personnel selection, technological components replace parts of the selection procedure that formerly were conducted by humans. In alignment with emerging discussions on both fairness, accountability, transparency and ethics in machine learning and artificial intelligence, as well as human interpretability of sophisticated state-of-the-art machine learning models, research into explainability and transparency in algorithmic candidate screening is currently gaining interest (Escalante et al. 2017, 2018; Langer et al. 2018).

Considering technologically-assisted personnel selection, there are several reasons why explainability and transparency can be considered as particularly important:

- *Moral considerations.* Algorithmic decisions on personnel selection consider humans. It should be ensured that these decisions will not be unfair towards, or even harmful to certain population subgroups.
- *Knowledge-related considerations.* Hiring managers, the ultimate adopters of technologically-assisted selection tools, are not computer scientists. Therefore, they might not be able to develop algorithm-based solutions on their own, nor understand the development process towards an algorithm-based solution.

  Within machine learning, particularly through the advances of deep neural networks, very sophisticated and successful statistical models have emerged for performing predictions and classifications, but understanding and interpreting the internal workings of these networks is far from trivial.
- *Concerns about methodological soundness.* Increasingly, commercial ready-to-use solutions are being offered, and their inner workings may be a business secret. Still, regulatory frameworks such as the European General Data Protection Regulation (Council of the European Union 2016) may grant the explicit right to end users to demand transparency on how their information is processed.

  Furthermore, in practice, a *research-practitioner gap* is frequently observed in personnel selection: several methodologically sound personnel selection procedures and good-practice recommendations that are developed through research never get adopted by hiring managers (Anderson et al. 2001). For instance, there are psychometrically sound measures of personality (e.g., Big Five measures (McCrae and Costa 1999)). However, in practice, a large variety of unvalidated measures are used, that are more appealing to practitioners (Diekmann and König 2015). Some reasons might simply be that the unvalidated measure is easier to use, or that it appears more efficient and allows more control for hiring managers (König et al. 2010; Klehe 2004). We will discuss main reasons for acceptance and adoption in more detail in Sect. 5.

For all these reasons, calls for explainability and transparency connect to the concept of *trust*: we want to ensure that a potential technological solution 'does the right thing', without causing harm. At the same time, where to focus on when aiming to 'do the right thing' or 'tackling the most challenging aspect' is differently understood by different people. This is a common issue for domains in which multiple disciplines and stakeholders come together, as for example also noticed in the domain of music information retrieval (Liem et al. 2012). Deeper insight into different disciplinary viewpoints on the problem and the relationships between them—from shared interests to fundamental methodological differences—will have great impact on understanding what would be needed for technological solutions to become truly acceptable to everyone.

1.2 Purpose and Outline of the Chapter

The current chapter emerged from discussions between computer scientists and psychologists in the context of an ongoing collaboration on identifying future-proof skill sets and training resources on Big Data in Psychological Assessment.

Our discussions were inspired by the emerging societal and scientific interest in technological solutions for the personnel selection problem, but also by ongoing concrete data challenges on inferring first-impression personality and interviewability assessments from online video (Ponce-López et al. 2016; Escalante et al. 2017, 2018). These challenges relate to an overall mission "*to help both recruiters and job candidates by using automatic recommendations based on multi-media CVs.*" (Escalante et al. 2017). As a consequence, computer vision and machine learning researchers are challenged to not only quantitatively, but also qualitatively optimize their algorithmic prediction solutions.

In discussing potential data-driven solutions to these types of challenges, it became clear that the authors of this chapter indeed departed from different methodological focus points, interests, and optimization criteria. We therefore felt the need to more explicitly collect observations of how our various disciplinary viewpoints meet and differ. As a consequence, we contribute this chapter, which is meant as a tutorial which is accessible to computer scientists, psychologists and practitioners alike. Herein, we reflect on similarities and dissimilarities in disciplinary interests, potential common connection points, and practical considerations towards fostering acceptability of technologically-supported personnel selection solutions for various stakeholders, with special interest in questions of explainability. With the current discussion, we aim to move from *multidisciplinary* (Choi and Pak 2006) debates about technologically-assisted selection mechanisms towards *inter*- and potentially *transdisciplinary* solutions, that can be implemented in responsible ways.

With regard to job candidate screening in personnel selection, we will focus primarily on the *early selection stage* of the process, in which we assume that there are suitable candidates in a large applicant pool, but no selection decisions have yet been made. As a consequence, all candidates should be evaluated, and based on the evaluation outcomes a subset of them should be selected for the next selection stage, which may e.g. be an in-person interview.

The remainder of the chapter is outlined as follows:

- In Sect. 2, we will explain major methodological interests in psychology and computer science (considering machine learning in particular) in a way that should be accessible to practitioners in either discipline. We will also discuss their major similarities and differences.
- Subsequently, in Sect. 3, we move towards the domain of personnel selection, introducing the domain, its major research questions and challenges, and several important focus areas with key references.
- As a use case, Sect. 4 discusses a data-driven explainable solution that was developed in the context of the ChaLearn Job Candidate Screening Coopetition, with explicit consideration of potential connection points for psychologists and practitioners.

- Then, Sect. 5 focuses on research on acceptability of technology-supported personnel selection solutions, as perceived by two categories of user stakeholders in the personnel selection problem: job applicants and hiring managers.
- Finally, in Sect. 6, considering the various viewpoints provided in this chapter, we will give several recommendations towards interdisciplinary personnel selection solutions.

2 Common Methodological Focus Areas

In this section, we will give broad and brief descriptions about how psychological and computer science are conducted. These descriptions are intended to neither be exhaustive nor highly detailed. Rather, they are meant as an introduction to the uninitiated in each field, in vocabulary that should be understandable to all. Our aim is to inspire discussion on the intersections where the two may meet, and the separate paths where they do not. As such, many of the points are presented with sparse references only where necessary; for readers seeking more thorough explanations and more domain-technical definitions, we will include references to several classical textbooks.

2.1 Psychology

2.1.1 Psychometrics

Psychology uses procedures, like questionnaires, interview protocols, and role-play exercises as tools to assess and quantify differences between individuals. Unlike direct forms of measurement such as height or weight, psychology investigates *constructs*, which are unseen aspects of individuals such as intelligence and personality. The assumption is that these constructs exist unseen in some quantity, and that individual differences in relation to these constructs are observable using reliable and valid procedures. By examining the relationship between measured constructs and observable behaviors, psychology seeks to increase our understanding of people.

While questionnaires are commonly used, any systematic procedure used to gather and quantify psychological phenomena can be considered as a psychological *instrument*. Investigating how well a psychological instrument is measuring what it is supposed to measure is called *psychometrics*. Given that psychological phenomena are both complex and challenging to observe, and that the data collected must be interpreted, a study of the instruments themselves is crucial. Psychometrics can be thought of as the analytical procedures that examine the type of data collected, and estimate how well the variables collected using psychological instruments are reliable and valid. A useful textbook on the subject matter is the book by Furr and Bacharach (2014).

2.1.2 Reliability

Reliability refers to the degree to which the variables produced by a procedure can be shown to be consistent, replicable, and free from measurement error. Similar to instruments in other fields, psychological questionnaires produce measurements that contain random 'noise'. Psychometric methods that assess reliability attempt to quantify the amount of 'signal' to 'noise', and how researchers might increase the amount of signal relative to the noise. By extension, reliability is a matter of degree; although two separate instruments may attempt to measure the same construct, one may have less measurement error than the other.

With regards to questionnaires, reliability is often concerned with *internal consistency*; specifically, how well the individual questions on the survey relate to each other, and to the overall survey scores. As we would expect multiple items on an instrument to measure the same construct, and as we would expect that construct to exist in individuals with some quantity, we would then expect responses to be consistent with each other. Measures of internal consistency, such as the alpha coefficient (Cronbach 1951), examine the degree to which responses to the items on the test correlate with each other, and with the overall test score. Over the course of the development of an instrument, items that do not correlate well with the rest of the questions may be reworded, removed, or replaced with questions that produce more consistent responses. Thus, an instrument is developed and made sufficiently reliable for use.

Another common form of reliability regards test scores over time; *test-retest reliability* is the degree to which scores administered by one test will correlate with scores from the same test at a different time. Whether test-retest reliability is relevant is related to the construct being examined. Because we would not expect mood to be perfectly stable—mood is regarded as a 'state' and not a 'trait'—expecting consistent responses on a questionnaire designed to assess mood over time is not sensible. However, because we expect personality to be stable, we would expect a participant's responses on one testing occasion to correlate with their responses on a second testing occasion, and therefore being replicable across occasions.

In situations where individuals are asked to give subjective ratings, two forms of reliability are relevant: how reliable the ratings are among a group of raters (*inter-rater* reliability), and how reliable the multiple ratings are from the same rater (*intra-rater* reliability). With regards to judgments of relevant constructs, such as personality, inter-rater reliability refers to the replicability of ratings across multiple raters who judge a target person. In other words, to what degree do the ratings gathered from multiple people correlate? Conversely, intra-rater reliability refers to the degree to which a single person's ratings are consistent. With regards to personality, for example, will the rater judge the same person consistently over time?

The more reliable the instrument, the less random uncorrelated 'noise' compared to an interpretable 'signal' is present. Further, the more reliable the instrument, the more the observed magnitude of construct will approach the true magnitude of the construct. As such, the reliability of instruments is paramount.

2.1.3 Validity

However, whether or not a procedure is measuring the underlying construct it is attempting to measure goes beyond whether or not it is consistent. Reliability concerns the more mechanical elements of the instrument, namely the degree to which there is consistency vs. error in the measurements. However, determining how to interpret the measurements gathered by psychological instruments is a matter of *validity*. More specifically, validity refers to the degree to which interpretations of the variables are supported by prior research and theory. In this sense, the measurements produced by a procedure are neither valid nor invalid. Rather, validity is determined by the degree to which the variables produced by the instrument are interpretable as reflecting some psychological phenomenon. Discourse on how best to demonstrate validity has produced a number of validity 'types'. While a complete discussion on validity is beyond the scope of this chapter, a brief summary follows.

Construct validity refers to demonstrating and explaining the existence of unseen constructs, also known as '*signs*', beyond their reliable measurement. For example, personality questionnaires are common instruments for collecting quantifiable observable behavior about a person. The Big Five (McCrae and Costa 1999) personality questionnaire is designed to allow researchers to assess personality along 5 dimensions. Specifically, it asks individuals to indicate how strongly they agree with a set of statements from 1 (strongly disagree) to 7 (strongly agree), thus producing a score for each *item*. If scores for the items vary between people, the variance can be quantified and examined, and underlying dimensions can be identified. By demonstrating the emergence of similar numbers of factors in procedures like the Big Five (or other personality questionnaires, such as the NEO-PIR, FFM, or HEXACO) in samples across cultures, and by demonstrating correlations to other meaningful variables, researchers have demonstrated construct validity for personality.

Criterion validity refers to the degree to which test scores (the predictor) correlate with specific criterion variables, such as job performance measures. It is often discussed in terms of two types: *concurrent validity*, which refers to the correlation of the predictor and criterion data that are collected at the same time, and *predictive validity*, which refers to the correlation of predictor data collected during a selection procedure and criterion data collected at a later time.

Predictive validity is often considered the most important form of validity when during a selection procedure, rather than testing for specific and explicit signs that are considered relevant to future job performance measures, the test would rather consist of taking holistic *samples* of intended future behavior. This means of assessment is based on the theory of behavioral consistency, stating that past behavior is the best predictor of future behavior. In this sense, the predictor data may be collected during the selection process, and later correlated with data collected when selected applicants have become employees. For example, a prospective aircraft pilot may be asked to perform an assessment using a flight simulator. If the variables extracted through the flight simulator correlate with later assessments of performance when the candidate has become an employee, the test allows for predictions of future performance. Therefore, we might conclude that the simulator test has demonstrated predictive validity.

In sample-based approaches, decomposition of the observed behavior into constructs is not sought, and as such, construct validity is less relevant. On the other hand, it is relevant whether or not the test produces scores that correlate to certain key criteria, like future ratings of job performance for example.

Content validity refers to the degree to which each item, question, or task in a procedure is relevant to what should be tested, and the degree to which all aspects of what should be tested are included. For example, personality research has shown evidence for multiple psychological dimensions, sometimes called personality *facets*. In other words, when we refer to the various aspects of one's personality, such as whether they are extraverted, agreeable, conscientious etc., these are various psychological dimensions that collectively comprise the construct we call 'personality'. Individual psychological dimensions may or may not be shown to correlate with each other, but are shown to be distinct e.g. via the results of an Exploratory Factor Analysis. If we were to develop a new method for assessing personality, the full spectrum of the various personality dimensions must be included in the assessment in order for us to demonstrate content validity. In addition, each element of the procedure must be shown to measure what it is designed to measure. In the case of questionnaires, the actual words in the questions should reflect what it is that they are designed to assess.

Face validity is the degree to which the items or tasks look plausible to, and can be understood by participants, and not just to experts. For example, when the test items concern questions on submissive behavior and the test is called the Submissive Behavior Test, participants may be persuaded that it is measuring submissiveness. Another example regards whether participants understand specifically what the questions are asking. If the questions are poorly translated or contain words that are ambiguous or unknown to participants, such as technical jargon or terms that have very specific meanings in one domain but various meanings in other domains, this may affect participant responses. Should the instructions or wording of a questionnaire be confusing to the participants taking it we might also say it lacks face validity.

Convergent validity refers to the degree to which two different instruments, which aim to measure the same construct, produce measures that correlate. For example, we would expect scores from multiple questionnaires that measure Extraversion, a dimension of personality, to correlate. We would further expect that a person's loved ones would rate their degree of Extraversion similarly, and that these ratings would correlate with each other and the individual's test scores. Furthermore, we would expect that measures of Extraversion would correlate with related constructs and observable behaviors. On the other hand, *divergent validity* refers to the expectation that the construct an instrument is measuring will not correlate with unrelated constructs. If a measure of Extraversion consistently correlates highly with another personality dimension, such as Conscientiousness, the measures may not be clearly distinct. In other words, both forms of validity are concerned with the degree to which test scores exhibit relationships to other variables, as would be expected by existing theory.

2.1.4 Experimentation and the Nomological Network

Psychology aims to explain constructs that are not directly observable, by examining the relationships between them, along with their relationships to observable behaviors. This involves demonstrating whether a construct exists in the first place, whether and how it can be reliably measured, and whether and how it relates to other constructs. The complete collection of evidenced and theoretical relationships (or lack thereof) between constructs, along with the magnitudes of their relationships, is called the *nomological network*. The nomological network surrounding a specific construct encapsulates all its relationships to other constructs, some of which will be strong and others of which will be weak.

Psychology develops knowledge by testing hypotheses that expand this network, testing competing theories in the network, or clarifying the magnitudes of the relationships in this network. The researcher derives hypotheses from what one might expect the relationships between variables to be, based on existing research and theory. Procedures are designed to collect data with as little 'noise' as possible, by creating controlled and repeatable conditions, and using reliable and valid instruments. The relationships between the measures from the various instruments are then subjected to statistical tests, usually in the family of general linear modeling (i.e. regression, F-tests, t-tests, correlations etc.), although Bayesian and algorithmic techniques have recently started to appear. In this way, psychology seeks to develop our understanding of the relationship between independent and dependent variables, and by extension, the nomological network surrounding a specific topic.

Although the variables are often described as independent/predictor variables or dependent/outcome/criterion variables, tests are often conducted on concurrent data, where all data points are collected at approximately the same time. As such, the placement of a variable as the independent or dependent may be a matter of statistical modeling, and not whether it is actually making a prediction.

Reliability and validity play an important role in this process. Reliability concerns itself with random error in measurements, which are expected to be uncorrelated with any of the variables being measured. As such, the lower the reliability, the more error in the data, the more attenuated the relationship between constructs will appear to be. The magnitude of the observed effect, in turn, affects the results of statistical significance tests which are often used to determine whether results are interpretable. On the other hand, part of the validation process is demonstrating the effect size of relationships. Specifically, it is necessary to determine how strong relationships between variables are, beyond whether their relationship is statistically significant. Based on prior theory, we often can estimate at least whether a relationship between two constructs ought to be statistically significant, and whether it ought to be strong or weak. When data show the predicted pattern of correlations between constructs, instruments demonstrate validity.

In areas of the nomological network where relationships have yet to be studied, exploratory studies may first be conducted to set the foundation for developing theory. Such studies may include qualitative techniques such as interviews, or questionnaires that allow participants to type their responses freely. Exploratory studies may also include quantitative techniques, such as Exploratory Factor Analysis (EFA). EFA is often used in the development of questionnaires with Likert-scale items, as it allows the researcher to examine whether or not multiple dimensions are present in the questionnaire, and by extension, the dimensionality of the construct it seeks to measure. By showing how individual items on a questionnaire correlate to one or more latent variables, the researcher can develop the theoretical structure of a construct. For example, personality researchers used such methods to develop theory on the various personality facets. Procedures like EFA may show that certain items on an instrument correlate with a hypothetical axis, much more so than with other hypothetical axes. Based on the wording and content of the questions that cluster together, these hypothetical constructs can be named (e.g., Extraversion vs. Conscientiousness). With an initial estimate of the structure of a construct, researchers can then use a more restricted analytical technique, such as Confirmatory Factor Analysis, to examine whether and how well the exploratory model fits newly collected data.

Psychology researchers are faced with certain limitations, however. The data collection process is often labor-intensive, time is necessary to stay current on theory and research in order to develop hypotheses, and samples are often drawn by convenience leading to a preponderance of student WEIRD samples (Western, Educated, Industrialized, Rich, Democratic) (Henrich et al. 2010). Nevertheless, by conducting exploratory and confirmatory studies, psychology researchers contribute knowledge about how individual constructs relate to each other and observable behaviors.

2.2 Computer Science and Machine Learning

The domain of computer science studies the design and construction of both computers, as well as the automated processes that should be conducted by them. *Generalization* and *abstraction* are important values of the domain. As for generalization, a solution to a problem should not only work in a specific case, but for a broader spectrum of cases—ideally, in any possible case that can be thought of for the given problem. For this reason, it may be needed to not always describe and treat the problem in full contextual detail, but rather in a more abstracted form, that can be used for multiple variants of the problem at once. Here, mathematics and logic contribute the language and governing principles necessary to express and treat generalization and abstraction in formalized, principled ways. Furthermore, *efficiency* and *scalability* are of importance too: through the use of computers, processes should be conducted faster and at larger scale than if their equivalent would be conducted in the physical world only.

Computer processes are defined in the form of *algorithms*, which are sets of explicit instructions to be conducted. Algorithms can be formally and theoretically studied as a scientific domain in itself: in that case, the focus is on formally quantifying and proving their properties, such as lower and upper bounds to the time and memory space they will require to solve a given problem (*computational* *complexity*). In many other cases, algorithms will rather be used as a tool within a broader computational context.

Within computer science, a domain receiving increasing attention is that of *artificial intelligence* (AI). In popular present-day discourse, 'AI' is often used to indicate specific types of *machine learning*. However, artificial intelligence is actually a much broader domain. While no single domain definition exists, it can be roughly characterized as the field focusing on studying and building intelligent entities. The classical AI textbook by Russell and Norvig (2010) sketches four common understandings of AI, including 'thinking humanly', 'thinking rationally', 'acting humanly', and 'acting rationally'. Furthermore, a philosophical distinction can be made between 'weak AI' and 'strong AI': in the case of weak AI, machines act as if they are intelligent, and only simulate thinking; in the case of strong AI, machines would be considered to actually think themselves. While popular discourse tends to focus on strong AI, in practice, many present-day AI advances focus on weak AI in limited, well-scoped domains. Within AI, many subdomains and focus areas exist, including studies of knowledge representation, reasoning and planning, dealing with uncertainty, learning processes, and applying AI in scenarios that require communication, perception, or action.

Machine learning can be considered as the AI subdomain that deals with *automatically detecting patterns from data*. The 'learning' in 'machine learning' denotes the capacity to automatically perform such pattern detections. In the context of the job candidate screening problem, machine learning is the type of AI that most commonly is applied, and therefore, the most relevant subdomain to further introduce in this section. First, we will focus on discussing the main focus points in fundamental machine learning, in particular, *supervised machine learning*. Then, we will focus on discussing how machine learning is typically used in applied domain settings. Following this, the next section will discuss how common methodological focus areas in psychology and machine learning are overlapping, contrasting, and complementing one another.

2.2.1 The Abstract Machine Learning Perspective

In machine learning, algorithms are employed to learn relevant patterns from data. Different categories of machine learning exist, most notably:

- *Unsupervised machine learning*, in which a dataset is available, but relevant patterns or groupings in the data are initially unknown. Statistical data analysis should be employed to reveal these.
- *Supervised machine learning*, in which in connection to data, known *targets* or labels are provided. The goal will then be to relate the data to these targets as accurately as possible.
- *Reinforcement learning* (Sutton and Barto 1998), in which the focus is on learning to act towards a desired outcome: an agent should learn those actions in an environment (e.g., game playing actions), that will lead to an optimal reward (e.g., a high score).

In this chapter, we focus on supervised machine learning. With a focus on generalization and optimal exploitation of statistical patterns encountered in data, supervised machine learning algorithms are not pre-configured to specialize in any particular application domain. Therefore, more formally and more abstractly, it can be stated that the goal of a supervised machine learning algorithm is to learn some function $f(\mathbf{x})$ that relates certain input observations $\mathbf{x}$ to certain output targets $\mathbf{y}$, in a way that is maximally generalizable and effective. If $\mathbf{y}$ expresses categorical class memberships, a *classification* problem is considered. If $\mathbf{y}$ rather expresses one or more continuous dependent variables, a *regression* problem is considered.

For simplicity, the remainder of this discussion focuses on cases in which $f(\mathbf{x})$ has the form $f : \mathbb{R}^d \rightarrow \mathbb{R}^1$. In other words, input observations are represented by $\mathbf{x}$, a d-dimensional vector, of which the values are in the set of all real numbers $\mathbb{R}$—in other words, $\mathbf{x}$ contains d real numbers. $\mathbf{x}$ should be mapped to a single real number value y, expressing the target output.

To learn the appropriate mapping, a *training* stage takes place first, based on a large corpus with various examples of potential inputs $\mathbf{x}_{train}$, together with their corresponding target outputs y_{train}. For this data, the human machine learning practitioner specifies the model that should be used for $f(\mathbf{x})$. Examples of models can e.g. be a linear model, a decision tree, a support vector machine, a neural network, or a deep neural network (Bishop 2006; Goodfellow et al. 2016). Initially, the parameters that the chosen model should have to optimally fit the data are unknown. For example, for a linear model, these would be the slope and intercept. During the training phase, considering statistical properties of $\mathbf{x}_{train}$ and y_{train}, a model-specific machine learning algorithm will therefore iteratively optimize the necessary model parameters, by minimizing an expert-defined error measure between estimated outputs $\hat{y}$ and true outputs y. For example, for a linear model, this may be the sum of squared errors between each $\hat{y}$ and y in the training set.

To assess whether the learning procedure has been successful in a generalizable way, the final reported performance of the learned $f(\mathbf{x})$ will be computed by running $f(\mathbf{x})$ on a *test* set, which contains input data that was not used during the training phase. As the final learned $f(\mathbf{x})$ specifies the necessary mathematical transformation steps that should be performed on $\mathbf{x}$ in order to predict y, it can be used as an optimized *algorithm* for predicting y from $\mathbf{x}$.

It should be re-emphasized that from a pure machine learning perspective, the only requirement on the nature of $\mathbf{x}$ and y is that they can be specified in numerical form. The only 'meaning' that $\mathbf{x}$ and y will have to the model learning procedure, is that they contain certain numeric values, which reflect certain statistical properties. With the focus on finding an optimal prediction function $f(\mathbf{x})$, the tacit assumption is that finding a mapping between $\mathbf{x}$ and y makes sense. However, the procedure for learning an optimal $f(\mathbf{x})$ only employs statistical analysis, and no human-like sense-making. It will not 'know', nor 'care', whether $\mathbf{x}$ and/or y consider synthetically generated data or real-world data, nor make any distinction between flower petal lengths, census data, survey responses, credit scores, or pathology predictions, beyond their values, dimensionality, and statistical properties. When considering real-world data, it thus is up to the human practitioner to propose correct and reasonable data for $\mathbf{x}$ and y.

While various machine learning models have various model-specific ways to deal with noise and variance, further tacit assumptions are that $\mathbf{x}$ realistically follows the distribution of future data that should be predicted for, and that y is 'objectively correct', even if it may contain some natural noise. In applied settings, in case the target outputs y consider labels that are obtained through an acquisition procedure (through empirical measurement, or by soliciting human annotations), y also is frequently referred to as 'ground truth', which again implies that y is truthful and trustable.

Being oblivious to human data interpretation, machine learning algorithms will not 'understand' any potential 'consequences' of correct or incorrect predictions by themselves. If such considerations should be taken into account, it is up to the human expert to encode them properly in the defined error measure. For example, in case of binary classification, in which y can only have the values 'true' or 'false', *false negative* classification errors (making a 'false' assessment where a 'true' assessment was correct) and *false positive* classification errors (making a 'true' assessment where a 'false' assessment was correct) may need to be weighted differently. For example, if a binary classification procedure would consider assessing the occurrence of a certain disease in a patient, false negatives (i.e., incorrectly labeling a diseased patient as healthy) may be deemed much graver mistakes than false positives (i.e., incorrectly labeling a healthy patient as diseased), as false negative assessments will cause diseased patients to not be treated. If so, for the error measure employed during learning, the penalty on making a false negative classification should be defined to be much larger than the penalty on making a false positive classification.

2.2.2 Machine Learning in Applied Domains

As discussed in the previous section, the focus in fundamental machine learning is on learning $f(\mathbf{x})$ in an optimal and mathematically well-founded way, considering the given statistical properties of $\mathbf{x}$ and y, as well as the specified error measure. While from a fundamental perspective, it does not matter whether $\mathbf{x}$ and y are synthetically generated or real-life data, interpretation of $\mathbf{x}$ and y does matter when machine learning techniques are considered in applied domains, such as computer vision and bioinformatics.

In such applied cases, typically, y represents a dependent variable considering a natural sciences observation, that can objectively be verified in the physical world. For example, it may denote the value depicted by a hand-written number, the occurrence of a disease, the boundaries of a physical object, or the identity of a person. The input data $\mathbf{x}$ often is the 'raw', high-dimensional result of a noisy sensory measurement procedure: for example, it may express color intensity values of different pixels in an image, an audio waveform, or microarray gene expression data. A human being will not be capable of relating such noisy measurements to their target outputs reliably; in contrast, a machine learning procedure has the power to systematically find relevant properties, rules and correlations between **x** and y.

Historically, before initiating the learning procedure, a pre-processing step would be performed on **x**. In such a step, raw data measurements would first be turned into semantically higher-level, humanly hand-crafted *features*. For example, the color intensity values of individual pixels in a full image may first be summarized in the form of a histogram; an audio waveform may first be summarized in the form of dominant frequencies over short-time analysis frames. This type of modeling is meant to narrow the *semantic gap* (Smeulders et al. 2000) between observations that are very obvious to humans, and the noisy low-level measurements from which this observation may be inferable. For example, when provided with pictures of cats and cartoon characters, a human will very easily be able to tell the two apart. However, it is hard to define what color a certain pixel at a certain location should have, in order to belong to a cat or a cartoon character. Generally, objects of focus may also be located at different parts in the image, implying that the exact pixel location may not even be relevant information. When choosing to use a histogram as feature, the picture color values are summarized. The pixel location information is then lost, but we obtain a color and color intensity distribution over the whole image instead. This is therefore a representation of lower dimensionality than when all pixels of the input image are considered in their raw form, but it may give more interpretable information for the statistical model to tell cats apart from cartoon characters.

In recent years, it has increasingly been debated whether going through a feature extraction step is necessary. As an alternative, provided that sufficient training data and powerful deep learning architectures are available, machine learning procedures can be employed for *representation learning* (Bengio et al. 2013), directly learning relevant feature representations from **x**, without a human expert indicating what information in **x** should be filtered or focused on. Going even further, *end-to-end learning* has also been proposed, in which case the relation between **x** and y is directly learned without the need for an intermediate representation. In many cases, this yields better performance than strategies including intermediate and human-crafted representations (e.g. Graves and Jaitly 2014; Long et al. 2015). At the same time, the ultimately learned function from **x** to y becomes harder to interpret for human beings this way.

Since the advent of machine learning, it has been applied to domains which consider phenomena that have natural, physical and objective evidence in the world, although this evidence may not encompass the full breadth of the phenomenon under study. Examples of such domains include speech and natural language (commonly manifesting as spoken audio and text) and music (commonly manifesting as audio). Beyond the physical representation and description of these phenomena, contextual layers of associated modalities, as well as social, human and subjective interpretation, play an important role in the way they are perceived and understood by humans (Liem et al. 2011; Davis and Scharenborg 2017).

While machine learning algorithms has proven effective in learning patterns regarding the more descriptive aspects of such phenomena (e.g. Collobert and

Weston 2008; Hamel and Eck 2010), it is still problematic for them to capture notions of true human-like 'understanding' (Hofstadter 2018; Sturm 2014). This does not only occur in domains in which 'meaning' may be a shared natural and social phenomenon, with observable and unobservable aspects. Even when the domain considers a pure natural sciences problem with fully objective ground truth, it is not guaranteed that an optimized machine learning procedure mimics human understanding of the problem. This especially can be seen when studying errors made by a seemingly optimized system. In the context of deep neural networks, the notion of *adversarial examples* has emerged: small, humanly unnoticeable perturbations of data on which correct model predictions were originally made, may provoke incorrect model answers with high model confidence (Goodfellow et al. 2015).

2.3 Contrasting Focus Areas in Psychology and Machine Learning

Considering the focus areas discussed above, several commonalities and contrasts can be found between interests in psychology and machine learning. Table 1 summarizes several conceptual approximate analogies, as well as their main differences.

In both domains, a prediction task may be studied, involving an $\mathbf{x}$, $f(\mathbf{x})$ and y. However, the parts of the prediction procedure considered to be of main interest, and the typical types of conclusions being drawn, differ, as also illustrated in Fig. 1.

The machine learning concept of training vs. testing has analogues to the difference between exploratory vs. confirmatory factor analysis in psychology. However, in psychology, the focus would be on understanding data, while in machine learning, it is used to verify that a robust model has been trained.

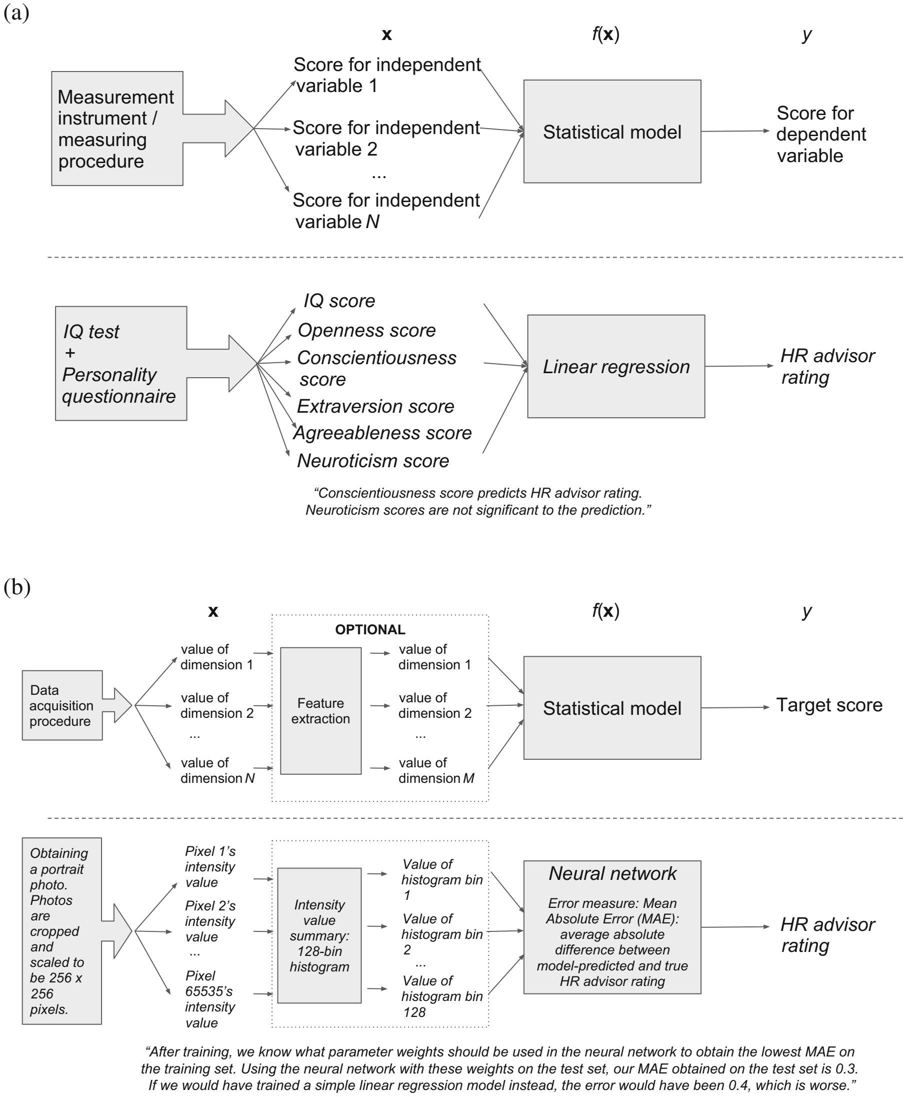
In psychology, human-interpretable meaning of $\mathbf{x}$ and y is essential: ensuring that $\mathbf{x}$ will only contain psychometrically validated measurable components that are understandable to a human being, selecting a set of such reasonable components to go into $\mathbf{x}$, understanding which aspects of $\mathbf{x}$ then turn out important regarding y, and understanding how y human end-users perceive and accept y and $f(\mathbf{x})$. It is critical that choices of $\mathbf{x}$ are driven by theory, and corresponding explicit hypotheses about significant relations between the components within $\mathbf{x}$ and y.

The above focus points are out of scope in machine learning. A machine learning expert typically is interested in understanding and improving the *learning procedure*: understanding why $f(\mathbf{x})$ gets learned in the way it is, where sensitivities lie in the transformation from $\mathbf{x}$ to y, and how prediction errors made by $f(\mathbf{x})$ can be avoided.

In fundamental machine learning, the focus will exclusively be on this $f(\mathbf{x})$, and the origins of $\mathbf{x}$ and y (as well as the reasonableness of any human-interpretable relationship existing between them) will be irrelevant, as long as their statistical properties are well-defined. In applied settings, $\mathbf{x}$ and y will have further meaning to

Table 1 Psychology vs. machine learning: conceptual approximate analogies

Psychology	Machine learning	Major conceptual differences
Exploratory factor analysis	Unsupervised learning	In both domains, if data is available but relationships within the data are unknown, these relationships can be revealed through data analysis. Exploratory factor analysis can be considered as one out of many unsupervised learning techniques, with special focus on explainability of relations in terms of the original input dimensions.
Independent/predictor variables	Input data	Each psychological independent variable, as well as its individual dimensions, is human-selected and human-interpretable. In a machine learning setup, input data is usually not manually picked at the individual dimension level. The semantic interpretation of individual dimensions in the data usually also is at a much lower level than that of independent variables in psychology.
Variable dimension	Feature	Features express interpretable subinformation in data, where psychological variable dimensions describe interpretable subinformation of an overall variable. Where psychological variable dimensions are explicitly human-selected and human-interpretable, features may be extracted by hand or through an automated procedure. They are still at a semantically lower level than psychological variables dimensions, and not restricted to be psychologically meaningful.
Dependent/outcome/criterion variables	Output/targets/labels/ground truth (if obtained through acquisition)	These concepts can be considered as equivalents.
Statistical model	Statistical model	In psychology, a linear regression model is commonly assumed, and considering other models is typically not the focus. In machine learning, identifying the model that obtains the most accurate predictions (which usually is not a linear regression model) would be the main focus.
Model fitting	Training	In psychology, the squared error between predicted and true values will commonly form the error measure to be minimized. In machine learning, more flexible error or cost functions may be used.


(a)
x
f(x)
y
Measurement instrument / measuring procedure
Score for independent variable 1
Score for independent variable 2
...
Score for independent variable N
Statistical model
Score for dependent variable
IQ test + Personality questionnaire
IQ score
Openness score
Conscientiousness score
Extraversion score
Agreeableness score
Neuroticism score
Linear regression
HR advisor rating
(b)
x
OPTIONAL
f(x)
y
Data acquisition procedure
value of dimension 1
value of dimension 2
...
value of dimension N
Feature extraction
value of dimension 1
value of dimension 2
...
value of dimension M
Statistical model
Target score
Obtaining a portrait photo. Photos are cropped and scaled to be 256 x 256 pixels.
Pixel 1's intensity value
Pixel 2's intensity value
...
Pixel 65535's intensity value
Intensity value summary: 128-bin histogram
Value of histogram bin 1
Value of histogram bin 2
...
Value of histogram bin 128
Neural network
Error measure: Mean Absolute Error (MAE): average absolute difference between model-predicted and true HR advisor rating
HR advisor rating

"Conscientiousness score predicts HR advisor rating. Neuroticism scores are not significant to the prediction."

"After training, we know what parameter weights should be used in the neural network to obtain the lowest MAE on the training set. Using the neural network with these weights on the test set, our MAE obtained on the test set is 0.3. If we would have trained a simple linear regression model instead, the error would have been 0.4, which is worse."

Fig. 1 Prediction pipelines in psychology and machine learning. Abstracted pipelines are given on top, simplified examples of how they may be implemented at the bottom, together with a typical conclusion as would be drawn in the domain. (**a**) Psychology (in an organizational psychology application). (**b**) Machine learning (in a computer vision application)

a human, although in many cases, they consider objectively measurable observations in the physical world, with **x** containing raw data with low-level noisy sensory information.

The flexibility in choosing $f(\mathbf{x})$ in machine learning is unusual in psychology, where linear regression models are commonly chosen for $f(\mathbf{x})$, and not typically contrasted with alternative models. The other way around, criterion validity, considering the alignment of y with that what is supposed to be measured, is hardly ever questioned in machine learning settings. In psychology, even though certain types of measures (e.g. supervisor rating as indicator of job performance in the personnel selection problem) tend to dominate, criterion validity is an explicitly acknowledged topic.

When machine learning is to be applied to psychological use cases, y will consider human-related latent concepts, for which no direct and objective measuring mechanisms exist yet in the physical world. When seeking to predict these concepts, it can be debated whether **x** should be expressed at the latent human concept level (constructs/meaningful independent variables) as well. This would be natural for a psychologist, but controversial for a machine learning expert.

Alternatively, an empiricist approach can be taken, purely considering sensory observations, and trying to relate these directly to y. This would be natural for a machine learning expert, but controversial for a psychologist. As a possible compromise, if **x** consists of raw data observations, the use of hand-crafted features forms a data-driven analogue to the use of variable dimensions relating to constructs in psychology, even though extracted features will be at a semantically much lower level.

Following these considerations, when applied machine learning methodology is to be integrated in a psychological predictive pipeline, various ways of integration can be imagined:

1. Keep a traditional psychological pipeline, with traditional input and output data, but consider alternative statistical models to the commonly used linear regression model. This would boil down to varying the choice of statistical model in a traditional psychological pipeline as shown in Fig. 1a, top.
2. Keep a traditional machine learning pipeline (as shown in Fig. 1b, top), but ensure that features extracted from raw signals are psychologically informed.
3. Explicitly replace a traditional measurement instrument by a data-driven equivalent. In that case, **x** consists of high-dimensional raw data (e.g., video data), but we wish to turn it into associated traditional instrument scores (e.g., personality trait assessments), so our y can be seen as a transformed version of **x**—say, $\mathbf{x}'$—, at a commonly understood semantic level in psychology, which then can be (re)used in more comprehensive pipelines.

   For going from **x** to $\mathbf{x}'$, hand-crafted features can also be extracted. Subsequently, a statistical machine learning model can be employed to learn correspondences between these feature values and the traditional instrument scores (Fig. 2a).

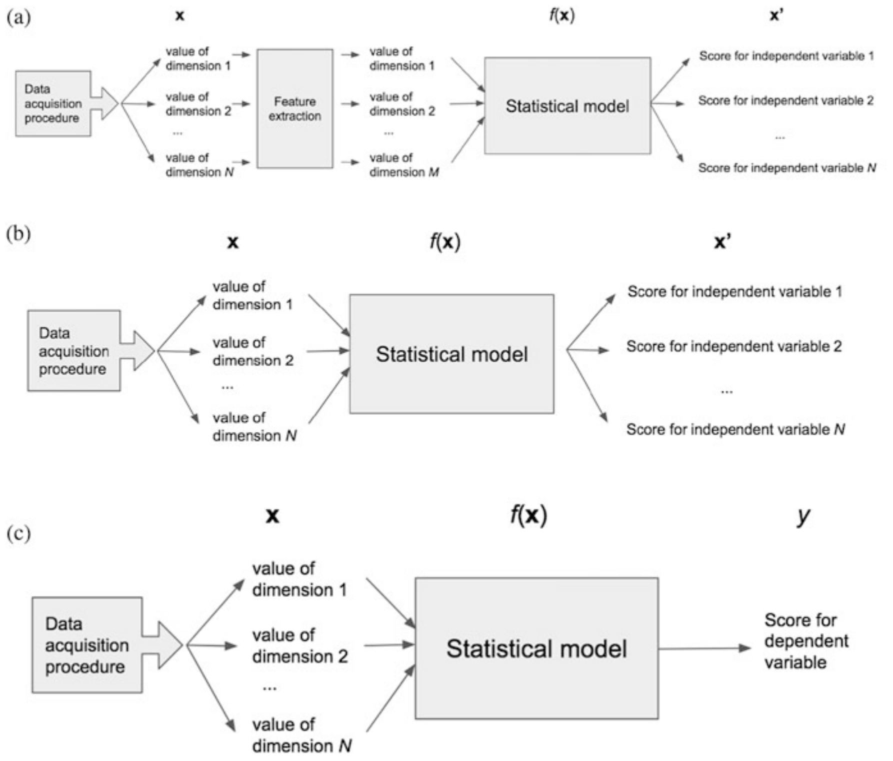


Fig. 2 Various ways in which psychological and machine learning prediction pipelines can be combined. (**a**) A machine learning approach replaces a traditional measurement instrument. Hand-crafted features extract information from raw data. These are subsequently used in a prediction pipeline, in which correspondences are learned between obtained feature scores, and psychologically meaningful variable scores that were obtained in correspondence with the raw input data. (**b**) A machine learning approach replaces a traditional measurement instrument. Representation learning is applied: a sophisticated statistical model should directly learn the correspondences between raw data input, and corresponding psychologically meaningful variable scores. (**c**) A machine learning approach replaces the full psychological pipeline. End-to-end learning is applied: a sophisticated statistical model should directly learn the correspondences between raw data input, and corresponding psychologically meaningful constructs

Alternatively, instead of performing a hand-crafted feature extraction step, a sophisticated machine learning model can be employed to directly learn a mapping from raw data observations to $\mathbf{x}'$ (Fig. 2b). This would be a way to apply automatic *representation learning* in psychological use cases.

In feature engineering, a human should explicitly define how an input signal should be transformed, while in representation learning, this would be the task of the chosen statistical model. Especially if it is not very clear how a target instrument score may concretely relate to information in sensory input data, automated representation learning may therefore yield more optimized mappings than a human can indicate.

In other words, if the predicted target labels are scores of traditional instruments, and the practitioner is sure that criterion and content validity are indeed maintained in the automated learning procedure, representation learning may be an interesting data-driven way to make use of known psychological vocabularies, while bypassing explicit treatment of the semantic gap. However, at the same time, the explicit treatment of the semantic gap through feature engineering can be likened to theory-forming, while in representation learning, a human will have much less control of what the learning algorithm will focus on.

4. Directly seek to learn a meaningful mapping from raw sensory data in $\mathbf{x}$ to a dependent variable y, omitting any intermediate feature or representation extraction steps. This would be an *end-to-end learning* scenario. Conceptually, this approach is close to the representation learning approach mentioned in the previous item. As major difference, in representation learning, the predicted variables are intended to become an alternative to outcomes of a traditional measurement instrument. Therefore, they usually form an intermediate step in a prediction pipeline, replacing the feature extraction block. In case of end-to-end learning, y is the direct output to predict, without including any intermediate explicit representation steps (Fig. 2c).

2.4 Conclusion

With the main methodological interests of psychology and machine learning being mapped, we identified relevant contrasts and correspondences between these interests. With this in mind, in the next section, we will proceed by giving an introduction to common personnel selection criteria. Then, Sect. 4 will illustrate how varying methodological insights into the personnel selection problem can come together in a data-driven solution.

3 The Personnel Selection Problem

Historically, personnel selection has been approached as a problem in which *future job performance* should be predicted from job candidate evidence, as provided during the personnel selection stages.

First of all, it is necessary to assume that suitable job candidates exist and that they are willing to apply for the job. Finding these suitable candidates is the focus of recruitment processes. Because it is necessary to have suitable candidates within the applicant pool to be able to select effectively, recruitment and selection are closely intertwined and decisions about selection procedures can influence both processes (Ployhart et al. 2017).

During the early selection stage, the interaction between the applicant and the hiring organization is still low. More precisely, organizations have to rely on limited information (e.g., applicant resumes) in order to decide who to reject and who to keep in the applicant pool. The next stage usually consists of more time-consuming selection procedures, such as face-to-face interviews and/or tests run by assessment centers.

A common hypothesis is that individual characteristics such as Knowledge, Skills, Abilities and Other characteristics (*KSAOs*) are predictive of individual outcomes, such as job performance (Guion 2011). Thus, candidates whose KSAOs fit the job demands are the ones that should be hired. This leads to several central classical questions of interest to personnel selection research, in which technological opportunities increasingly play a role, as discussed below.

3.1 How to Identify Which KSAOs Are Needed?

When an organization needs to select applicants, the first question to be posed is what the organization is looking for. This will be expressed in the form of KSAOs. The logical process to determine KSAOs is to derive these from the job description, and a description of how the job contributes to the organizational goals. For example, if the goal of a hospital is to cure patients, a surgeon in the hospital will be expected to e.g. successfully operate upon patients, correctly analyze the patient's history, coordinate assistants' activities and follow recognized practices during the operation. The needed KSAOs will then, among others, include knowledge and skills regarding techniques for diagnosing and treating injuries and diseases, the ability to tell when something is wrong, and deductive reasoning. Attention to detail, stress tolerance, concern for others and dependability will be further important characteristics.

The KSAOs ideally are derived from a thorough job analysis. A well-known systematic taxonomy of job descriptions, resulting from decades of analyzing jobs, is the occupational net O*NET,[1] which forms the largest digital job taxonomy, containing experience and worker requirements and worker characteristics. In practice, however, job descriptions and person specifications sometimes are drawn up in only a few hours by an organization (Cook 2016).

The characteristics which will be measured during a selection procedure should logically follow from the required KSAOs. In the example of applicants for the occupation of a surgeon, it therefore is important to not only collect information about an applicants' education and experience, but also to measure abilities and traits such as deductive reasoning capacities, attention to detail, concern for others and stress tolerance. A large array of measurement procedures exist to assess applicants' capacities and traits, varying from self-reported personality questionnaires to cognitive tests, work sample tests, structured interviews and role play exercises. As discussed earlier in Sect. 2, the measures that are explicitly intended

[1] https://www.onetonline.org

to assess constructs (traits, abilities) are often labeled ‘signs’, whereas measures which aim to assess a sample of relevant performance or behavior (e.g., simulating an operation on a mock patient) are often labeled ‘samples’. In practice, most often sign-based measures such as interviews are used (because they are efficient and easy to conduct), although samples often show a good predictive validity (Schmidt and Hunter 1998).

Smith (1994) distinguishes between three domains of job characteristics: *universals*, which are characteristics required by all work, *occupationals*, which refer to characteristics required by certain jobs but not others, and *relationals*, referring to characteristics needed to relate to others in specific organizational settings. According to Smith, cognitive ability, vitality, and work importance form the category of universals. The personality factor Conscientiousness (i.e, being organized and structured and able to work on a problem untill the end) may arguably also be seen as a universal. While the aforementioned characteristics have been shown to be relevant for good job performance across most professions, specialized knowledge and certain aspects of personality are examples of occupationals. For a career as a musician, for instance, emotional sensitivity, which is an aspect of emotional intelligence, may be more important than for a job as accountant. Relationals are important to specific settings, and imply a focus on values and norms, and the fit (‘chemistry’) with the people working in those settings such as co-workers, supervisors and management. Relationals mostly are referred to as aspects of *person-organization fit*. More precisely, relationals play an important role when comparing occupations in different organizational settings. For instance, a lawyer in a large commercial bank might require other relationals than a lawyer in a non-profit governmental organization that assists people in poor neighborhoods.

3.2 How to Measure KSAOs?

After defining which KSAOs are needed, it is necessary to develop or decide for the personnel selection procedures in order to find out which applicants fits the job best. Usually, personnel selection is a multi-hurdle approach, meaning that applicants have to pass different stages before they actually receive a job offer. In a first step, applicants might provide a written resume, afterwards they could be asked to answer to a personality and cognitive ability test. Finally, they might be invited to show their abilities within a face-to-face job interview. Desirably, every single step of the selection process should be psychometrically sound and useful to reveal applicants’ KSAOs. As described in Sect. 2, this means that the selection procedures have to prove to be reliable and valid. For instance, if hiring managers develop a job interview to measure applicants’ KSAOs, they have to decide about at least three aspects that may influence psychometric properties of the interview:

- They need to decide for an administration medium. Face-to-face interviews, videoconference interviews and digital interviews all have an impact on applicants' performance ratings (Blacksmith et al. 2016; Langer et al. 2017a) which consequently may affect validity of the interview.

- The degree of standardization of the interview must be decided. This can affect its reliability (Schmidt and Hunter 1998). In the case of an unstructured interview (i.e., interviewers are allowed to engage in unstructured conversation with the applicant and they have no standardized evaluation criteria), reliability of the interview is at risk because interviewer *A* may evaluate an applicant based on different evaluation standards than interviewer *B*. In other words, if these two interviewers interview the same applicant, the interview scores will likely differ, the interviewers will come to different conclusions about hirability of the applicant, and one interviewer might want to hire while the other might want to reject. In contrast, questions and evaluation of answers in a structured interview are highly standardized. This makes interviews and therefore interview scores more comparable, leading to less noise in the data.
- Lastly, hiring managers need to decide about potential interview questions to capture required KSAOs (Pulakos and Schmitt 1995). If a job requires programming skills and the interviewer asks questions about applicants' behavior in conflict situations, the interview will neither appear face valid (i.e., applicants would not understand why this is a job related question), nor content valid (i.e., its content will not reflect programming skills as the construct it aims to measure), nor will it be construct valid (i.e., the score on this question will not correlate with other measures capturing programming skills), nor will it demonstrate concurrent (i.e., if the applicant had good grades in a programming course) or predictive (i.e., predict if the applicant will be a good programmer) validity.

To conclude, assessing a selection procedure's reliability means to assess if applicants hirability ratings will be similar for each time that the applicant undergoes (parts of) the selection procedure. In order to evaluate validity of a selection procedure, it is necessary to estimate if a selection procedure appears job related, if it correlates to related constructs and if it predicts important outcomes.

Spreading the attention to other selection procedures, tests focusing on general mental ability (GMA), such as intelligence tests, were shown to have high validity at low application cost (Schmidt and Hunter 1998; Cook 2016). Considerable attention has also been paid to personality measures (Morgeson et al. 2007). The five factor model of personality (known as the Big Five: Agreeableness, Conscientiousness, Extraversion, Openness to experience, Neuroticism) (McCrae and Costa 1999) is widely accepted and used in and outside the field of psychology. In the case of personnel selection, Conscientiousness has especially shown to be a valid predictor for job performance in various organizational contexts (Barrick and Mount 1991).

However, caution is warranted when assessing personality in the early selection stage, in which resumes are the most frequently used selection instrument. Recruiters may infer impressions from resume data that go beyond the reported factual content. For example, they may attempt to assess an applicant's personality from the resume, which in turn is used to evaluate the applicant's employability. Disconcertingly, there is no research showing that resume-based impressions of applicants' personality are correct. Still these impressions may influence applicants' hirability ratings. In other words, hiring managers have very limited insight into applicants' actual behavior and individual characteristics as they may only have seen applicants' resumes, yet they may still infer much more from the resumes than is appropriate.

This might be a reason why organizations and researchers search for new, efficient sources of information in order to gain additional insights into applicants in early stages of the selection process. However, evidence on the validity of recruiter impressions of the applicants' characteristics based on new, possibly richer sources of applicant information than classical resumes (e.g., from video resumes) is still scarce.

An exception is an experimental study by Waung et al. (2014) on the effect of resume format on candidate evaluation and screening outcomes among a group of MBA students. When mock applicants were evaluated based on their video resumes, they were rated as less open, extraverted, physically attractive, socially skilled, and mentally capable, and more neurotic than when the same applicants were evaluated based on their paper resumes. Those who were rated as more socially skilled and more conscientious had a higher probability of positive ratings. In another study, Apers and Derous (2017) examined the equivalence of video versus paper resumes on applicants' personality and job suitability ratings. They concluded that resume type did not clearly affect applicant ratings. For instance, personality inferences from video resumes appeared as (in)valid as those from paper resumes. Furthermore, Nguyen et al. (2014) developed a computational framework to predict personality based on nonverbal cue extraction. However, with exception to the prediction of Extraversion, results did not support the claim that it is possible to accurately predict various applicant characteristics through automatic extraction of nonverbal cues.

Recent technological developments have opened the door to measuring personality in innovative and possibly more valid ways, such as via Facebook behavior or serious games (Chamorro-Premuzic et al. 2018). These technological developments have sparked interest in both psychologists and computer scientists. For instance, there is evidence that computer-based personality judgments based on digital cues are more accurate than those made by humans (Youyou et al. 2015). The data-driven challenges discussed in this chapter, focusing on predicting personality from online video resumes and YouTube clips (Ponce-López et al. 2016; Escalante et al. 2018) can be considered as further examples of interest in these new algorithm-based methods, even if it has explicitly been presented and disseminated in the technical world.

3.3 Dealing with Judgment

Selection procedures rely severely on assessors who judge applicants' characteristics. Assessors include interviewers but also assessment center observers and managers assessing work-sample performances. As particularly interviews are among the most frequently used selection methods (e.g. Ryan et al. 1999), it is important to focus on judgment accuracy and the characteristics of good judges. Furthermore, a focus on ratings by others seems warranted, as it has been proposed that one of the reasons for the relatively low predictive validity of personality measures is the heavy reliance on self-reports, which may contain several biases such as individual differences in faking (Morgeson et al. 2007).

Oh et al. (2011) indeed provided evidence for this idea by showing that other-ratings of personality improve the predictive validity of personality for job performance. Similarly, among a sample of sales people, Sitser (2014) was able to demonstrate that other-rated personality traits were able to better predict manager-rated job performance than self-rated traits. In particular, the other-rated personality trait Proactivity, proved to be a strong predictor of job performance. Generally, it can be stated that observer ratings contribute to explaining job performance over and above solely self-report ratings of personality, while this is not the case the other way around (i.e., self-report ratings do not add to explaining variance in job performance over and above the variance explained via observer ratings). However, it has to be noted that observer ratings are also not free from problems, as these ratings might also be faked (König et al. 2017).

As can be seen, studies such as the above have mainly focused on the difference between self- and other-ratings in terms of predictive validity. In the domain of person perception research, the focus has been somewhat different, namely focusing on the search for 'the good judge': "*the oldest concern in the history of research on accuracy is the search for the good judge the kind of individual who truly understands his or her fellow humans*" (Funder 1999). In this tradition, Ambady et al. (2000) have demonstrated that merely 'thin slices' of expressive behavior related to Extraversion already result in remarkably accurate judgments of unacquainted judges.

To approach the issue of judgment accuracy, Funder (1999) has developed the well-known Realistic Accuracy Model (RAM). RAM states that the degree to which judgments are accurate is moderated by the following factors: good targets, good traits, good information, and finally, good judges (Funder 2012).

Good targets are very judgeable individuals who may be more transparent than poor targets. Good traits (e.g., extraversion) are more visible than others (e.g., neuroticism) and therefore can be more easily judged. Good information implies good quantity (e.g., a one-hour assessment provides more trait information than a speed-dating exchange) and good quality (e.g., when a person is comfortable and responds to good interview questions, higher-quality information will result). Finally, good judges are better able to detect and use behavior cues to form an accurate personality trait inference.

Yet, HR practices seem to disregard the possibility that individual differences exist in judgment accuracy. A stream of research has focused on potential judge characteristics which may explain individual differences in judgment accuracy. Among these researchers are Christiansen et al. (2005), who used the term *dispositional reasoning* to label individual differences between judges in their complex knowledge of how traits relate to each other and to behaviors, and of situations' potential to elicit traits into manifest behaviors. Christiansen et al. were able to show the importance of dispositional reasoning in predicting judgmental accuracy. Taking this thinking further, De Kock et al. (2015, 2017) provided support for the idea that dispositional reasoning showed incremental validity above general intelligence in predicting judgmental accuracy. In sum, such studies show the importance of asking the question who the external observer is, if we seek better predictive validity of other-ratings.

3.4 What Is Job Performance?

So far, a discussion on selection procedures has been provided; however, how 'job performance', the criterion that should be predicted, is appropriately measured has not been discussed. Usually, job performance is considered at an individual level. Frequently, organizations use *supervisor ratings* of past and existing employees as criterion, which are usually easy to generate and/or readily available. However, supervisors are humans, and their ratings may be biased. As a consequence, the usefulness of supervisor ratings as indicators of job performance can be challenged. For example, a supervisor may really like employees who chat about football, which then boosts these employees' performance ratings. Similar issues might occur when designing algorithm-based selection procedures. If the algorithm is trained on predicting supervisor ratings, it will likely learn from biases that supervisors inject into the rating. In the end, the algorithm selects applicants who like to watch football instead of focusing on job relevant skills and abilities.

Beyond supervisor ratings, other common performance indicators for individual employees involve scores regarding sales, number of successful actions or interventions, and customer satisfaction. Recently, new criteria have received attention from researchers and practitioners, namely *extra-role performance*, such as *organizational citizenship behavior* (e.g., helping co-workers), *work engagement* and *deviant behavior* (counterproductive work behavior).

These new criteria may all account for the fact that individual performance, which is most commonly the main criterion of most selection procedures, may not actually translate to organizational performance (Ployhart et al. 2017). For instance, employees showing best possible job performance, but at the same time leading to a negative climate in their teams, may consequently be of more harm for the organization than that they benefit the organization.

Furthermore, in selection research distinction is made between *maximal behavior* (how a person could perform) and *typical behavior* (how a person typically performs). Classical selection procedures, such as job interviews and assessment centers, often only assess applicants' maximal performance, as applicants try to create the best possible impression in such selection situations (Peck and Levashina 2017). This also implies that they may exhibit impression management behavior (e.g., they exaggerate their past achievements or behave unnaturally in the assessment center Peck and Levashina 2017). Therefore, these selection procedures might not really predict applicants' actual everyday job performance.

3.5 Conclusion

In this section, we introduced the job selection problem mainly from a psychological point of view. We highlight that it usually is a multi-hurdle approach aiming at finding the best suited applicant given a job description which includes the necessary KSAOs for a job. Selection approaches such as interviews should prove to be valid predictors of relevant criteria (e.g., job performance). In the next section, we will describe a use case of a potential new way of selecting applicants.

4 Use Case: An Explainable Solution for Multimodal Job Candidate Screening

In this section, we will discuss the data-driven 2017 ChaLearn Looking at People Job Candidate Screening Challenge (Escalante et al. 2017). Besides, as a use case, we will focus on a particular submission to this Challenge (Achmadnoer Sukma Wicaksana and Liem 2017) and its expansion (Achmadnoer Sukma Wicaksana 2017). In this work, the chosen solution was explicitly designed to be explainable and understandable to personnel selection experts in psychology.

In alignment with the overall themes of this chapter, the current section will particularly focus on discussions with respect to psychological and machine learning viewpoints on data-driven personnel selection and explainability. As a consequence, technical discussions will only be presented in summarized form; for further details, the reader is referred to the original introduction of the Challenge (Escalante et al. 2017), the overview paper of solutions submitted to the Challenge (Escalante et al. 2018), and the paper and thesis originally describing the solution presented as a use case here (Achmadnoer Sukma Wicaksana and Liem 2017; Achmadnoer Sukma Wicaksana 2017).

4.1 The Chalearn Looking at People Job Candidate Screening Challenge

The 2017 ChaLearn Looking at People Job Candidate Screening Challenge (Escalante et al. 2017)[2] is part of a series of data-driven 'Looking at People' Challenges, focusing on automated visual analysis of human behavior. For each

[2] http://chalearnlap.cvc.uab.es/challenge/23/description/

Challenge, an unsolved analysis problem is proposed, and for this problem, data and target labels are acquired at scale by the Challenge organizers. Subsequently, participant teams sign up to the Challenge, upon which they get access to training data (the data on which solutions are to be trained), as well as validation data (data which can be used for evaluation, while participants are refining their solutions), both also including ‘ground truth’ target labels. Participants will then propose a final system solution, that will be run on an evaluation dataset, for which the target labels were not released to the participants before.

The Challenge is run in *coopetition* format: on one hand, it is a competition in which centralized data sets are used for training, intermediate validation, and final testing. On the other hand, cooperation is possible and encouraged, as participants are required to openly share their solutions to the problem. As all participants had access to exactly the same data, the Challenge offers useful benchmarking insight, allowing different solutions to be compared against each other.

Following an earlier Challenge on apparent personality analysis (Ponce-López et al. 2016), the Job Candidate Screening Challenge focused on predicting apparent personality (the Big Five personality dimensions), as well as interviewability, from short video clips. These can be seen as ‘thin slices’ (Ambady et al. 2000), giving short but informative insight into a participant.

Given the importance of explainability in job candidate screening processes, the coopetition had both a *quantitative* stage and a *qualitative* stage. The quantitative stage was framed as a pure machine learning problem. For this, the Mean Absolute Error (MAE) was chosen as the evaluation metric, comparing the predictions made by proposed systems with the ‘true’ scores in the ground truth dataset. MAE comparisons between participant submissions were performed separately for each of the Big Five traits, as well as the interviewability score.

MAE is a common evaluation metric to measure accuracy for a continuous variable. It is a negatively-oriented score, meaning that the lower the score is, the better. It can be turned into a positively-oriented *accuracy* score by subtracting it from 1 (‘a perfect system’).

More precisely, MAE can be formulated as

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |p_i - g_i|$$

with N being the total number of video excerpts in the test set, p_i being the predicted value for the variable of interest, and g_i being the ground truth value. As a consequence, the Accuracy A can be formulated as

$$A = 1 - MAE.$$

In the qualitative stage, participants were instructed to focus on the explainability of interviewability scores. The required output for this stage was a textual description: it should explain both the workings of a chosen quantitative model, as well as the result of the prediction obtained by using this model. As for the choice of the quantitative model, participants could (re)use any of the solutions submitted to the quantitative stage, or propose a solution of their own. For the assessment of the qualitative textual descriptions, experts in psychological behavior analysis, recruitment, machine learning and computer vision were invited as jury members. Solutions were scored on a scale of 0 to 5 on five criteria:

- **Clarity**: Is the text understandable / written in proper English?
- **Explainability**: Does the text provide relevant explanations to the hiring decision made?
- **Soundness**: Are the explanations rational and, in particular, do they seem scientific and/or related to behavioral cues commonly used in psychology?
- **Model Interpretability**: Are the explanations useful to understand the functioning of the predictive model?
- **Creativity**: How original / creative are the explanations?

For further details on the Challenge setup and various participants' submissions, the interested reader is referred to the overview papers in (Escalante et al. 2017, 2018).

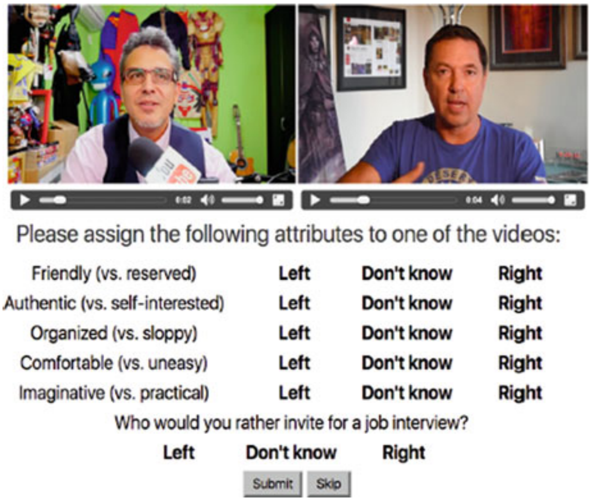
4.2 Dataset

The dataset for the Challenge was acquired as a corpus for first-impression and apparent trait analysis. For this, HD 720p YouTube videos of people facing and speaking English to camera were acquired. Care was taken that the dataset encompassed diversity on several properties, such as gender, age, nationality, and ethnicity. Only good-quality videos in which a unique adult person was facing the camera were considered; from these, at most six 15-second clips were generated for each video, which would not have visual or audio cuts in them. In the end, this yielded 10,000 15-second video clips. For the coopetition, 6000 of these clips were marked as training data, 2000 as validation data, and 2000 as test data, on which the final rankings would be obtained.

Besides the audiovisual video data, speech transcripts were provided for the Job Candidate Screening Challenge, transcribed by a professional transcription service which yielded 435,984 words (out of which 14,535 unique words), with 43 words per clip on average. A full data summary is given in Ponce-López et al. (2016).

Regarding the annotation of the video clips in terms of personality traits and interviewability, crowdworkers on the Amazon Mechanical Turk platform were provided with an online annotation interface involving pairs of 15-second videos, as shown in Fig. 3 (Ponce-López et al. 2016). The following instructions were provided to the crowdworkers:

> You have been hired as a Human Resource (HR) specialist in a company, which is rapidly growing. Your job is to help screening potential candidates for interviews. The company is using two criteria: (A) competence, and (B) personality traits. The candidates have already



Please assign the following attributes to one of the videos:

Friendly (vs. reserved)	Left	Don't know	Right
Authentic (vs. self-interested)	Left	Don't know	Right
Organized (vs. sloppy)	Left	Don't know	Right
Comfortable (vs. uneasy)	Left	Don't know	Right
Imaginative (vs. practical)	Left	Don't know	Right

Who would you rather invite for a job interview?

Left Don't know Right

Fig. 3 Interface of pairwise comparison to collect labels

> been pre-selected for their competence for diverse positions in the company. Now you need to evaluate their personality traits from video clips found on the Internet and decide to invite them or not for an interview. Your tasks are the following. (1) First, you will compare pairs of people with respect to five traits: Extraversion = Friendly (vs. reserved); Agreeableness = Authentic (vs. self-interested); Conscientiousness = Organized (vs. sloppy); Neuroticism = Comfortable (vs. uneasy); Openness = Imaginative (vs. practical). (2) Then, you will decide who of the 2 people you would rather interview for the job posted. (Ponce-López et al. 2016)

Not all possible video pairs were evaluated; instead, the small-world algorithm (Watts and Strogatz 1998) was used to generate a strategic subset of video pairs with good overall coverage, as it provides high connectivity, avoids disconnected regions in the graph, has well-distributed edges, and a minimum distance between nodes (Humphries et al. 2006). As a result, 321,684 pairs were obtained to label 10,000 videos. In order to convert pairwise scores to cardinal scores, the Bradley-Terry-Luce (BTL) model (Bradley and Terry 1952) was fitted using Maximum Likelihood estimation. Detailed explanations on how this can be done are described in (Chen et al. 2016). The final cardinal scores were set to be within the [0, 1] interval. Annotation reliability was verified through reconstruction; the reconstruction accuracy of all annotations was found to be over 0.65, and the apparent trait annotations were found to be highly predictive of invite-for-interview annotations, with a significantly above-chance coefficient of determination of 0.91 (Escalante et al. 2018).

Summarizing the descriptions above, for the quantitative stage of the Challenge, input data consists of 15-second video fragments (video and audio) and their corresponding textual transcripts. The associated target labels consider scores on each of the Big Five personality traits, as well as interviewability, which all were obtained through crowdsourcing: in all cases, these scores are a numeric value in the [0, 1] range.

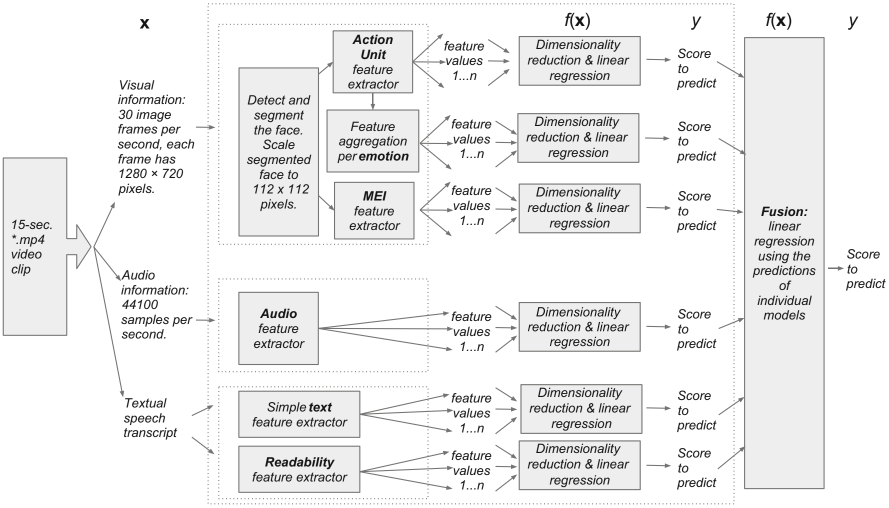
4.3 General Framework of a Potential Explainable Solution

As use case illustration of a potential explainable solution to the Challenge, the work of Achmadnoer Sukma Wicaksana and Liem (Achmadnoer Sukma Wicaksana and Liem 2017; Achmadnoer Sukma Wicaksana 2017) is presented here. This work was intended to provide an explainable machine learning solution to the data-driven job candidate screening problem, while explicitly keeping the proposed solution understandable for non-technical researchers and practitioners with expertise in organizational psychology. This was done by designing the system pipeline in consideration of common traditional methodological practice and focus points in job candidate screening (see Sect. 2). This way, the system was meant as an illustration to trigger discussions and collaborations across disciplines.

The overall system diagram for the proposed system pipeline is given in Fig. 4. The general framing closely follows an applied machine learning pipeline (similar to Fig. 1b), including an explicit feature extraction step. As such, the setup follows the second suggestion for potential integrations between psychological and machine learning setups, as outlined in Sect. 2.3.

The input data considers video, audio and text: for each of these, dedicated hand-crafted features are extracted from raw data in various modalities and categories. In other words, the authors proposed several types of information to be extracted from the raw visual, audio and textual data, which all should be understandable with respect to the job candidate screening problem. The details of the chosen categories will be further discussed in Sect. 4.3.1.

The choice to transform the raw data into hand-crafted features, rather than employing an automatically learned representation or an end-to-end learning setup



x
f(x)
y
f(x)
y
15-sec. *.mp4 video clip
Visual information: 30 image frames per second, each frame has 1280 × 720 pixels.
Audio information: 44100 samples per second.
Textual speech transcript
Detect and segment the face. Scale segmented face to 112 x 112 pixels.
Action Unit feature extractor
Feature aggregation per emotion
MEI feature extractor
Audio feature extractor
Simple text feature extractor
Readability feature extractor
feature values 1...n
Dimensionality reduction & linear regression
Score to predict
Fusion: linear regression using the predictions of individual models
Score to predict

Fig. 4 Overall system diagram for the work in Achmadnoer Sukma Wicaksana (2017)

(see Sect. 2.3), was explicit and deliberate. From an accuracy perspective, machine learning solutions employing an intermediate, hand-crafted feature extraction step typically do not perform as well as solutions which employ heavier automatic learning from raw data. However, as clear benefit, in a hand-crafted feature extraction step, the information extracted from the raw data is controlled and informed by the insight and interpretation of a human practitioner. As such, the explicit definition of features to be extracted in a machine learning pipeline can be seen as an alternative to the explicit choice of theory-driven independent variable dimensions in a traditional psychological setup.

Also regarding the choice of $f(\mathbf{x})$ (the model that relates the feature values to the dependent variable), it was taken into account that traditional psychological approaches would usually fit a linear regression model. In the current pipeline, this also was done, although in a slightly more elaborate setup than in traditional psychological practice.

First of all, rather than only employing Ordinary Least Squares estimation for the linear model fitting, various regression optimization variants were studied, as further explained in Sect. 4.3.2. Furthermore, a way had to be found to apply *fusion* of the information from different modalities and feature categories. For this, after training separate linear models per feature category for a dependent variable of interest, the predictions of each of these linear models were used as input to a second regression layer, in which a meta linear model was trained for the dependent variable of interest. This process was separately performed for each of the dependent variables relevant to the Challenge (the scores for each of the Big Five traits, and the interviewability score).

As will further be detailed in the following subsections, within individual feature categories, several dozens of feature dimensions were considered. The final regression step takes six values (one for each feature category) as input. From a traditional psychology perspective, this would be considered a relatively big regression, with many variable dimensions. In contrast, from a machine learning perspective, the approach uses unusually few dimensions: as also will be discussed in Sect. 4.3.3, it is not uncommon for machine learning pipelines to employ thousands of feature dimensions.

4.3.1 Chosen Features

The dataset contained information in several modalities: visual information in the video, audio information in the video, and textual information in the form of the speech transcripts.

In the visual modality, information relating to persons' facial movement and expression were considered: in various previous works, these were mentioned as good indicators for personality traits (Naumann et al. 2009; Borkenau et al. 2009; Waung et al. 2014). More specifically, regarding visual content, the open-source OpenFace library (Baltrušaitis et al. 2015) was used to detect and segment the face from frames in each video. Segmented face images were standardized to

Table 2 Action Units that are recognized by OpenFace and their description

Action unit	Description
AU1	Inner brow raiser
AU2	Outer brow raiser
AU4	Brow lowerer
AU5	Upper lid raiser
AU6	Cheek raiser
AU7	Lid tightener
AU9	Nose wrinkler
AU10	Upper lip raiser
AU12	Lip corner puller
AU14	Dimpler
AU15	Lip corner depressor
AU17	Chin raiser
AU20	Lip stretcher
AU23	Lip tightener
AU25	Lips part
AU26	Jaw drop
AU28	Lip suck
AU45	Blink

be 112 × 112 pixels. Beyond segmenting faces, OpenFace also offers a feature extraction library that can extract and characterize facial movements and gaze. Using this feature extraction library, the three visual feature sets were obtained: an Action Unit representation, an Emotion representation, and a Motion Energy Image representation.

Action Units (AU) are subcomponents of facial expressions, which both have been studied in psychology and social and affective signal processing, and which are encoded in the Facial Action Code System (FACS) (Ekman and Friesen 1978; Ekman and Rosenberg 2005). OpenFace is able to extract several of these AUs, as listed in Table 2, and indicate AU *presence* (indicating whether a certain AU is detected in a given time frame) and *intensity* (indicating how intense an AU is at a given time frame).

For each AU, three statistical features are derived for usage in our system, aggregating information from the different frames in the particular video. The first feature is the percentage of time frames during which the given AU was visible in a video. The second feature considers the maximum intensity of the given AU in the video. The third feature considers the mean intensity of the AU in the video. As 18 AUs are detected, with three features per AU, 52 features are considered in total for the Action Unit representation.

In affective analysis, combinations of AUs are usually studied. For example, Happiness is evidenced in a face when the cheeks are raised and the lip corners are pulled up. Therefore, AU combinations were hard-coded for the seven basic emotions (Happiness, Sadness, Surprise, Fear, Anger, Disgust and Contempt), as

Table 3 Emotions and their corresponding Action Units that construct them

Emotion	Action Units
Happiness	6 + 12
Sadness	1 + 4 + 15
Surprise	1 + 2 + 5 + 26
Fear	1 + 2 + 4 + 5 + 7 + 20 + 26
Anger	4 + 5 + 7 + 23
Disgust	9 + 15
Contempt	12 + 14

shown in Table 3. Then, the three statistical features as above were considered, but now aggregated over all AUs relevant to the emotion. This yields 21 features in total for the Emotion representation.

Finally, the resulting face segmented video from OpenFace was also used for a Motion Energy Image (MEI) representation. MEI is a grayscale image that shows how much movement happens on each pixel throughout the video, with white indicating a lot of movement and black indicating less movement (Bobick and Davis 2001). In order to capture the overall movement of a person's face, a Weighted Motion Energy Image (wMEI) is constructed from the resulting face segmented video. wMEI was proposed in the work by Biel et al. (2011) as a normalized version of MEI, by dividing each pixel value by the maximum pixel value.

For the construction of wMEI, it was important to use face-segmented video data, rather than unsegmented full frames. This is because in several cases, videos were recorded in public spaces or while the subject was moving. As a consequence, many pixels in the video corresponding to the background of the scene will also display considerable movement. By only considering face-segmented video data, the focus of analysis will be on the subject's true facial movement. As feature description of the wMEI image of a given video, several statistical features were chosen: the mean, median, and entropy.

For the audio, the focus was on prosodic features, capturing emphasis patterns during speaking. In previous work (Biel et al. 2011), these also were shown to correlate with personality traits. Paralinguistic speech emphasis patterns, which give insight into the tone of voice, have been recognized to be powerful social signals (Nass and Brave 2005). For this work, speech features were extracted using the MATLAB toolbox developed by the MIT Media Lab (Pentland 2004; Caneel 2005). The features that were used are listed in Table 4; in all cases, the mean and standard deviation over the full video's audio were used. As a consequence, 12 features were used here in total.

Based on findings in organizational psychology, personality traits are not the only (and neither the strongest) predictors for job suitability and hiring decisions. As mentioned in (Schmidt and Hunter 1998), for example, General Mental Ability (GMA) also is both a valid and strong predictor for job performance.

While formal GMA assessments were not available for subjects in the Challenge dataset, it was considered that language use may indirectly reveal GMA

Table 4 Audio features and their description

Audio features	Description
F0	Main frequency of audio
F0 conf.	Confidence of F0
Loc. R0 pks	Location of autocorrelation peaks
# R0 pks	Number of autocorrelation peaks
Energy	Energy of the voice
D Energy	Derivative of the energy

characteristics, such as the use of difficult words. Therefore, for the textual video transcripts, features were chosen that would capture the comprehensiveness and sophistication of speech.

Two categories of textual features were considered. First of all, speaking density was approximated by two simple measures: total word count and the number of unique words spoken in the video. Furthermore, linguistic sophistication was approximated by calculating several Readability indexes over the spoken transcripts: ARI (Smith and Senter 1967), Flesch Reading Ease (Flesch 1948), Flesch-Kincaid Grade Level (Kincaid et al. 1975), Gunning Fog Index (Gunning 1952), SMOG Index (McLaughlin 1969), Coleman Liau Index (Coleman and Liau 1975), LIX, and RIX (Anderson 1983), as implemented in an open-source contributed library for the Python NLTK toolkit. Each of these Readibility indexes stemmed from existing literature, targeted at quantitative assessment of the reading difficulty level of a given text.

4.3.2 Regression Model

Prediction of the dependent variable scores was done through regression. Given the large amount of derived related features (for example, multiple alternative Readability indexes), multicollinearity between input variables is likely to occur. This is undesirable, as the considered feature dimensionality may be higher than the true dimensionality, considering independent components. Furthermore, if a regression model is fitted with highly correlated features as input, it becomes harder to determine the effect per individual feature on the end result.

In order to mitigate the effect of multicollinearity in the model, several techniques were considered. The first one used *Principal Component Regression* (PCR): employing the prominent Principal Component Analysis (PCA) technique before feeding the results to Ordinary Least Squares (OLS) Regression. Next to this, Ridge and Lasso Regression were considered, which incorporate $l2$ and $l1$ regularization technique on the linear regression model, respectively.

PCA is a linear transformation that converts a set of correlated variables into uncorrelated variables called principal components. This technique also ensures that the highest principal component accounts for the highest variation of data. Thus, by selecting several principal components, data variation over the most important principal component dimensions is maintained, while the amount of dimensions to work with reduces significantly. The transformation from original feature vectors to new principal components can be expressed as a linear matrix multiplication:

$$Y = X * W$$

where X is the original feature matrix, having N rows of K-dimensional observations, W is the linear transformation matrix, with K eigenvectors of M dimensions, and Y is the transformed feature matrix, expressing the same N observations as M principal components.

These principal components then will be fed as input to OLS Regression. This regression technique is a simple linear regression technique that estimates the coefficients by minimizing a loss function with a least squares method:

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \| y - X\beta \|_2^2 .$$

The other two regression models that are considered for the system incorporate a penalizing function to the least squares regression model. By doing so, they try to shrink coefficients, so that the significance of a subset of input features will be eminent by the value of the coefficients. Coefficient estimation for Ridge and Lasso regression is conducted as follows:

$$\hat{\beta}^{Ridge} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\| y - X\beta \|_2^2}_{\text{Loss}} + \lambda \underbrace{\| \beta \|_2^2}_{\text{Penalty}}$$

$$\hat{\beta}^{Lasso} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\| y - X\beta \|_2^2}_{\text{Loss}} + \lambda \underbrace{\| \beta \|_1}_{\text{Penalty}}$$

with λ expressing the tuning parameter. When λ is equal to zero, this becomes a least squares regression; when λ is infinitely large, the $\hat{\beta}^{Ridge}$ is 0. For other values of λ, a balance is taken between fitting a linear model and shrinking the coefficients.

These three regression models were considered both for the individual feature category modeling, as well as for the fusion step.

4.3.3 Quantitative Performance

For understanding the quantitative performance aspects of the system, two experiments were done. First of all, it was assessed which of the three regression techniques would perform best. Secondly, regarding input features, it was assessed whether all features should be used in the system, or only those that through an initial correlation analysis were revealed to be significant ($p < 0.05$) with respect to the dependent variable to be predicted.

Table 5 Comparison of quantitative performance (accuracy) between the system described as use case in this chapter (Achmadnoer Sukma Wicaksana 2017), an earlier version of the system presented at the ChaLearn workshop (Achmadnoer Sukma Wicaksana and Liem 2017), and two other proposed solutions for the ChaLearn Job Candidate Screening Challenge

Categories	Use case system	Earlier version	Gorbova et al. (2017)	Kaya et al. (2017)
Interview	0.8950	0.8877	0.894	0.9198
Agreeableness	0.9008	0.8968	0.902	0.9161
Conscientiousness	0.8873	0.8800	0.884	0.9166
Extraversion	0.9001	0.8870	0.892	0.9206
Neuroticism	0.8945	0.8848	0.885	0.9149
Openness	0.8991	0.8903	0.896	0.9169

From the experimental results, which are reported in detail in Achmadnoer Sukma Wicaksana (2017), the best-performing regression technique differed per situation, although the absolute differences in accuracy for the different regression techniques were very small. As for the choice of feature sets, slightly better results were obtained for using full feature sets, rather than pre-selected feature sets resulting from correlation analysis. Full configurations and detailed performance tables can be found in Achmadnoer Sukma Wicaksana (2017).

Quantitative accuracy performance scores of the final, optimized system are reported for all dependent variables in Table 5. For comparison, the table also reports system performance on an earlier published version of the system (Achmadnoer Sukma Wicaksana and Liem 2017) (which used a smaller feature set and did not yet optimize regression techniques). Furthermore, performance scores are reported for two other proposed solutions: the work in Gorbova et al. (2017), employing similar features to ours, but with a multi-layered perceptron as statistical model; and the work in Kaya et al. (2017), which obtained the highest accuracies of all participants in the quantitative Challenge.

This latter work employed several state-of-the-art feature sets, some of which resulting from representations learned using deep neural networks, with considerably higher dimensionality than our features (thousands of feature dimensions). While the system described in this chapter does not outperform the scores of Kaya et al. (2017), performance differences are small in the absolute sense, at the benefit of an easily understandable model with simple regression architectures, and a much small number of feature dimensions.

4.4 Opportunities for Explanation

For the qualitative stage of the Job Candidate Screening Challenge, a textual explanation to accompany a quantitative prediction had to be automatically generated. For the system described in this chapter, the decision was made to generate an extensive report, displaying an explanation and a contextualization of measured values corresponding to each feature used in the system.

The choice was made to describe each feature, and not to make a more optimized textual summary that would pre-filter descriptions of particular variables. This was done, as the authors felt that in a real-life setting, a practitioner with domain knowledge should have the freedom to choose whether to see a full report, or only parts of it. Furthermore, the authors wished to avoid that any information would inadvertently be hidden from an end user, while an end user may actually have interest in it. Indeed, as will be discussed in Sect. 5.2, perceived controllability of an algorithmic solution is an important requirement for making it acceptable for end users.

As all features in the system were chosen to be humanly interpretable, a short human explanation was made for each feature, that was printed in the report. Furthermore, as an early screening scenario was adopted, the purpose of the explanation would be to allow for a selection of interviewable candidates to be made from a larger candidate pool. Therefore, for each feature, the score of a candidate for this feature was contextualized against 'what usually would be observed': in this case, the minimum and maximum feature values obtained on the pool of 6000 earlier rated subjects in the training set. Furthermore, it was indicated at what percentile the current video's score would be with respect to the training set candidates, to further give a sense of how 'usual' this person's observed feature value was.

As all dependent variable score predictions of the system are based on linear data transformations, the weight of each input feature dimension with respect to the final prediction model can easily be traced back. This information was not used for selecting or prioritizing information. However, for those feature values that had the strongest absolute weights with respect to the final prediction, the report would indicate whether this feature value would correlate positively or negatively with the dependent variable.

A sample excerpt from a generated report is given in Fig. 5. For possible future work, it will be interesting to develop a more user-friendly presentation of the descriptions, in connection to dedicated user interaction optimizations.

4.5 Reflection

The ChaLearn Challenge is in many ways interesting to the job candidate screening problem. Generally, the Challenge outcomes suggest that each of the personality characteristics, as well as the interviewability score, can be predicted with high accuracy using algorithmic procedures. At the same time, when aiming to connect these findings to psychological practice, there still are important open questions that will need more explicit attention, in particular regarding validity. For example, a major question to be asked is *what kind of information the ground truth scores truly indicated*.

```
******************
* USE OF LANGUAGE *
******************

Here is the report on the person's language use:


** FEATURES OBTAINED FROM SIMPLE TEXT ANALYSIS **
Cognitive capability may be important for the job. I looked at a
few very simple text statistics first.

*** Amount of spoken words ***
This feature typically ranges between 0.000000 and 90.000000. The
score for this video is 47.000000 (percentile: 62).
In our model, a higher score on this feature typically leads to a
higher overall assessment score.
```

Fig. 5 Example description fragment

Regarding validity of the dependent variable scores, the use of crowdsourcing to get non-expert first-impression annotations at scale is interesting. Considering the findings on observer judgment vs. self-reporting in Sect. 3.3, crowdsourcing could be a useful way to get observer judgments at scale. While crowdsourcing allows for reaching a population with higher diversity than the typical WEIRD (Henrich et al. 2010) samples (Behrend et al. 2011), crowdwork usually is offered in a marketplace setting, in which anyone interested in performing a task and meeting the task's qualifications can do so. This means that certain workers may perform many ratings in a batch, but others may only perform a single annotation task and then move on, causing potential annotator biases within the data that are hard to control up front.

Typically, crowdworkers would also perform work for monetary reasons, and would only be willing to spend little time on a single task, meaning that the tasks should be compactly presented. This is also evidenced in the way the annotation task was presented (see Fig. 3): only a single question is asked per personality trait. Even if this question may have come from a psychometrically validated instrument, there are more underlying facets to a psychological trait than the single question currently being posed. Fully equating the currently posed item questions with the underlying trait (e.g., considering that 'Extraversion == Friendly (vs. reserved)') would not be logical to a psychologist, and this choice should be explicitly defended. While the requirement for crowdsourcing tasks to be compact makes it unrealistic to employ full-length instruments, it still is possible to employ more than one item per trait, and it should be investigated whether doing so will yield higher psychometric reliability and validity.

While it was reported that the personality trait scores were highly predictive for the interviewability score (Escalante et al. 2018), another concern involves potential response bias. Looking at the annotation task, all items were presented with the positive valuation on the left, and the negative valuation on the right. This, together with the pairwise setup, may invite annotators to consistently prefer the person ‘they like best’. It is not guaranteed that the commonly advised strategy of reverse wording (varying positively and negatively phrased items) will truly yield better results (van Sonderen et al. 2013); especially in a crowdsourcing setup, in which workers may be focused on finishing the task fast, high attention to wording variations is not guaranteed. However, this aspect should be researched more deeply.

Looking at the ChaLearn data, especially at what kinds of videos score particularly high and low on each of the traits and the interviewability score (as shown in Table 6), one may wonder whether the first impression ratings may alternatively be interpreted as youthful attractiveness ratings. Again, this may be a consequence of the preference-oriented setup of the annotation task.

Escalante et al. (2018) analyzed potential judgment biases in the data regarding ethnicity, race and age, and found low-valued but significant positive biases towards judgment of female subjects on all personality traits except for Agreeableness, and low-valued but significant negative biases towards judgment of African-American subjects. Further analyses on potential age biases indicate that the youngest and oldest people in the dataset (estimated age under 19 or over 60) had below-chance probabilities for interview invitations, and that within the ‘common working age’ range, younger women and older men had higher prior probabilities of interview invitations. Perfectly performing systems trained on this data will therefore inhibit the same biases, and explicit awareness of this is needed.

Finally, it should be remarked that the data did not consider official job applications, but rather the general impression that candidates would leave in a more spontaneous setting. In a real application setting, a broader set of KSAOs will be of relevance, and not all personality traits may be equally important to job performance. Therefore, again, the interviewability assessments should at present most strongly be interpreted as preference ratings, rather than true invite-to-interview probabilities.

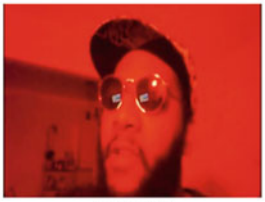
5 Acceptability

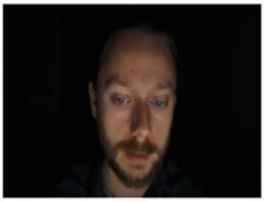
So far, explainability in the context of job candidate screening has solely been considered with respect to scientific stakeholders: computer scientists and psychologists interested in data-driven technologically-supported solutions. However, when implementing novel personnel selection approaches, there are two further stakeholders that need special attention: *applicants* and *hiring managers*.

Applicants are affected by novel personnel selection procedures, as their information and job application will be subject to the novel procedures. Second, hiring managers need to decide which personnel selection procedures are adequate to select applicants for a job.

In this section, research on the acceptance of novel selection technologies by applicants and hiring managers is therefore discussed, as understanding main interests and concerns of these stakeholders will be paramount in successfully implementing novel selection technologies in practice.

Table 6 Snapshots of videos with high and low values for each dependent variable of interest to the quantitative state of the ChaLearn challenge

Traits	Extraversion	Agreeableness	Conscientiousness
			
score	0.046729	0.000000	0.048544
score	0.925234	0.912088	0.951456

Traits	Neuroticism	Openness	Interview
			
score	0.031250	0.111111	0.149533
score	0.937500	0.977778	0.915888

5.1 Applicants

Most research on applicants' acceptance of personnel selection procedures was and still is influenced by Gilliland (1993), who proposed a model for the justice of selection procedures. In his model, he highlighted the importance of formal (e.g., job relatedness), interpersonal (e.g., interpersonal treatment) and transparency related characteristics (e.g., honest during the selection process) but also distributive justice (e.g., fairness of outcomes) of selection procedures on the overall acceptance of these procedures. Additionally, he pronounced that all of these variables consequently affect applicants' self-perceptions (e.g., self-esteem), reactions to the organization (e.g. organizational attractiveness) and eventually later job performance. Based on his model, scales to measure acceptance of selection procedures were developed (e.g., Bauer et al. 2001) and a tremendous amount of research supports the importance of examining acceptance of selection procedures (Chapman et al. 2005).

Unfortunately, research about acceptance of novel technologies for personnel selection lags at least 10 years behind current technological possibilities (Ployhart et al. 2017). To be clear, in the last two decades, most research focused on the acceptance of technology-mediated job interviews see Blacksmith et al. (2016) or Bauer et al. (2006). Just recently, acceptance research has called for studies using more up-to-date technologies (Ployhart et al. 2017) which was answered by Langer et al. (2017b) who found that an algorithm-based job interview including automatic analysis of social behavior (e.g., smiling) and a virtual agent as interviewer is less accepted than a videoconference interview with a human interviewer. More specifically, they found that lower transparency and interpersonal warmth of the algorithm-based procedure decreased its acceptance.

In the context of algorithm-based selection procedures, Gilliland's model in combination with findings from the study of Langer and colleagues and research about more classical technology-enhanced selection approaches can shed light on variables influencing acceptance of algorithm-based selection procedures. More precisely, applicants who are confronted with algorithm-based selection procedures will likely be concerned about *formal characteristics*, *interpersonal characteristics*, and *transparency-related characteristics* of a selection procedure.

First of all, applicants who are screened by any kind of algorithm-based personnel selection approach will be concerned about *formal characteristics* of the procedure. In the terms of Gilliland, these would be perceived job relatedness of the procedure, applicants' opportunity to perform (i.e., applicants' opportunity to show their skills and abilities) and objectivity (i.e., objective treatment during and results of the selection procedure). Regarding job relatedness, if it is obvious for applicants that a selection procedure is relevant to predict job performance, it will be accepted. In the case of algorithm-based selection procedures, there are approaches that appear more job related than others. For instance, using web scraping and machine learning approaches to scan through applicants' social media profiles may appear less job related than a serious game which mimics the aspects of a job and measures actual behavior during the game.

Similar examples are useful to understand that some selection procedures offer more opportunity to perform than others. It may be hard for applicants to put their best foot forward when an organization uses their social media information to evaluate applicants' job fit, whereas algorithm-based job interview solutions could at least appear to provide more opportunity to show one's skills. Compared to classical job interview procedures, however, algorithm-based procedures may provide less perceived opportunity to perform, as applicants do not really know how they can influence the algorithm in a way that it will positively evaluate their performance (Langer et al. 2017b). In the case of objectivity, algorithm-based solutions could even possess advantages over classical selection procedures, as automatically evaluated resumes or job interviews might be less prone to subjective human influence (e.g., applicants attractiveness (Gilmore et al. 1986)). However, as discussed in Sect. 4.5 of this chapter, algorithms themselves might have learned from human biases and consequently not be more consistent than human hiring managers (Caliskan et al. 2017).

Second, *interpersonal characteristics* of selection procedures influence their acceptance. For instance, the behavior of hiring managers can positively influence applicants' willingness to accept a job offer (Chapman et al. 2005). In the case of algorithm-based personnel selection, applicants might be concerned that human influence is minimized, such that there is no representative of the organization taking his or her time to at least look at their application. Applicants may perceive this as a signal of lower appreciation, thus detrimentally affecting acceptance (Langer et al. 2017b). However, positively influencing interpersonal characteristics of algorithm-based selection procedures appears to be challenging. An idea could be to add virtual agents to the algorithm-based selection situation (e.g., in the case of algorithm-based job interviews). However, the results of Langer and colleagues show that this does not seem to entirely solve the problem, and instead introduces new issues, such as negative feelings against the virtual character (which might be caused by the uncanny valley (Mori et al. 2012)).

Third, *transparency-related issues* seem to relate to applicant reactions. In the sense of Gilliland, a procedure is transparent if applicants are treated honestly, if they receive information about the selection procedure, and if they receive timely and helpful feedback about their performance. It is worth mentioning that the acceptance variables Job relatedness, Opportunity to perform, and Objectivity might all be affected by transparency: for a transparent procedure, it is more obvious if it is job related, if it is possible to show ones skills and abilities, and to evaluate if it treats applicants objectively. More precisely, applicants in a transparent selection procedure know which decision criteria underlie the selection decision; furthermore, if rejected, they receive information about why they were rejected. In the case of algorithm-based selection, it is not yet commonly made explicit which input variables led to a certain outcome (e.g., a rejection). Therefore, it would be impossible to derive any explanation about which decision criteria were involved. As a consequence, applicants do not know what is expected of them, neither do they know how to improve if they were rejected. In an attempt to increase acceptance of algorithm-based selection tools, incorporating ideas generated in the field of explainable artificial intelligence (Biran and Cotton 2017) will therefore be useful.

At the same time, Langer and colleagues (Langer et al. 2018) tried to improve transparency of an algorithm-based selection procedure through provision of information about an algorithm-based job interview procedure (e.g., about technical details, and about what an algorithm-based selection procedure is looking for). In the end, participants were positively and negatively affected by this information, indicating that the relation between transparency and acceptance is not just a simple ‘the more the better’ relation. Instead, it seems that transparency consists of different aspects that need to be addressed in order to understand its influence on acceptance. More precisely, transparency consists of technical details about the selection procedures (e.g., which data are used), justifications of the selection procedure (i.e., why exactly this procedure should be job relevant). Future research should try to reveal other aspects require consideration in order to understand the impact of transparency on acceptance.

5.2 Hiring Managers

In addition to applicants’ view on personnel selection situations, the perspective of hiring managers, which is closely related to the perspective of organizations (Klehe 2004), needs attention, as they are the ones who will be requested to select an applicant based on the information they receive from any type of screening tool. Additionally, they are also the ones who might be afraid of algorithm-based tools, making them superfluous in personnel selection contexts. For the means of raising acceptance of algorithm-based selection tools, it should therefore be an important step to include hiring managers’ opinions and ideas about novel selection devices. Based on previous research (Chapman et al. 2003; Klehe 2004; König et al. 2010), it is suggested that hiring managers evaluate algorithm-based selection tools considering the tools’ perceived *usefulness*, *objectivity*, *anticipated applicant reactions*, *probability of legal actions*, *controllability*, and *transparency*.

Hiring managers expect novel personnel selection methods to be useful to support their everyday work (Chapman and Webster 2003). In the case of algorithm-based tools, *efficiency* is the first thing that comes to mind, as these tools may have the potential to quickly screen many applicants. Especially as the use of technology has increased the applicant pool for many organizations, algorithm-based screening tools helping to manage the large amount of applications seem to be a logical solution. Additionally, hiring managers seem to be attracted by easy-to-use selection tools (Diekmann and König 2015) which should be considered when trying to improve acceptance of algorithm-based screening tools. More specifically, easy-to-use seems to imply easy to apply, easy to understand, and easy to interpret (Diekmann and König 2015).

Second, hiring managers hope for *enhancing objectivity* of selection procedures when implementing novel technologies. For instance, Chapman and colleagues (Chapman and Webster 2003) propose that by reducing human influence on selection situations, adverse impact (i.e., discrimination of minorities (Hough et al. 2001)) and human biases (e.g., better ratings for more attractive applicants (Gilmore et al. 1986)) might be reduced. Therefore, if an algorithm-based tool can actually prove that it is able to increase objectivity of selection situations, hiring managers will appreciate this fact.

Third, hiring managers seem to anticipate *applicant reactions* towards novel selection tools when considering to implement these tools (Klehe 2004). If hiring managers conclude that applicants may not like a novel selection procedure, it is less likely to be used for future selection procedures. As we have seen in the section on applicant reactions, they actually cover a wide range of different acceptance variables. Currently, it is still unclear which applicant reaction variables hiring managers consider to be most influential. Nevertheless, this makes it clear that algorithm-based tools do not only need to appear adequate to applicants, they also need to appear reasonable in the eyes of hiring managers.

Fourth, the *probability of legal actions* is closely related to applicant reactions: when applicants react extremely negatively to selection procedures, they might even sue the hiring company (Bauer et al. 2001). In the case of algorithm-based selection tools, legal actions seem possible, especially when an organization cannot prove the algorithms' validity and objectivity in the sense of preventing adverse impact (Klehe 2004). Generally, following the European General Data Protection Regulation (Council of the European Union 2016), applicants will also have the right to demand insight into how their data is processed by algorithmic procedures. In the absence of empirical studies relating to these issues, it seems to be hard for organizations and for developers of algorithm-based selection tools to support validity and to provide evidence for unbiased evaluations made by the algorithm.

Regarding validity, there are studies showing that algorithm-based tools correlate with personality (Campion et al. 2016) or with job interview performance (Naim et al. 2015) but empirical findings regarding its predictive validity for actual job performance or other important outcomes influencing organizational performance (e.g., organizational citizenship behavior: employees' positive behavior at work) are scarce. Regarding biases in the evaluation of applicants, recent research indicates that this might be a problem, as algorithms can learn from human biases (Caliskan et al. 2017). It is therefore necessary to not only evaluate the predictive validity of the algorithm, but also its development process, in particular its training procedure, in order to realize whether there could be any bias in the training data that may result in biased applicant scoring.

Fifth, *controllability* (i.e., being able to control a selection situation) could be hard to achieve when using algorithm-based tools. For instance, the scorings and rankings of applicants performed by algorithms may be used for a fully automated pre-screening, but in this case, there is less controllability for hiring managers, which often is unacceptable. Algorithms should therefore offer the possibility to regain control over the decision, when hiring managers want this option. For instance, it might be possible to develop algorithms in which hiring managers can choose to which aspects of applicants they attach more importance (e.g., personality, cognitive ability).

In the context of controllability, it is further important to note that perceived controllability of algorithm-based tools will likely be lower, if hiring managers have the impression that this tool will replace them in any way. Therefore, it should be clear what the algorithm is intended to do in the selection process—generally, a full replacement solution will not meet acceptance, but rather, the algorithm should support and simplify the work of hiring managers.

Sixth, an antecedent of all the aforementioned conditions for a positive evaluations of algorithm-based selection tools is *transparency* of the procedure. If a tool is transparent to hiring managers, it is easier to evaluate its usefulness, its objectivity, anticipate applicant reactions and the possibility for legal actions, and to assess its controllability (Langer et al. 2017b). In this case, transparency would mean that the process in which applicants are evaluated should be *comprehensible* (i.e., it is clear which characteristics and behavior of applicants will be used for their evaluation), *traceable* (i.e., it is possible to have an insight into why one applicant was preferred over another) and *explainable* (i.e., it is possible for hiring managers to formulate feedback to applicants about why they were rejected).

The previous discussion makes it clear that applicants' and hiring managers' acceptance of technology-supported tools can be affected by many different variables; not all of these necessarily relate to the algorithms or technology themselves. In the following and final section, we will discuss where, within the technological realm, acceptability can be fostered and stimulated.

6 Recommendations

In previous sections, we have introduced the job candidate screening problem, as well as common methodologies and viewpoints surrounding this problem, perceived by various scientific disciplines and stakeholders. It is undisputed that explainability is important in the context of algorithmic job candidate screening, and technologically-supported hiring in general. It even may be critical for allowing true interdisciplinary collaboration. However, following the discussions throughout this chapter, it becomes clear that 'explainability' in job candidate screening can actually have many different interpretations, and is relevant to many different parties.

As discussed in Sects. 2 and 3, for psychologists, explainability will closely relate to understanding what variables are given to the system, whether their inclusion is supported by evidence and theory, and to what extent these variables have been collected using reliable procedures. As discussed in Sects. 2 and 4, for computer science researchers with machine learning interests, explainability will mostly lie in understanding why and how an algorithm will learn certain patterns from data. Finally, as discussed in Sect. 5, for applicants, algorithmic explainability will mostly deal with formal characteristics and transparency-related characteristics (interpersonal characteristics being a matter of presentation), while for hiring managers, explainability will be desired regarding usefulness, objectivity, anticipated applicant reactions, probability of legal actions, controllability, and transparency.

Against these considerations, in this section, we will make several recommendations on how technologically-supported job candidate screening mechanisms can be improved in ways that will raise their acceptance and foster interdisciplinary collaboration, considering all relevant stakeholders.

6.1 Better Understanding of Methodology and Evaluation

6.1.1 Stronger Focus on Criterion Validity

In early selection procedures, the scoring of candidates will focus on interviewability: the decision of whether or not this candidate should more closely be screened in person by representatives of the entity that is hiring. At the same time, the selection procedure is actually intended as a way to assess future job performance. As such, this aim should be clearly reflected in the procedure and the resulting scores.

At the same time, generally, as discussed in Sect. 3, there is no single definition of what 'good job performance' exactly means. A more comprehensive set of variables may need to be assessed here (e.g. not only individual performance, but possibly also organizational citizenship behavior). We expect that exposing these variables transparently to all stakeholders throughout the process will increase trust in the overall system. As another suggestion, it may be useful to more explicitly include validated KSAOs in automated prediction setups.

In machine learning settings, ground truth labeling and further data annotation tasks are commonly done through crowdsourcing. However, most annotation validation methods focus on reliability (high inter-rater agreement, clear majority votes, accurate reconstruction), but not on validity. While this is less of an issue for objectively verifiable phenomena in the natural world, this is problematic in the case of constructs which are not directly observable. To ensure validity, it is advisable to consider psychometric principles when setting up the instruments to solicit the necessary input from humans. The work by Urbano et al. (2013) on evaluation in music information retrieval gives further useful examples on how comprehensive evaluation, including verification of validity, can be done in computational settings which partially rely on human judgment.

6.1.2 Combining Methodological Focus Points

In machine learning, main attention will be given to $f(\mathbf{x})$, the mapping from input to output, while in psychology, the main attention is given to ensuring the independent variables $\mathbf{x}$ are explainable given evidence and existing theory. Psychologists also are interested in searching for *mediator* (variables mediating the influence of a predictor on an outcome) and *moderator* variables (variables influencing the relation between other variables), while in machine learning, paying detailed human attention to individual input data dimensions is often irrelevant, also as the input data is usually at semantically lower levels.

As a consequence, while in popular discourse on technologically-supported hiring, the question tends to emerge 'whether human psychologists or algorithms do a better job at candidate assessment', this question does not make much methodological sense. Considering the value of anticipated applicant reactions, probability of legal actions, controllability, and transparency to a hiring manager, as well the desire of applicants for interpersonal relations, the expertise of a human who is knowledgeable about hiring cannot be omitted and replaced by a machine.

There is interest from both the psychology and computer science/machine learning domains to connect their methods to provide better solutions. As mentioned in Sect. 2.3, data-driven methods can be integrated with the psychological prediction pipeline at several points. They may offer useful and better alternatives to common linear regression models, inform feature engineering, offer data-driven alternatives for traditional measurement instruments, or can be employed in end-to-end learning. It is possible to define explicit feature extraction steps to extract relevant information from raw data; alternatively, relevant—but usually less interpretable—mappings can automatically be learned from the data.

In terms of expected controversy, it will not be controversial, and easily adoptable, to integrate machine learning methods in a traditional psychological pipeline, as an alternative to the common linear regression model. The other way around, a main interesting challenge for machine learning applied in psychological settings is to ensure that information in the prediction pipeline is psychologically informed. One way to do this, as also performed in the system discussed in Sect. 4, would be to employ the extraction of hand-crafted features from raw signals, even though they will be at a semantically lower level than common psychological instruments and vocabularies.

It will be interesting to consider offering data-driven replacements of traditional measurement instruments. However, in this case, it is important to carefully integrate theory and psychometrically validated findings in the data and target label preparations. While hand-crafted feature extraction is considered old-fashioned in machine learning, it is useful to ensure human interpretability of information extracted from raw signals.

If an explicit feature engineering step is omitted, and there rather would be interest in direct representation learning of equivalent outcomes to a traditional measurement instrument, the advantage would be that the first extracted representation will have a well-known form to a psychologist (e.g., a predicted Big Five trait score). At the same time, with the information extraction procedure in representation learning falling fully on the side of a machine learning algorithm, extreme care should be taken: systems do not always learn what they are supposed to learn, but may inadvertently pick up on other patterns in the data (Sturm 2014). To mitigate this, it is important to consider various concurrent facets of the problem in the representation learning procedure, and perform careful and extensive validation, as e.g. performed in Kim et al. (2018).

Given the psychological emphasis on understandable data and constructs, the most controversial integration would be to apply end-to-end machine learning approaches in psychological settings. These will not likely allow for meaningful collaborations, as directly learning mappings from raw data to a dependent variable will not be deemed meaningful to a psychologist, due to the lack of clear interpretable variables underlying the prediction procedure.

6.2 Philosophical and Ethical Awareness

Psychology belongs to the social sciences, while computer science belongs to the natural sciences. In combining the two worlds, viewpoints and validation techniques from both these sciences will need to be bridged: the previous subsection has already discussed several ways in which this may be done.

The differences between theory-driven methodology in psychology and data-driven approaches in computer science touch upon philosophical epistemological debates. When formulating theories and hypotheses, do we miss out on important information, or pick what we want to see without solid foundations? At the same time, when blindly and only trusting data, how solid is this data really, and is it justified to fully give up human interpretation?

The difference has also been discussed and debated within the natural sciences, with several authors pointing out that theory will keep playing an essential role, while data at the same time can help in revealing unexpected effects, disproving earlier beliefs, or steering discovery towards theories we did not think of yet (Mazzocchi 2016; Bar-Yam 2016).

A major concern regarding machine learning in the context of job candidate screening has been bias. Algorithms do not have an ethical compass by themselves; if training data reveals undesired societal biases, this will be mirrored in any machine learning solution built on top of this data. For example, if a machine learning model intended to assess potential CEOs will be trained on data from the first half of the twentieth century, it may infer that being a Caucasian male is a necessary condition in order to be deemed a suitable CEO.

These are undesirable effects, and the machine learning community has started focusing on blind spots and algorithmic improvements to ensure fairness, reduce bias, and avoid that certain population subgroups will be disadvantaged through algorithmic means (Dwork et al. 2012; Bolukbasi et al. 2016; Buolamwini and Gebru 2018). At the same time, it should be emphasized that societally undesired effects of algorithmic procedures typically occur because of biased input data, or because of the algorithmic predictions being unjustly held for the absolute truth. The sensibility to deal with this is a shared responsibility between machine learning experts, domain experts regarding the data, and stakeholders involved with practical implementation. In this sense, algorithms may suitably be used as ‘mirrors’ to reflect on whether predicted outcomes indeed align with their purpose in the broader context of socio-technical social systems (Crawford and Calo 2016).

6.3 Explicit Decision Support

For many stakeholder parties, having the opportunity for human control in technologically-supported predictions concerning human beings is an important requirement for acceptability. It already was argued that rather than considering technologically-supported solutions as full replacement of a human being, they should rather be considered as complementary assisting tools for decision support. This aligns with the recent proposition by Barabas et al. (2018) to consider algorithmic predictions as indicators of intervention opportunities, rather than binding predictions. We foresee similar opportunities in job candidate screening: as discussed in the previous subsection, algorithms can assist in pinpointing bias and unfairness issues in data, before full decision pipelines are based upon them. Furthermore, they can usefully help in scaling up the early selection stage; however, this mostly would be to provide a selection of potentially interesting job candidates to a human assessor. As such, only a rough first selection may be needed; rather than seeking a full and 'true' ranking and scoring, the only thing that matters may be that a candidate would fall in the upper quadrant of the candidate pool. If so, evaluation metrics should be adjusted accordingly.

6.4 The Goal of Explanation

As discussed throughout this chapter, the need for explanation may lie at different points for researchers in psychology and machine learning, for job applicants, and for hiring managers.

Regarding the academic perspective on 'how good' a prediction model is, a balance between accuracy and explainability needs to be found. Baseline models can be improved in accuracy by increasing model complexity; at the same time, this makes the model's working less understandable for humans. While a model that clearly fails in finding the best applicants will never be accepted, there might be a point at which increasing accuracy does not bring that much benefit, and better comprehensibility will be favored over pushing accuracy another percent.

Throughout the discussions between co-authors in preparing this chapter, we found that literacy regarding each others' methodologies was a first necessary starting point. If the job candidate screening problem should be tackled from an interdisciplinary or transdisciplinary perspective, psychologists will need to gain basic computer science and machine learning literacy, while computer scientists will need to deepen their knowledge on psychometric validation. Preferably, curricula should be developed that do not only train interdisciplinary literacy, but also hands-on basic skills.

In ongoing discussions on explainability in machine learning, common counter-arguments against explainability are that 'humans beings cannot explain their own reasoning processes well themselves' and 'if it works, it just works'. Considering explainability in the context of technologically-supported job candidate screening methods for hiring managers and candidates, an interesting observation is that explainability actually may not be needed so much in positive cases, but much more so in negative cases: the parties that will demand explainability, will most likely be candidates who do not get hired.

A question here is whether rejected candidates indeed will be helped by explaining why an algorithm did not assess them well; as discussed in Langer et al. (2018), more transparency about algorithmic procedures and criteria may actually increase user skepticism. Furthermore, pointing the user at candidates who were successful in their place will also not be a pro-active type of feedback. It may be more beneficial to focus on constructive feedback towards future success; it will be a next grand challenge to research whether machine learning can play a role in this.

These observations align to the review on explanation in the social sciences by Miller (2017). As a main insight to include in future research, it is mentioned that explanations are often *contrastive*, *selected* and *social*, rather than only being a presentation of causes. However, within AI, also considering the job candidate screening problem, the main focus so far has been almost exclusively on the latter. By more explicitly including contrastive, selected and social elements, it is expected that explanations towards end users will improve in quality and acceptability.

6.5 Conclusion

As we discussed throughout this chapter, psychology and machine learning have complementary methodological interests, that may be combined in various novel ways. Careful and explicit treatment of validity, insight into the diversity of explainability opportunities, solid understanding of varying needs and interests of different stakeholders, and literacy across disciplines will be essential in making interdisciplinary collaborations work out in practice. If this can successfully be achieved, we foresee substantial innovation in the field, positively impacting researchers, practitioners and job candidates alike.

Acknowledgements This chapter results from collaborative work in the Erasmus+ Strategic Partnership ‘Big Data in Psychological Assessment’ (2017–2019), ref. 2017-1-DE01-KA203-003569, which is funded by the European Commission.

References

Achmadnoer Sukma Wicaksana (2017) Layered Regression Analysis on Multimodal Approach for Personality and Job Candidacy Prediction and Explanation. URL http://resolver.tudelft.nl/uuid:a527395d-f42c-426d-b80b-29c3b6478802

Achmadnoer Sukma Wicaksana, Liem CCS (2017) Human-Explainable Features for Job Candidate Screening Prediction. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, vol 2017-July, https://doi.org/10.1109/CVPRW.2017.212

Ambady N, Bernieri FJ, Richeson JA (2000) Toward a histology of social behavior: Judgmental accuracy from thin slices of the behavioral stream. Advances in Experimental Social Psychology 32:201–271, https://doi.org/10.1016/S0065-2601(00)80006-4, URL https://www.sciencedirect.com/science/article/pii/S0065260100800064

Anderson J (1983) Lix and Rix: Variations on a Little-known Readability Index. Journal of Reading 26(6):490–496, URL http://www.jstor.org/stable/40031755

Anderson N, Herriot P, Hodgkinson GP (2001) The practitioner-researcher divide in industrial, work and organizational (IWO) psychology: Where are we now, and where do we go from here? Journal of Occupational and Organizational Psychology https://doi.org/10.1348/096317901167451

Apers C, Derous E (2017) Are they accurate? Recruiters' personality judgments in paper versus video resumes. Computers in Human Behavior https://doi.org/10.1016/j.chb.2017.02.063

Baltrušaitis T, Mahmoud M, Robinson P (2015) Cross-dataset learning and person-specific normalisation for automatic Action Unit detection. In: FG, vol 06, pp 1–6, https://doi.org/10.1109/FG.2015.7284869, URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7284869

Bar-Yam Y (2016) The limits of phenomenology: From behaviorism to drug testing and engineering design. Complexity 21(S1):181–189, https://doi.org/10.1002/cplx.21730, URL http://doi.wiley.com/10.1002/cplx.21730

Barabas C, Dinakar K, Ito J, Virza M, Zittrain J (2018) Interventions over Predictions: Reframing the Ethical Debate for Actuarial Risk Assessment. In: Proceedings of the Conference on Fairness, Accountability and Transparency, Machine Learning Research, New York, vol 81, pp 1–15, URL http://proceedings.mlr.press/v81/barabas18a/barabas18a.pdf

Barrick MR, Mount MK (1991) The big fice personality dimensions and job performance: A meta-analysis. Personnel Psychology 44:1–26, https://doi.org/10.1111/j.1744-6570.1991.tb00688.x

Bauer TN, Truxillo DM, Sanchez RJ, Craig JM, Ferrara P, Campion MA (2001) Applicant reactions to selection: Development of the Selection Procedural Justice Scale. Personnel Psychology 54:387–419, https://doi.org/10.1111/j.1744-6570.2001.tb00097.x

Bauer TN, Truxillo DM, Tucker JS, Weathers V, Bertolino M, Erdogan B, Campion MA (2006) Selection in the information age: The impact of privacy concerns and computer experience on applicant reactions. Journal of Management 32:601–621, https://doi.org/10.1177/0149206306289829

Behrend TS, Sharek DJ, Meade AW, Wiebe EN (2011) The viability of crowdsourcing for survey research. Behavior Research Methods 43(3):800–813, https://doi.org/10.3758/s13428-011-0081-0, URL http://www.springerlink.com/index/10.3758/s13428-011-0081-0

Bengio Y, Courville A, Vincent P (2013) Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(8):1798–1828, https://doi.org/10.1109/TPAMI.2013.50

Biel JI, Aran O, Gatica-Perez D (2011) You Are Known by How You Vlog: Personality Impressions and Nonverbal Behavior in YouTube. Artificial Intelligence pp 446–449, URL http://www.idiap.ch/~jibiel/pubs/BielAranGaticaICWSM11.pdf

Biran O, Cotton C (2017) Explanation and justification in machine learning: A Survey. In: Proceedings of the 17th international joint conference on artificial intelligence IJCAI, Melbourne, Australia, pp 8–13

Bishop CM (2006) Pattern Recognition and Machine Learning. Springer-Verlag

Blacksmith N, Willford JC, Behrend TS (2016) Technology in the employment interview: A meta-analysis. Personnel Assessment and Decisions 2:2, https://doi.org/10.25035/pad.2016.002

Bobick AF, Davis JW (2001) The recognition of human movement using temporal templates. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(3):257–267, https://doi.org/10.1109/34.910878

Bolukbasi T, Chang KW, Zou J, Saligrama V, Kalai A (2016) Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. In: Proceedings of the 30th Conference on Neural Information Processing Systems, Barcelona, URL https://papers.nips.cc/paper/6228-man-is-to-computer-programmer-as-woman-is-to-homemaker-debiasing-word-embeddings.pdf

Borkenau P, Brecke S, Möttig C, Paelecke M (2009) Extraversion is accurately perceived after a 50-ms exposure to a face. Journal of Research in Personality 43(4):703–706, https://doi.org/10.1016/j.jrp.2009.03.007

Bradley R, Terry M (1952) Rank analysis of incomplete block designs: I. The method of paired comparisons. Biometrika 39(3/4):324–345, https://doi.org/10.2307/2334029, URL http://www.jstor.org/stable/10.2307/2334029

Buolamwini J, Gebru T (2018) Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification *. In: Proceedings of the Conference on Fairness, Accountability and Transparency, Machine Learning Research, vol 81, pp 1–15, URL http://proceedings.mlr.press/v81/buolamwini18a/buolamwini18a.pdf

Caliskan A, Bryson JJ, Narayanan A (2017) Semantics derived automatically from language corpora contain human-like biases. Science 356:183–186, https://doi.org/10.1126/science.aal4230

Campion MC, Campion MA, Campion ED, Reider MH (2016) Initial investigation into computer scoring of candidate essays for personnel selection. Journal of Applied Psychology 101:958–975, https://doi.org/10.1037/apl0000108

Caneel R (2005) Social Signaling in Decision Making. PhD thesis, Massachusetts Institute of Technology, URL http://groupmedia.media.mit.edu/datasets/Social_Signaling_in_Decision_Making.pdf

Chamorro-Premuzic T, Winsborough D, Sherman RA, Hogan R (2018) New Talent Signals: Shiny New Objects or a Brave New World? Industrial and Organizational Psychology 9(3):621–640, https://doi.org/10.1017/iop.2016.6

Chapman DS, Webster J (2003) The use of technologies in the recruiting, screening, and selection processes for job candidates. International journal of selection and assessment 11:113–120, https://doi.org/10.1111/1468-2389.00234

Chapman DS, Uggerslev KL, Webster J (2003) Applicant reactions to face-to-face and technology-mediated interviews: A field investigation. Journal of Applied Psychology 88:944–953, https://doi.org/10.1037/0021-9010.88.5.944

Chapman DS, Uggerslev KL, Carroll SA, Piasentin KA, Jones DA (2005) Applicant attraction to organizations and job choice: A meta-analytic review of the correlates of recruiting outcomes. Journal of Applied Psychology 90:928–944, https://doi.org/10.1037/0021-9010.90.5.928

Chen B, Escalera S, Guyon I, Ponce-Lopez V, Shah N, Simon MO (2016) Overcoming calibration problems in pattern labeling with pairwise ratings: Application to personality traits. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol 9915 LNCS, pp 419–432, https://doi.org/10.1007/978-3-319-49409-8_33

Choi BC, Pak AW (2006) Multidisciplinarity, interdisciplinarity and transdisciplinarity in health research, services, education and policy: 1. Definitions, objectives, and evidence of effectiveness. Clinical and Investigative Medicine https://doi.org/10.1016/j.jaac.2010.08.010

Christiansen ND, Wolcott-Burnam S, Janovics JE, Burns GN, Quirk SW (2005) The good judge revisited: Individual differences in the accuracy of personality judgments. Human Performance 18, https://doi.org/10.1207/s15327043hup1802_2

Coleman M, Liau TL (1975) A Computer Readability Formula Designed for Machine Scoring. Journal of Applied Psychology 60(2):283–284, https://doi.org/10.1037/h0076540, URL http://content.apa.org/journals/apl/60/2/283

Collobert R, Weston J (2008) A unified architecture for natural language processing: deep neural networks with multitask learning deep neural networks with multitask learning. International Conference on Machine Learning https://doi.org/10.1145/1390156.1390177

Cook M (2016) Personnel selection: adding value through people - a changing picture. Wiley-Blackwell

Council of the European Union (2016) General Data Protection Regulation. URL http://data.consilium.europa.eu/doc/document/ST-5419-2016-INIT/en/pdf

Crawford K, Calo R (2016) There is a blind spot in AI research. Nature 538(7625):311–313, https://doi.org/10.1038/538311a, URL http://www.nature.com/doifinder/10.1038/538311a

Cronbach LJ (1951) Coefficient alpha and the internal structure of tests. Psychometrika https://doi.org/10.1007/BF02310555

Davis MH, Scharenborg O (2017) Speech perception by humans and machines. In: Gaskell G, Mirkovi\'{c} J (eds) Speech Perception and Spoken Word Recognition., Routledge Psychology Press, chap 10, pp 181–204

De Kock FS, Lievens F, Born MP (2015) An In-Depth Look at Dispositional Reasoning and Interviewer Accuracy. Human Performance https://doi.org/10.1080/08959285.2015.1021046

De Kock FS, Lievens F, Born MP (2017) A closer look at the measurement of dispositional reasoning: Dimensionality and invariance across assessor groups. International Journal of Selection and Assessment https://doi.org/10.1111/ijsa.12176

Diekmann J, König CJ (2015) Personality testing in personnel selection: Love it? Leave it? Understand it! In: Nikolaou I, Oostrom J (eds) Employee recruitment, selection, and assessment: Contemporary issues for theory and practice, Psychology Press, Hove, UK, pp 117–135

Dwork C, Hardt M, Pitassi T, Reingold O, Zemel R (2012) Fairness through awareness. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference on - ITCS '12, ACM Press, New York, New York, USA, pp 214–226, https://doi.org/10.1145/2090236.2090255, URL http://dl.acm.org/citation.cfm?doid=2090236.2090255

Ekman P, Friesen WV (1978) Facial Action Coding System: A Technique for the Measurement of Facial Movement

Ekman P, Rosenberg E (2005) What the face reveals. Oxford University Press

Escalante HJ, Guyon I, Escalera S, Jacques J, Madadi M, Baro X, Ayache S, Viegas E, Gucluturk Y, Guclu U, Van Gerven MA, Van Lier R (2017) Design of an explainable machine learning challenge for video interviews. In: Proceedings of the International Joint Conference on Neural Networks, https://doi.org/10.1109/IJCNN.2017.7966320

Escalante HJ, Kaya H, Albert, Salah A, Escalera S, Gmur Güçlütürk Y, Güçlü U, Baró X, Guyon I, Junior JJ, Madadi M, Ayache S, Viegas E, Gürpinar F, Achmadnoer, Wicaksana S, Liem CCS, Van Gerven MAJ, Van Lier R, Salah AA (2018) Explaining First Impressions: Modeling, Recognizing, and Explaining Apparent Personality from Videos. ArXiv e-prints 1802.00745

Flesch R (1948) A New Readability Yardstick. The Journal of Applied Psychology 32(3):221–233, https://doi.org/10.1037/h0057532

Funder DC (1999) Personality judgment: a realistic approach to person perception. Academic Press

Funder DC (2012) Accurate Personality Judgment. Current Directions in Psychological Science https://doi.org/10.1177/0963721412445309

Furr RM, Bacharach VR (2014) Psychometrics: an introduction, second edition edn. SAGE Publications

Gilliland SW (1993) The perceived fairness of selection systems: An organizational justice perspective. Academy of Management Review 18:694–734, https://doi.org/10.2307/258595

Gilmore DC, Beehr TA, Love KG (1986) Effects of applicant sex, applicant physical attractiveness, type of rater and type of job on interview decisions*. Journal of Occupational Psychology 59:103–109

Goodfellow I, Bengio Y, Courville A (2016) Deep Learning. MIT Press

Goodfellow IJ, Shlens J, Szegedy C (2015) Explaining and Harnessing Adversarial Examples. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR2015), San Diego, URL https://arxiv.org/pdf/1412.6572.pdf

Gorbova J, Lusi I, Litvin A, Anbarjafari G (2017) Automated Screening of Job Candidate Based on Multimodal Video Processing. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, pp 1679–1685, https://doi.org/10.1109/CVPRW.2017.214, URL http://ieeexplore.ieee.org/document/8014948/

Graves A, Jaitly N (2014) Towards End-To-End Speech Recognition with Recurrent Neural Networks. In: Proceedings of the 31st International Conference on Machine Learning (ICML-14)

Guion RM (2011) Assessment, measurement, and prediction for personnel decisions. Routledge

Gunning R (1952) The Technique of Clear Writing. McGraw-Hill, URL https://books.google.nl/books?id=ofI0AAAAMAAJ

Hall GS (1917) Practical relations between psychology and the war. Journal of Applied Psychology https://doi.org/10.1037/h0070238

Hamel P, Eck D (2010) Learning Features from Music Audio with Deep Belief Networks. In: International Society for Music Information Retrieval Conference (ISMIR)

Henrich J, Heine SJ, Norenzayan A (2010) The weirdest people in the world? Behavioral and Brain Sciences 33(2–3):61–83, https://doi.org/10.1017/S0140525X0999152X

Hiemstra AM, Derous E, Serlie AW, Born MP (2012) Fairness Perceptions of Video Resumes among Ethnically Diverse Applicants. International Journal of Selection and Assessment https://doi.org/10.1111/ijsa.12005

Hofstadter D (2018) The Shallowness of Google Translate, The Atlantic. URL https://www.theatlantic.com/technology/archive/2018/01/the-shallowness-of-google-translate/551570/

Hough LM, Oswald FL, Ployhart RE (2001) Determinants, detection and amelioration of adverse impact in personnel selection procedures: Issues, evidence and lessons learned. International Journal of Selection and Assessment 9:152–194, https://doi.org/10.1111/1468-2389.00171

Humphries M, Gurney K, Prescott T (2006) The brainstem reticular formation is a small-world, not scale-free, network. Proceedings of the Royal Society B: Biological Sciences 273(1585):503–511, https://doi.org/10.1098/rspb.2005.3354, URL http://rspb.royalsocietypublishing.org/cgi/doi/10.1098/rspb.2005.3354

Kaya H, Gurpinar F, Salah AA (2017) Multi-modal Score Fusion and Decision Trees for Explainable Automatic Job Candidate Screening from Video CVs. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, pp 1651–1659, https://doi.org/10.1109/CVPRW.2017.210, URL http://ieeexplore.ieee.org/document/8014944/

Kim J, Urbano J, Liem CCS, Hanjalic A (2018) One Deep Music Representation to Rule Them All? A comparative analysis of different representation learning strategies. ArXiv e-prints 1802.00745

Kincaid JP, Fishburne RP, Rogers RL, Chissom BS (1975) Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel. Technical Training Research B(February):49, https://doi.org/ERIC:ED108134, URL http://www.dtic.mil/dtic/tr/fulltext/u2/a006655.pdf

Klehe UC (2004) Choosing how to choose: Institutional pressures affecting the adoption of personnel selection procedures. International Journal of Selection and Assessment 12:327–342, https://doi.org/10.1111/j.0965-075x.2004.00288.x

König CJ, Klehe UC, Berchtold M, Kleinmann M (2010) Reasons for being selective when choosing personnel selection procedures. International Journal of Selection and Assessment 18:17–27, https://doi.org/10.1111/j.1468-2389.2010.00485.x

König CJ, Steiner Thommen LA, Wittwer AM, Kleinmann M (2017) Are observer ratings of applicants' personality also faked? yes, but less than self-reports. International Journal of Selection and Assessment 25:183–192, https://doi.org/10.1111/ijsa.12171

Langer M, König CJ, Krause K (2017a) Examining digital interviews for personnel selection: Applicant reactions and interviewer ratings. International Journal of Selection and Assessment 25:371–382, https://doi.org/10.1111/ijsa.12191

Langer M, König CJ, Papathanasiou M (2017b) User reactions to novel technologies in selection and training contexts. In: Annual meeting of the Society for Industrial and Organizational Psychology (SIOP), Orlando, FL

Langer M, König CJ, Fitili A (2018) Information as a double-edged sword: The role of computer experience and information on applicant reactions towards novel technologies for personnel selection. Computers in Human Behavior 81:19–30, https://doi.org/10.1016/j.chb.2017.11.036

Liem CCS, Müller M, Eck D, Tzanetakis G, Hanjalic A (2011) The need for music information retrieval with user-centered and multimodal strategies. In: MM'11 - Proceedings of the 2011 ACM Multimedia Conference and Co-Located Workshops - MIRUM 2011 Workshop, MIRUM'11, pp 1–6, https://doi.org/10.1145/2072529.2072531

Liem CCS, Rauber A, Lidy T, Lewis R, Raphael C, Reiss JD, Crawford T, Hanjalic A (2012) Music Information Technology and Professional Stakeholder Audiences: Mind the Adoption Gap. In: Dagstuhl Follow-Ups, vol 3, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, https://doi.org/10.4230/DFU.VOL3.11041.227, URL http://drops.dagstuhl.de/opus/volltexte/2012/3475/

Long J, Shelhamer E, Darrell T (2015) Fully Convolutional Networks for Semantic Segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), https://doi.org/10.1109/CVPR.2015.7298965

Mazzocchi F (2016) Could Big Data be the end of theory in science? EMBO reports 16(10), https://doi.org/10.15252/embr.201541001, URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4766450/pdf/EMBR-16-1250.pdf

McCrae RR, Costa PT Jr (1999) The five-factor theory of personality. In: Handbook of Personality: Theory and Research, Guilford Press, https://doi.org/10.1007/978-1-4615-0763-5_11

McLaughlin G (1969) SMOG grading: A new readability formula. Journal of reading 12(8):639–646, https://doi.org/10.1039/b105878a, URL http://www.jstor.org/stable/40011226

Miller T (2017) Explanation in Artificial Intelligence: Insights from the Social Sciences. ArXiv e-prints 1706.07269

Morgeson FP, Campion MA, Dipboye RL, Hollenbeck JR, Murphy K, Schmitt N (2007) Reconsiderung the use of personality tests in personnel selection contexts. Personnel Psychology 60:683–729, https://doi.org/10.1111/j.1744-6570.2007.00089.x

Mori M, MacDorman K, Kageki N (2012) The uncanny valley. IEEE Robotics & Automation Magazine 19:98–100, https://doi.org/10.1109/MRA.2012.2192811

Naim I, Tanveer MI, Gildea D, Hoque ME (2015) Automated analysis and prediction of job interview performance: The role of what you say and how you say it. In: 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, Ljubljana, Slovenia, pp 1–14, https://doi.org/10.1109/fg.2015.7163127

Nass C, Brave S (2005) Wired for Speech: How Voice Activates and Advances the Human Computer Relationship. Computational Linguistics 32(3):451–452, https://doi.org/10.1162/coli.2006.32.3.451, URL https://www.lib.uwo.ca/cgi-bin/ezpauthn.cgi?url=http://search.proquest.com/docview/1037392793?accountid=15115%5Cnhttp://vr2pk9sx9w.search.serialssolutions.com/?ctx%7B_%7Dver=Z39.88-2004%7B&%7Dctx%7B_%7Denc=info:ofi/enc:UTF-8%7B&%7Drfr%7B_%7Did=info:sid/ProQ%7B%25%7D3Aeducation%7B&%7Drft%7B

Naumann LP, Vazire S, Rentfrow PJ, Gosling SD (2009) Personality judgments based on physical appearance. Personality and social psychology bulletin 35(12):1661–1671, https://doi.org/10.1177/0146167209346309

Nguyen LS, Frauendorfer D, Mast MS, Gatica-Perez D (2014) Hire me: Computational inference of hirability in employment interviews based on nonverbal behavior. IEEE Transactions on Multimedia https://doi.org/10.1109/TMM.2014.2307169

Oh IS, Wang G, Mount MK (2011) Validity of Observer Ratings of the Five-Factor Model of Personality Traits: A Meta-Analysis. Journal of Applied Psychology https://doi.org/10.1037/a0021832

Peck JA, Levashina J (2017) Impression management and interview and job performance ratings: A meta-analysis of research design with tactics in mind. Frontiers in Psychology https://doi.org/10.3389/fpsyg.2017.00201

Pentland A (2004) Social Dynamics: Signals and Behavior. Proceedings of the 3rd International Conference on Developmental Learning, Oct 2004 5:263–267, URL http://vismod.media.mit.edu//tech-reports/TR-579.pdf

Ployhart RE, Schmitt N, Tippins NT (2017) Solving the Supreme Problem: 100 Years of selection and recruitment at the Journal of Applied Psychology. Journal of Applied Psychology 102:291, https://doi.org/10.1037/apl0000081.supp

Ponce-López V, Chen B, Oliu M, Corneanu C, Clapés A, Guyon I, Baró X, Escalante HJ, Escalera S (2016) ChaLearn LAP 2016: First Round Challenge on First Impressions - Dataset and Results. In: European Conference on Computer Vision (ECCV 2016) Workshops, Springer, Amsterdam, pp 400–418, https://doi.org/10.1007/978-3-319-49409-8_32, URL https://link.springer.com/chapter/10.1007/978-3-319-49409-8_32

Pulakos ED, Schmitt N (1995) Experience-based and situational interview questions: Studies of validity. Personnel Psychology 48:289–308, https://doi.org/10.1111/j.1744-6570.1995.tb01758.x

Russell SJ, Norvig P (2010) Artificial Intelligence - A Modern Approach (3. internat. ed.). Pearson Education

Ryan AM, McFarland L, Shl HB, Page R (1999) An International Look At Selection Practices: Nation and Culture As Explanations for Variability in Practice. Personnel Psychology https://doi.org/10.1111/j.1744-6570.1999.tb00165.x

Schmidt FL, Hunter JE (1998) The validity and utility of selection methods in personnel psychology: Practical and theoretical implications of 85 years of research findings. Psychological bulletin (1998)

Sitser T (2014) Predicting sales performance: Strengthening the personality – job performance linkage. PhD thesis, Erasmus University Rotterdam, URL https://repub.eur.nl/pub/51139/

Smeulders A, Worring M, Santini S, Gupta A, Jain R (2000) Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence https://doi.org/10.1109/34.895972

Smith EA, Senter RJ (1967) Automated readability index. AMRL-TR Aerospace Medical Research Laboratories (6570th) pp 1–14

Smith M (1994) A theory of the validity of predictors in selection. Journal of Occupational and Organizational Psychology https://doi.org/10.1111/j.2044-8325.1994.tb00546.x

van Sonderen E, Sanderman R, Coyne JC (2013) Ineffectiveness of reverse wording of questionnaire items: let's learn from cows in the rain. PloS one 8(7):e68,967, https://doi.org/10.1371/journal.pone.0068967, URL http://www.ncbi.nlm.nih.gov/pubmed/23935915 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3729568

Sturm BL (2014) A simple method to determine if a music information retrieval system is a 'horse'. IEEE Transactions on Multimedia https://doi.org/10.1109/TMM.2014.2330697

Sutton RS, Barto AG (1998) Reinforcement learning: an introduction. MIT Press

Urbano J, Schedl M, Serra X (2013) Evaluation in music information retrieval. Journal of Intelligent Information Systems https://doi.org/10.1007/s10844-013-0249-4

Watts DJ, Strogatz SH (1998) Collective dynamics of'small-world'networks. Nature 393(6684):440–442, https://doi.org/10.1038/30918, URL http://202.121.182.16/Course/slides2012/NetSci-2012-7.pdf, 0803.0939v1

Waung M, Hymes RW, Beatty JE (2014) The Effects of Video and Paper Resumes on Assessments of Personality, Applied Social Skills, Mental Capability, and Resume Outcomes. Basic and Applied Social Psychology 36(3):238–251, https://doi.org/10.1080/01973533.2014.894477, URL http://www.tandfonline.com/doi/abs/10.1080/01973533.2014.894477

Youyou W, Kosinski M, Stillwell D (2015) Computer-based personality judgments are more accurate than those made by humans. Proceedings of the National Academy of Sciences of the United States of America 112(4):1036–40, https://doi.org/10.1073/pnas.1418680112, URL http://www.ncbi.nlm.nih.gov/pubmed/25583507 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4313801

Multimodal Personality Trait Analysis for Explainable Modeling of Job Interview Decisions

Heysem Kaya and Albert Ali Salah

Abstract Automatic analysis of job interview screening decisions is useful for establishing the nature of biases that may play a role in such decisions. In particular, assessment of apparent personality gives insights into the first impressions evoked by a candidate. Such analysis tools can be used for training purposes, if they can be configured to provide appropriate and clear feedback. In this chapter, we describe a multimodal system that analyzes a short video of a job candidate, producing apparent personality scores and a prediction about whether the candidate will be invited for a further job interview or not. This system provides a visual and textual explanation about its decision, and was ranked first in the ChaLearn 2017 Job Candidate Screening Competition. We discuss the application scenario and the considerations from a broad perspective.

Keywords Explainable machine learning · Job candidate screening · Multimodal affective computing · Personality trait analysis

1 Introduction

Affective and social computing applications aim to realize computer systems that are responsive to social signals of people they interact with. Under this research program, we find robots and virtual agents that engage their users in affect-sensitive interactions, educational monitoring tools, systems that track user behavior for

H. Kaya (✉)
Department of Computer Engineering, Namik Kemal University, Corlu, Tekirdag, Turkey
e-mail: hkaya@nku.edu.tr

A. A. Salah
Department of Computer Engineering, Bogazici University, Istanbul, Turkey

Future Value Creation Research Center, Nagoya University, Nagoya, Japan
e-mail: salah@boun.edu.tr

H. J. Escalante et al. (eds.), *Explainable and Interpretable Models in Computer Vision and Machine Learning*, The Springer Series on Challenges in Machine Learning, https://doi.org/10.1007/978-3-319-98131-4_10

improved prediction capabilities and better services. With the increase of real-time capabilities of such systems, new application areas are becoming feasible, and such technologies are becoming more widespread. It is not uncommon now to have a camera that automatically takes a picture when the people in the frame are smiling. Yet with more widespread use, and more integration of such smart algorithms, there arises the need to design accountable systems that explain their decisions to their users, particularly for cases where these decisions have a major impact on the lives and wellbeing of other people. In this chapter, we describe one such application scenario, and discuss related issues within the context of a solution we have developed for this specific case.

Job interviews are one of the primary assessment tools for evaluating job seekers, and for many corporations and institutions, an essential part of the job candidate selection process. These relatively short interactions with individuals have potentially life-changing impact for the job seekers. In 2017, the ChaLearn Job Candidate Screening (JCS) Competition[1] was organized at CVPR, to investigate the value of automatic recommendation systems based on multimedia CVs (Escalante et al. 2017).

A system that can analyze a short video of the candidate to predict whether the candidate will be invited to a job interview or not is valuable for multiple reasons. For the recruiter, it can help visualize the biases in candidate selection, and assist in the training of the recruitment staff. For the job seeker, it can be a valuable tool to show what impression the candidate is giving to the recruiter, and if properly designed, could even suggest improvements in attitude, speaking style, posture, gaze behavior, attire, and such.

At this point, we caution the reader. It may be tempting to use such a system to automatically screen candidates when the job application figures are overwhelming. If the system approximates the human recruiter's behavior sufficiently well, it may even have a result very similar to the human recruiter's selection. However, what the system is evaluating is the first impression caused by the candidate, and this is not a sound basis to judge actual job performance. For example, overweight people are shown to be more negatively rated in job interviews compared to people with average weight, and were seen as less desirable, "less competent, less productive, not industrious, disorganized, indecisive, inactive, and less successful" (Larkin and Pines 1979). These stereotypes that the human recruiters have will be learned by the automatic system that relies on human annotations for its supervision. Subsequently, the system will also exhibit such biases. Therefore, it is necessary both to investigate any systematic biases in the system, and to design mechanisms where the system gives an explanation about its particular decision, by looking at its own decision process. This resembles endowing the system with a meta-cognitive module. If the output of such a module can be fed back into the system for removing biases in its learning, we will be on our way for much smarter systems.

[1] It was officially called a co-opetition, as it promoted sharing code and results between participants.

In this chapter, we first report some related work on apparent personality estimation and evaluation of video resumes for job interviews. We describe the Job Candidate Screening Challenge briefly, and then describe an end-to-end system that officially participated in the Challenge. We report our experimental results, and then investigate both the biases inherent in the annotations, and in the ensuing system. We also describe the meta-cognitive part of the system, namely, the module that explains its decisions. We discuss our findings, the contributions of the challenge to our understanding of the problem, our shortcomings, and what the future looks like for this research area.

2 Related Work

From a physchological perspective, personality is observed as a long term summary of behaviors, having a complex structure that is shaped by many factors such as habits and values. Analysis of personality is difficult, and requires psychological testing on the subject for obtaining a ground truth. Researchers in the field also analyze the "apparent personality," i.e. the *impressions* a subject leaves on other people (the annotators), instead of the actual personality (Gürpınar et al. 2016b; Lopez et al. 2016; Junior et al. 2018). This is easier to annotate, as only external evaluations are required for annotations, and the actual subject is not involved. Both real and apparent personality are typically assessed along the "Big Five" personality traits, namely, Openness to Experience, Conscientiousness, Extraversion, Agreeableness, and Neuroticism (commonly abbreviated as OCEAN), respectively (Valente et al. 2012).

Modeling and predicting apparent personality is studied from different modalities, particulary speech acoustics (Schuller et al. 2012; Valente et al. 2012; Madzlan et al. 2014), linguistics (Alam et al. 2013; Gievska and Koroveshovski 2014; Nowson and Gill 2014) and visual input (Fernando et al. 2016; Qin et al. 2016). In the literature, short segments of audio or video are used for automatic predictions (Kaya and Salah 2014; Celiktutan and Gunes 2016). Furthermore, multimodal systems that benefit from complementary information are increasingly studied (Alam and Riccardi 2014; Farnadi et al. 2014; Sarkar et al. 2014; Sidorov et al. 2014; Gürpınar et al. 2016a; Barezi et al. 2018).

Deep learning based classifiers have been shown to work well for predicting apparent personality ratings from visual input (Lopez et al. 2016; Zhang et al. 2016; Güçlütürk et al. 2016, 2017; Kaya et al. 2017a; Escalante et al. 2018; Barezi et al. 2018). However, the need for large amounts of training data and high memory/computational complexity of training deep network models are some of the disadvantages for deep learning based methods.

Deep learning models for personality analysis typically look at the face or the facial behavior of the person to determine what stereotypes it will activate in the viewers. An advantage of deep neural networks for analysing facial images is that the earlier layers of the network learn good internal representations for faces, regardless of the facial analysis task targeted by the supervised learning process. Since it is relatively easy to collect large amounts of face images together with identity labels (e.g. famous persons) from the Internet, it is possible to train a deep neural network for a face recognition task with millions of samples. Once this is done, the resulting deep (convolutional) neural network can serve as a pre-trained model to enable efficient and effective transfer learning on other tasks, such as emotional expression recognition (Kaya et al. 2017b).

There are different approaches to transfer learning (Pan and Yang 2010). The approach we use in this work is one where we start from a model pre-trained with a very large database, and fine-tune the model for a different task using a smaller database. This approach is ideal if there are not sufficiently many samples for training a deep model in the target task, but when the task shares structural similarities (i.e. analysis of faces in our case) with a task that does have such large data for training (e.g. face recognition).

3 Job Candidate Screening Challenge

The CVPR 2017 Job Candidate Screening Challenge was organized to help both recruiters and job candidates using multi-media CVs (Escalante et al. 2017). The challenge relied on a publicly available dataset[2] that contains more than 10,000 clips (average duration 15 s) from more than 3000 videos collected from YouTube (Escalante et al. 2016). These are annotated via Amazon Mechanical Turk annotators for apparent personality traits, as well as a variable that measured whether the candidate would be invited to a job interview, or not. Basic statistics of the dataset partitions are provided in Table 1. The detailed information on the Challenge and the corpus can be found in Lopez et al. (2016).

The apparent personality annotations were made through a single question asked per dimension. The annotators saw a pair of candidates, and assigned an attribute to one of the videos (with an option of not assigning it to any video). The attributes used to measure the “Big Five” personality traits were as follows: Friendly vs. Reserved (for Extraversion), Authentic vs. Self-interested (for Agreeableness), Organized vs. Sloppy (for Conscientiousness), Uneasy vs. Comfortable (for Neuroticism), Imaginative vs. Practical (for Openness to Experience). Previously, the

Table 1 Dataset summary

	Train	Val	Test
#Clips	6000	2000	2000
#YouTube videos	2624	1484	1455
#Given frames	2.56M	0.86M	0.86M
#Detected frames	2.45M	0.82M	0.82M

[2] The dataset can be obtained from http://chalearnlap.cvc.uab.es/dataset/24/description/.

ChaLearn Looking at People 2016 First Impression Challenge was organized to develop systems that can predict these apparent personality ratings (Lopez et al. 2016). Additionally, the question of “Who would you rather invite for a job interview?” was posed to obtain a ground truth for the job candidate screening task. These annotations were post-processed to produce cardinal scores for each clip (Escalante et al. 2018).

The Challenge itself was composed of two stages: a quantitative challenge to predict the “invite for interview” variable, and a qualitative challenge to justify the decision with verbal/visual explanations, respectively. The participants were encouraged to use the personality trait dimensions in prediction (quantitative) and explanation (qualitative) stages.

4 Proposed Method

The prediction problem we focus in this paper is based on assessing a short input video for the “Big Five” personality traits and the “invite for interview” variable. The available modalities for analysis include the facial image of the candidate, the acoustics of his or her voice, and the features that can be extracted from the background, which we call the scene. Inspired from the winning system of ICPR 2016 ChaLearn Apparent Personality Challenge that was organized with the same corpus and protocol (Gürpınar et al. 2016b), we implement a multimodal system that evaluates audio, scene, and facial features as separate channels, and use Extreme Learning Machine classifiers to produce intermediate results for each channel. These first-level predictions are then combined in a second modeling stage to produce the final predictions.

The second stage of the competition required the submitted systems to produce explanations for the decisions of the system. It is possible to investigate the system dynamics, the learned features, the weights of the individual classifiers in the system, etc., and follow the path of a decision from the input to the output. This would generate a lot of information, and might make interpretation difficult. We choose a simple approach, where the first-level predictions are treated as a black-box, and no insights are generated for these predictions. However, the final prediction, which is based on the intermediate apparent personality trait estimations of the system, is generated with a tree-based classifier to enable the generation of an explanation. We describe all the components of this system in this section.

The pipeline of the proposed system for the quantitative challenge is illustrated in Fig. 1. The input is represented on the left hand side, which consists of a video and its associated audio track. The face is detected, and two sets of features are extracted from the facial image. These are combined via feature-level fusion in the first kernel ELM classifier in the Modeling part. The scene features and the audio features are combined in another, similar classifier. On the right hand side, there is a stacked random forest classifier to give the final predictions, and it is this classifier that the system uses to generate an explanation about its behavior.

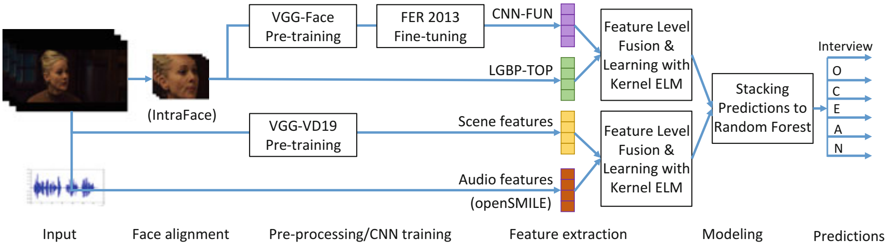

VGG-Face Pre-training
FER 2013 Fine-tuning
CNN-FUN
LGBP-TOP
(IntraFace)
VGG-VD19 Pre-training
Scene features
Audio features (openSMILE)
Feature Level Fusion & Learning with Kernel ELM
Feature Level Fusion & Learning with Kernel ELM
Stacking Predictions to Random Forest
Interview
O
C
E
A
N
Input
Face alignment
Pre-processing/CNN training
Feature extraction
Modeling
Predictions

Fig. 1 Flowchart of the proposed method (Kaya et al. 2017a)

We now briefly describe the main steps of our pipeline, namely, face alignment, feature extraction, and modeling, respectively. We refer the reader to Gürpınar et al. (2016b), Kaya et al. (2017a) for more technical details.

4.1 Visual Feature Extraction

The system detects faces, and locates 49 facial landmarks on each faces using the Supervised Descent Method (SDM) (Xiong and De la Torre 2013). These points are extremely important to align facial images, so that a comparative analysis can be performed. The roll angle of the face is estimated from the eye corners to normalize the facial image. The distance between the two eyes is called the *interocular distance*, and it is frequently used to normalize the scale of the facial image. Our system adds a margin of 20% of the interocular distance around the outer landmarks to crop the facial image. Each such cropped image is resized to 64×64 pixels. These images are processed in two ways.

The first way uses a deep neural network. We start with the pre-trained VGG-Face network (Parkhi et al. 2015), which is optimized for the face recognition task on a very large set of faces. We change the final layer (originally a 2622-dimensional recognition layer), to a 7-dimensional emotion recognition layer, where the weights are initialized randomly. We then fine-tune this network with the softmax loss function using more than 30K training images of the FER-2013 dataset (Goodfellow et al. 2013). We choose an initial learning rate of 0.0001, a momentum of 0.9 and a batch size of 64. We train the model only for 5 epochs. The final, trained network has a 37-layer architecture (involving 16 convolution layers and five pooling layers). The response of the 33rd layer is used in this work, which is the lowest-level 4096-dimensional descriptor.

We combine deep facial features with a second set of features. We use a spatio-temporal descriptor called Local Gabor Binary Patterns from Three Orthogonal Planes (LGBP-TOP) (Almaev and Valstar 2013) that is shown to be effective in emotion recognition (Kaya et al. 2017b). The LGBP-TOP descriptor is extracted by applying 18 Gabor filters on aligned facial images with varying orientation and scale parameters. The resulting feature dimensionality is 50,112.

Facial features are extracted over an entire video segment and summarized by functionals. The functionals include mean, standard deviation, offset, slope, and curvature. Offset and slope are calculated from the first order polynomial fit to each feature contour, while curvature is the leading coefficient of the second order polynomial. Scene features, however, are extracted from the first image of each video only. The assumption is that videos do not stretch over multiple shots.

In order to use the ambient information in the images to our advantage, we extract a set of features using the VGG-VD-19 network (Simonyan and Zisserman 2014), which is trained for an object recognition task on the ILSVRC 2012 dataset. Similar to face features, we use the 4096-dimensional feature from the 39th layer of the 43-layer architecture, hence we obtain a description of the overall image that contains both face and scene. Using a deep neural network originally trained for an object recognition task basically serves to detect high level objects and object-like parts in these images, which may be linked to the decision variables. It is theoretically possible to analyze this part of the system in greater detail, to detect which objects in the scene, if any, are linked to particular trait predictions. However, the number of training samples is small compared to the number of object classes the network is originally trained for, and consequently, such an analysis may be misleading.

It would be really interesting to conduct a more extensive study to see which objects are associated with which personality traits strongly. Obviously, cultural factors should also be considered for this purpose. In our previous work, we have illustrated the effectiveness of scene features for predicting Big Five traits to some extent (Gürpınar et al. 2016a,b). For the Job Candidate Screening task, these features contribute to the final decision both directly (i.e. in a classifier that predicts the interview variable) and indirectly (i.e. over the personality trait predictions that are used in the final classifier for the interview variable).

4.2 Acoustic Features

There are excellent signal processing approaches for using the acoustic features. The open-source openSMILE tool (Eyben et al. 2010) is popularly used to extract acoustic features in a number of international paralinguistic and multi-modal challenges. The idea is to obtain a large pool of potentially relevant features by passing an extensive set of summarizing functionals on the low level descriptor (LLD) contours (e.g. Mel Frequency Cepstral Coefficients—MFCC, pitch, energy and their first/second order temporal derivatives).

We use the toolbox with a standard feature configuration that served as the challenge baseline sets in INTERSPEECH 2013 Computational Paralinguistics Challenge (Schuller et al. 2013). This set includes energy, spectral, cepstral (MFCC) and voicing related low-level descriptors (LLDs). Additionally, there are LLDs that complement these features, such as logarithmic harmonic-to-noise ratio (HNR), spectral harmonicity, and psychoacoustic spectral sharpness. In our former work, we compared INTERSPEECH 2013 configuration with the other baseline feature sets used in the computational paralinguistics challenges, and found it to be the most effective for the personality trait recognition task (Gürpınar et al. 2016b). Thus, based on the former analyses, here we use the configuration from (Schuller et al. 2013).

4.3 Classification

We use several levels of classifiers to obtain the model predictions. In all levels, we use simple classifiers with few meta-parameters to prevent overfitting. Overfitting typically happens if the number of free parameters in the classifier and the dimensionality of the samples are large with respect to the number of training samples. Since our models base their decisions on many features obtained from different channels, overfitting is a very important issue.

We use kernel extreme learning machines (ELM) in our first tier classification. The ELM classifier is basically a single-layer neural network, but the first layer weights are determined from the training data (in the kernel version), and the second layer weights are analytically computed. Subsequently, it is very fast to train. We have observed in our simulations that its accuracy is good, and the system is robust. We do not detail the classifier here, and refer the reader to Huang et al. (2004) for technical details. We use a linear kernel, which has only a single parameter (the regularization coefficient), which we optimize with a sixfold subject independent cross-validation on the training set.

Once the model has generated a number of predictions from multiple modalities via ELM classifiers, these are stacked to a Random Forest (RF) classifier in the second stage of classification. This is the fusion stage, where the classifier learns to give appropriate weights to different modalities, or features. The RF classifier is an ensemble of decision tree (DT) classifiers. Tree based classifiers base their decisions on multiple tests, where each internal node of the tree tests one attribute, or a feature of the input sample, deciding which branch will be taken next. The root note contains the first test, and the leaf nodes of the tree will contain the decision, i.e. the assigned class of the sample. It is possible to trace the decisions from root to branch, and see which attributes have led to the particular decision. Consequently, decision trees are easy to interpret. The random forest introduces robustness to decision trees by randomly sampling subsets of instances with replacement, and by training multiple trees based on these samples (Breiman 2001).

To increase the interpretability of the final decision on the interview variable, we use the training set mean value of each attribute to binarize each score prediction of the RF as HIGH or LOW. Thus, if the model predicts the agreeableness of a person as higher than the average agreeableness of the training samples, it is labeled as HIGH AGREEABLENESS. The final classifier that decides on the interview variable is a decision tree, which takes the binarized apparent personality scores, predicted by the RF, and outputs the binary interview class (i.e. invited, or not invited).

Once the decision is given, the system converts it into an explicit description using "if-then" rules and a template, by tracing the decision from the root of the tree to the leaf. The template is formed as follows (Kaya et al. 2017a):

- If the invite decision is 'YES' → 'This [gentleman/lady] is invited due to [his/her] high apparent {list of high scores on the trace}' [optional depending on path:', although low {list of low scores on the trace} is observed.']
- If the invite decision is 'NO' → 'This [gentleman/lady] is not invited due to [his/her] low apparent {list of low scores on the trace}' [optional depending on path: ' , although high {list of high scores on the trace} is observed.']

In the preliminary weighted fusion experiments we have conducted, we have observed that the video modality typically has higher weight in the final prediction. Similarly, in the audio-scene model, the audio features are more dominant. We reflect this prior knowledge in the automatically generated explanations by checking whether the high/low scores of each dimension have the same sign with that of the model trained on facial features. After this check, the system includes some extra information for the leading apparent personality dimension that helped admittance (or caused rejection). The template for this information is:

'The impressions of {list of traits where visual modality has the same sign with the final decision} are primarily gained from facial features.' [optional, depending on existence: 'Furthermore, the impression of {the list of audio-dominant traits} is predominantly modulated by voice.']

Finally, each record is accompanied with the aligned face from the first face-detected frame of the video and with a bar graph of the mean-normalized predicted scores. This helps the decision maker visualize more precisely what the system computed to base its decision. We give several output examples in the next section.

5 Experimental Results

The "ChaLearn LAP Apparent Personality Analysis: First Impressions" challenge consists of 10,000 clips collected from 5563 YouTube videos, where the poses are more or less frontal, but the resolution, lighting and background conditions are not controlled, hence providing a dataset with in-the-wild conditions. Each clip in the training set is labeled for the Big Five personality traits and an "interview invitation" annotation using Amazon Mechanical Turk. The former is an apparent personality trait, and does not necessarily reflect the actual personality of the person. Similarly, the latter is a decision on whether the person in the video is invited to the interview or not, and signifies a positive or negative general impression.

For brevity, we skip corpus related information here, and refer the reader to Lopez et al. (2016) for details on the challenge. The performance score in this challenge is the Mean Absolute Error subtracted from 1, which is formulated as follows:

$$1 - \sum_{i}^{N} \frac{|\hat{y}_i - y_i|}{N}, \tag{1}$$

where N is the number of samples, $\hat{y}$ is the predicted label and y is the true label ($0 \leq y \leq 1$). This means the final score varies between 0 (worst case) and 1 (best case).

The competition has a clear experimental protocol, which is followed in this work. The test set labels are sequestered, and limited number of test score submissions were allowed to prevent overfitting. We describe two sets of experiments, by taking a regression and a classification approach, respectively.

5.1 Experimental Results Using Regression

The natural way to predict continuous apparent personality traits is via regression. We train our regressors with 6000 training set instances, using a sixfold cross-validation (CV) to optimize model hyper-parameters for each feature type and their combinations. Training and validation sets were combined for training the final system for test set predictions.

In Table 2, we report the validation set performances of individual features, as well as their feature-, score- and multi-level fusion alternatives. Here, System 0 corresponds to the top entry in the ICPR 2016 Challenge (Gürpınar et al. 2016b), which uses the same set of features and fuses scores with linear weights. For the weighted score fusion, the weights are searched in the [0,1] range with steps of 0.05. Face, scene, and audio features are used individually, and reported in lines 1–4. These indicate the accuracy of single-modality subsystems. Lines 5–8 are the multimodal fusion approaches.

In general, fusion scores are observed to benefit from complementary information of individual sub-systems. Moreover, we see that fusion of two different types of face features improves over their individual performance. Similarly, the feature level fusion of audio and scene sub-systems is observed to benefit from complementarity.

Table 2 Validation set performance of the proposed framework (System 8) and its sub-systems

SysID	System	INTER	AGRE	CONS	EXTR	NEUR	OPEN	TRAIT AVG
0	ICPR 2016 Winner	N/A	0.9143	0.9141	0.9186	0.9123	0.9141	0.9147
1	Face: VGGFER33	0.9095	0.9119	0.9046	0.9135	0.9056	0.9090	0.9089
2	Face: LGBPTOP	0.9112	0.9119	0.9085	0.9130	0.9085	0.9103	0.9104
3	Scene: VD_19	0.8895	0.8954	0.8924	0.8863	0.8843	0.8942	0.8905
4	Audio: OS_IS13	0.8999	0.9065	0.8919	0.8980	0.8991	0.9022	0.8995
5	FF(Sys1, Sys2)	0.9156	0.9144	0.9125	0.9185	0.9124	0.9134	0.9143
6	FF(Sys3, Sys4)	0.9061	0.9091	0.9027	0.9013	0.9033	0.9068	0.9047
7	WF(Sys5, Sys6)	0.9172	**0.9161**	0.9138	0.9192	0.9141	0.9155	0.9157
8	RF(Sys5, Sys6)	**0.9198**	**0.9161**	**0.9166**	**0.9206**	**0.9149**	**0.9169**	**0.9170**

FF: Feature-level fusion, WF: Weighted score-level fusion, RF: Random Forest based score-level fusion. INTER: Interview invite variable. AGRE: Agreeableness. CONS: Conscientiousness. EXTR: Extraversion. NEUR: Neuroticism. OPEN: Openness to experience

Table 3 Test set performance of the top systems in the CVPR'17 coopetition—quantitative stage

Participant	INTER	AGRE	CONS	EXTR	NEUR	OPEN	TRAIT AVG
Ours	**0.9209**	**0.9137**	**0.9198**	**0.9213**	**0.9146**	**0.9170**	**0.9173**
Baseline	0.9162	0.9112	0.9152	0.9112	0.9104	0.9111	0.9118
First Runner Up	0.9157	0.9103	0.9138	0.9155	0.9083	0.9101	0.9116
Second Runner Up	0.9019	0.9032	0.8949	0.9027	0.9011	0.9047	0.9013

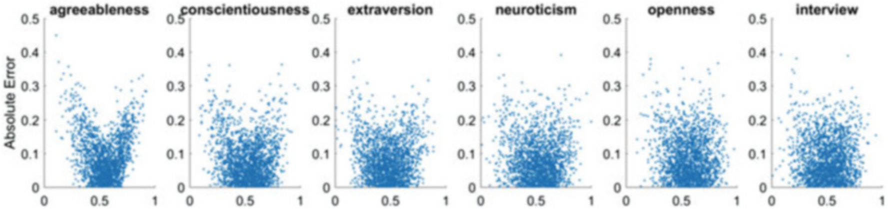


Fig. 2 Absolute error of the test set predictions (y-axis) as a function of ground truth (x-axis)

The final score fusion with RF outperforms weighted fusion in all but one dimension (agreeableness), where the performances are equal.

Based on the validation set results, the best fusion system (System 8 in Table 2) is obtained by stacking the predictions from Face feature-fusion (FF) model (System 5) with the Audio-Scene FF model (System 6). This fusion system renders a test set performance of 0.9209 for the interview variable, ranking the first and beating the challenge baseline score (see Table 3). Furthermore, the average of the apparent personality trait scores is 0.917, which advances the state-of-the art result (0.913) obtained by the winner of ICPR 2016 ChaLearn LAP First Impression contest (Gürpınar et al. 2016b).

The test set results of the top ranking teams are both high and competitive. When individual personality dimensions are analyzed, we see that our system ranks the first in all dimensions, exhibiting the highest improvement over the baseline in prediction of Extraversion and the Interview variable. We also observe that the proposed system's validation and test accuracies are very similar: the mean absolute difference of the six dimensions is 0.13%. Therefore, we can conclude that the generalization ability of the proposed system is high.

After the official challenge ended, we have obtained the test set labels from the organizers and analyzed the distribution of the absolute error in our system with respect to the ground truth. Figure 2 shows the scatter plots of the six target variables. The x-axis denotes the ground truth scores, and the V shape we observe in these plots, with a mass centered around the point (0.5, 0.05), means that our least squares based regressor is conservative, trying to avoid extreme decisions. The largest errors are made for high and low value assignments, particularly for Agreeableness. A cumulative distribution analysis shows that over all dimensions, 37.5% of the test set predictions have MAEs less than 0.05, and 67.3% of predictions have MAEs less than 0.1, with only 5.2% of the predictions having a MAE higher than 0.2.

Table 4 Test set classification accuracies for the top single and multimodal systems

Sys.	Modality	Features/fusion	Interview	Trait Avg.
1	Audio + video	EF(FaceSys,AudioSceneSys)	**77.10**	**75.63**
2	Video (face seq.)	FF(VGGFER33,LGBPTOP)	76.35	74.45
3	Audio + scene + first face	FF(IS13,VGGFER33,VGGVD19,LBP)	74.00	72.31
4	Audio + scene	FF(IS13,VGGVD19)	71.95	70.47
5	First face + scene	FF(VGGFER33,VGGVD19)	71.15	69.97
6	Audio	IS13 functionals	69.25	67.93

The scene feature is extracted from the first video frame. FF: Feature Fusion, EF: Equal weighted score fusion

5.2 *Experimental Results Using Classification*

For improved interpretability, the prediction problem can be handled as a binary classification into LOW and HIGH values, which we investigate in this section. Additionally, we analyze how well a parsimonious system can do by looking at a single frame of the video, instead of face analysis in all frames.

To adapt the problem for classification, the continuous target variables in the [0,1] range are binarized using the training set mean statistic for each target dimension, separately. For the single-image tests, we extracted deep facial features from our fine-tuned VGG-FER DCNN, and accompanied them with easy-to-extract image descriptors, such as Local Binary Patterns (LBP) (Ojala et al. 2002), Histogram of Oriented Gradients (HOG) (Dalal and Triggs 2005) and Scale Invariant Feature Transform (Lowe 2004).

Hyper-parameter optimization and testing follow similar schemes as the previous section. The test set classification performances of the top systems for single- and multi-modal approaches are shown in Table 4. As expected, we see that the audio-visual approach also performs best in the classification task (77.10% accuracy on the interview variable). This is followed by the video-only approach using facial features (76.35%), and the fusion of audio with face and scene features from the first image (74%). Although this is relatively 4.6% lower compared to the best audio-visual approach, it is highly motivating, as it uses only a single image frame to predict the personality impressions and interview invitation decision, which the annotators gave by watching the whole video. It shows that without resorting to costly image processing and DCNN feature extraction for all images in a video, it is possible to achieve high accuracy, comparable to the state-of-the-art.

The dimension that is the hardest to classify is agreeableness, whereas accuracy for conscientiousness was consistently the highest (see Fig. 3). Among the conventional image descriptors, HOG was the most successful, with an average validation set recognition accuracy (over traits) of 70%, using only a single facial image. On the other hand, the fusion of scene and face features from the first video frame outperform acoustic features on both the development and test sets by 3%.

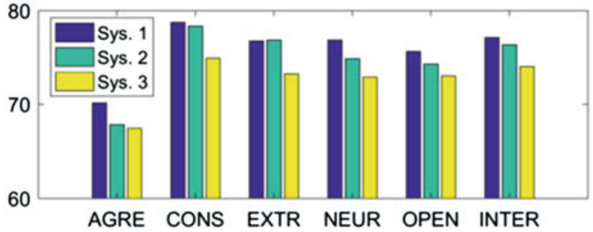


Fig. 3 Test set classification performance of top three fusion systems over personality traits and the interview variable. Sys. 1: Audio-Video system, Sys. 2: Video only system, Sys. 3: Audio plus a single image based system. NEUR refers to non-Neuroticism as it is used throughout the paper

6 Explainability Analysis

We now turn to the explainability analysis, which was tackled in the qualitative part of the ChaLearn competition. We use the final score outputs of the quantitative stage, as well as the classifiers themselves to produce readable explanations of the decisions.

To make the scores more accessible, we binarize them (as LOW-HIGH) by thresholding each dimension at corresponding training set mean, and feed them to a decision tree classifier, as explained in Sect. 4. In the preliminary experiments, we tried grouping the scores into more than two levels, using the mean and variance statistics. However, the final classification accuracy suffered, and this was abandoned.

The decision tree trained on the predicted Big Five personality dimensions gives a classification accuracy of 94.2% for the binarized interview variable. A visual illustration of the decision tree (DT) is given in Fig. 4.

The learned model is intuitive in that the higher scores of traits generally increase the chance of interview invitation. As can be seen from the figure, the DT ranks the relevance of the predicted Big Five traits from the highest (Agreeableness) to the lowest (Openness to Experience) with respect to information gain between corresponding trait and the interview variable. The second most important trait for job interview invitation is Neuroticism, which is followed by Conscientiousness and Extraversion. Neuroticism is the only trait which correlates negatively with the Interview variable, so it was represented with its opposite (i.e. non-Neuroticism) during annotations, to ensure sign consistency. Throughout this paper, we use non-Neuroticism as a feature. If the Openness score is high, then having a high score in any of the non-Neuroticism, Conscientiousness or Extraversion variables suffices for invitation. Chances of invitation decrease if Agreeableness is low: only three out of eight leaf nodes are "YES" in this branch. In two of these cases, one has to have high scores in three out of four remaining traits.

There is an interesting rule related to Openness. In some cases high Openness leads to "invite", whereas in others it leads to "do not invite". If Agreeableness is low, but non-Neuroticism and Extraversion are high, then the Openness should

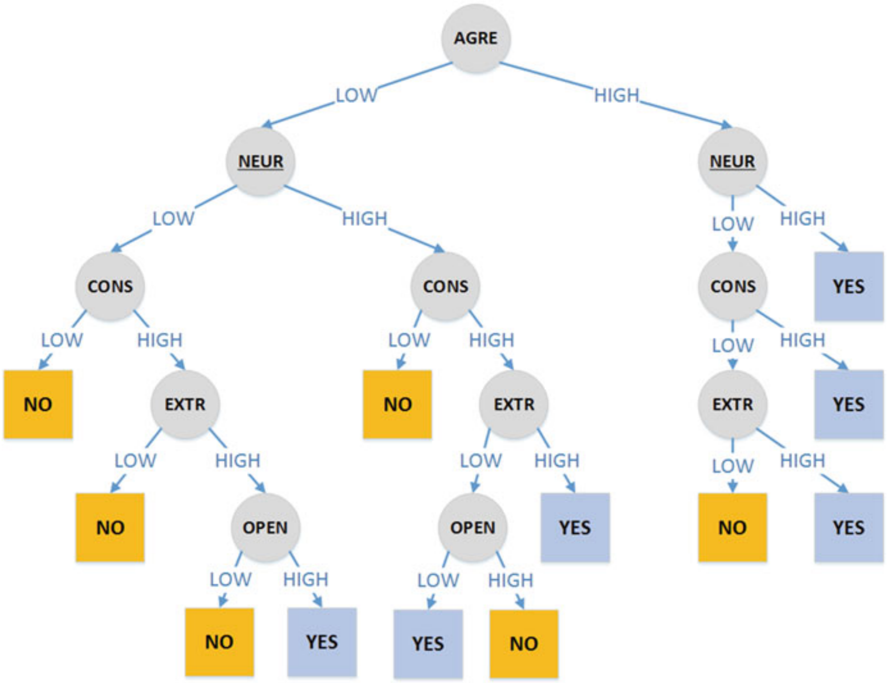
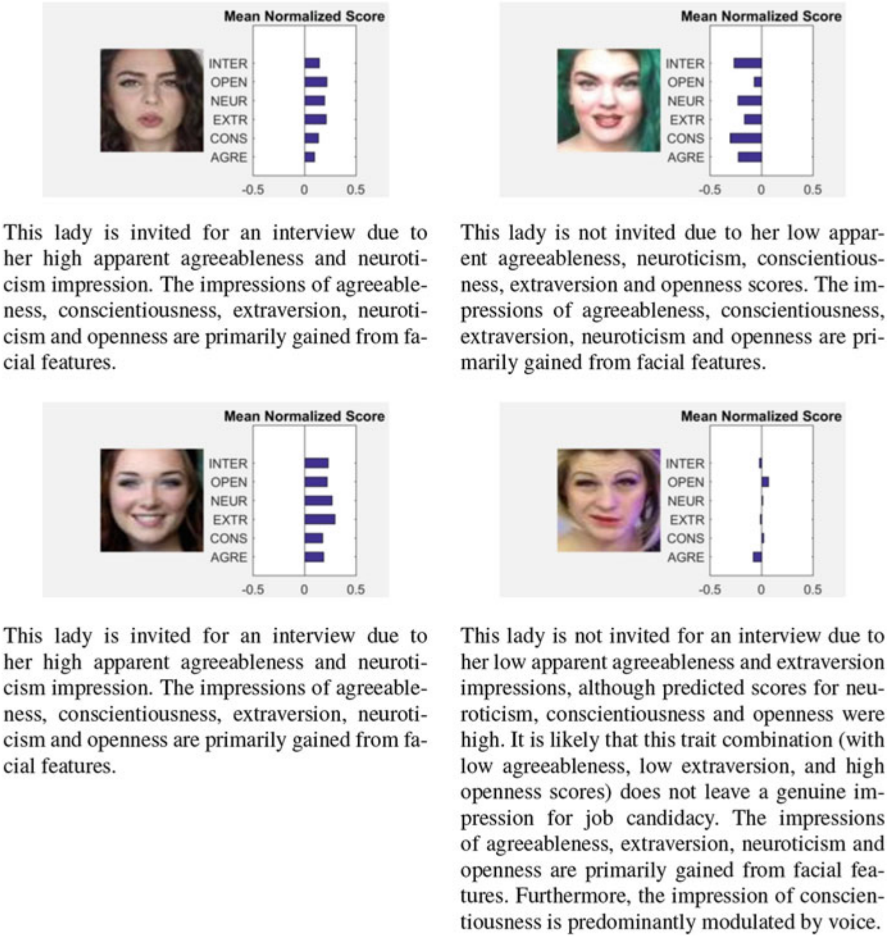


Fig. 4 Illustration of the trained decision tree for job interview invitation. *NEUR* represents non-Neuroticism, as explained in the text

be low for interview invitation (a high Openness score results in rejection). This may be due to an unwanted trait combination: someone with a low Agreeableness, Extraversion, and Neuroticism, but high Openness may be perceived as insincere and arrogant.

For verbal explanations, we converted the DT structure into a compact set of "if-then" rules in the form mentioned earlier. The metadata provided by the organizers do not contain sex annotations, which could have been useful in explanatory sentences. For this purpose, we have initially annotated 4000 development set (training + validation) videos using the first face-detected frames, then trained a sex prediction model based on the audio and video features used in the apparent personality trait recognition. The ELM based sex predictors gave 97.6% and 98.9% validation set accuracies using audio (openSMILE) and video (CNN-FUN) features, respectively. We fused the scores of audio and video models with equal weight and obtained a validation set accuracy of 99.3%, which is close to perfect. We then used all annotated data for training with the optimized hyper-parameters and cast predictions on the remaining 6000 (validation + test set) instances. After the challenge, we annotated the whole set of 10,000 videos for apparent age, sex, and ethnicity.

The verbal explanations are finally accompanied with the aligned image from the first face-detected frame and the bar graphs of corresponding mean normalized



This lady is invited for an interview due to her high apparent agreeableness and neuroticism impression. The impressions of agreeableness, conscientiousness, extraversion, neuroticism and openness are primarily gained from facial features.

This lady is not invited due to her low apparent agreeableness, neuroticism, conscientiousness, extraversion and openness scores. The impressions of agreeableness, conscientiousness, extraversion, neuroticism and openness are primarily gained from facial features.

This lady is invited for an interview due to her high apparent agreeableness and neuroticism impression. The impressions of agreeableness, conscientiousness, extraversion, neuroticism and openness are primarily gained from facial features.

This lady is not invited for an interview due to her low apparent agreeableness and extraversion impressions, although predicted scores for neuroticism, conscientiousness and openness were high. It is likely that this trait combination (with low agreeableness, low extraversion, and high openness scores) does not leave a genuine impression for job candidacy. The impressions of agreeableness, extraversion, neuroticism and openness are primarily gained from facial features. Furthermore, the impression of conscientiousness is predominantly modulated by voice.

Fig. 5 Sample verbal and visual explanations automatically generated by the system

scores. When we analyze the results, we observe that individually processed clips cut from different places of a single input video have very similar scores, and the same reasons for the invitation decision, showing the consistency of the proposed approach. Figure 5 illustrates some automatically generated verbal and visual explanations for this stage.

The test set of the quantitative challenge was based on the accuracy (1-MAE) of the interview variable. In the qualitative stage, the submissions (one for each team) were evaluated by a committee based on the following criteria:

- **Clarity**: Is the text understandable/written in proper English?
- **Explainability**: Does the text provide relevant explanations to the hiring decision made?

Table 5 Qualitative stage test stage winner teams' scores

Participant	Our team	First runner up
Clarity	4.31±0.54	3.33±1.43
Explainability	3.58±0.64	3.23±0.87
Soundness	3.40±0.66	3.43±0.92
Interpretability	3.83±0.69	2.40±1.02
Creativity	2.67±0.75	3.40±0.8
Mean score	3.56	3.16

- **Soundness**: Are the explanations rational and, in particular, do they seem scientific and/or related to behavioral cues commonly used in psychology.
- **Model interpretability**: Are the explanation useful to understand the functioning of the predictive model?
- **Creativity:** How original/creative are the explanations?

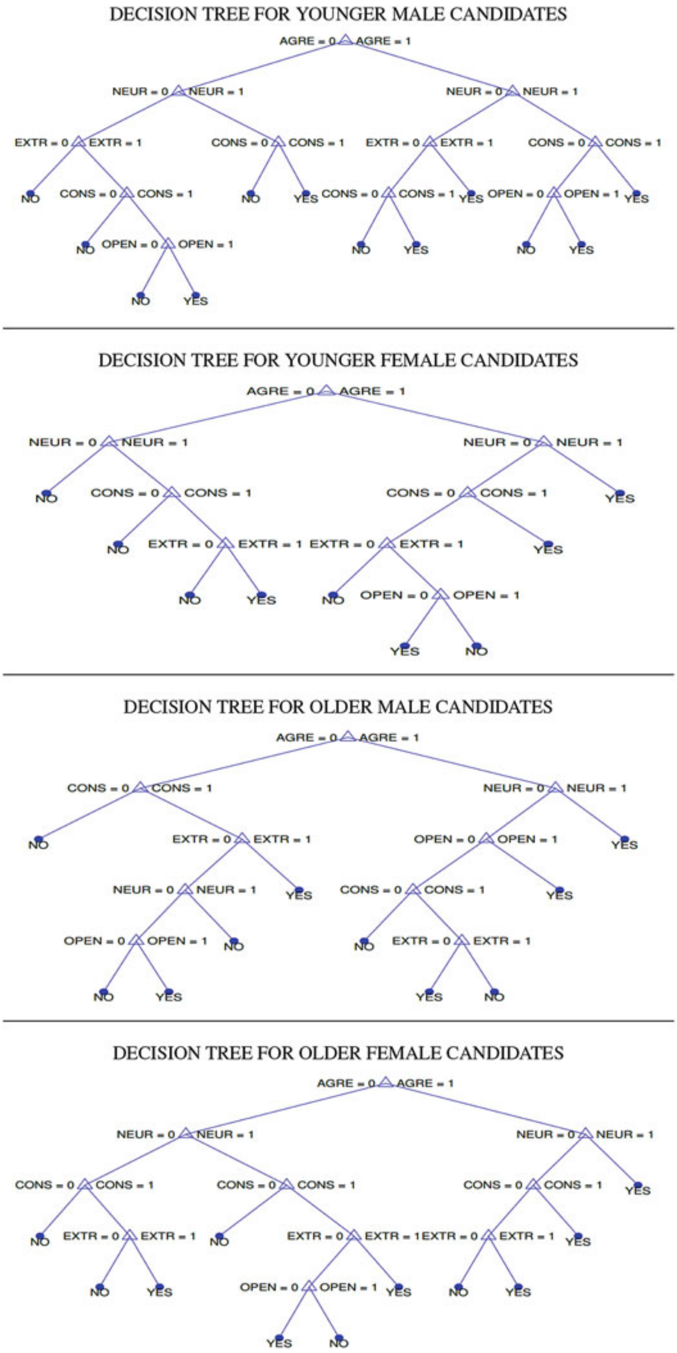
The test set scores of the official competition for this stage are shown in Table 5. Our team ranked the first in terms of the overall mean score. However, since the first runner up has better Creativity scores and the mean scores are not significantly different, both teams are designated as winners.

6.1 The Effect of Ethnicity, Age, and Sex

Automatic machine learning approaches that rely on human-supplied labels for supervised learning are prone to learn the biases inherent in these labels. To investigate potential biases in job interview screening, 10,000 videos of the ChaLearn corpus are annotated for apparent ethnicity, age, and sex in Escalante et al. (2018). It is shown that people who originally annotated the corpus for the interview variable are negatively biased toward African-Americans, while being positively biased towards Caucasians, both in terms of personality traits and the interview variable.

When biases for age and sex are investigated, they are found to be strongly correlated. As can be expected, the prior probability of job interview invitation is lower than 0.5 for people who are outside the working-age group, i.e. not in the age range of [18, 60]. Within the working-age group, the prior probability of job invitation is positively (and strongly) correlated with age of male candidates, while it is negatively correlated with the age of female candidates. In other words, the annotators prefer younger female candidates and older male candidates for invitation to a job interview.

We have analyzed how the proposed explanation system varies with respect to apparent age group and sex combinations. To preserve simplicity, we thresholded the working-age group at the age of 33, thus having a younger working age group with range [18, 32] and an older age group with range [33, 60]. With two age groups and two different sexes, we trained four decision trees. The results are shown in Fig. 6. We observe that while all trees are different in structure, they all


DECISION TREE FOR YOUNGER MALE CANDIDATES

DECISION TREE FOR YOUNGER FEMALE CANDIDATES

DECISION TREE FOR OLDER MALE CANDIDATES

DECISION TREE FOR OLDER FEMALE CANDIDATES

Fig. 6 Visualization of age group and sex dependent decision trees to be used as explanation models. Here, NEUR refers to non-Neuroticism

have Agreeableness at their root node, which indicates the importance of this cue for invitation to interview. Moreover, the importance ordering of the variables (i.e. apparent personality traits) imposed by the DTs for females are the same as those obtained from the whole dataset (given in Fig. 4).

7 Discussion and Conclusions

In this chapter, we have discussed an automatic system for the multimedia job candidate screening task. The proposed multi-level fusion framework uses multimodal fusion followed by a decision tree (DT), in order to produce text-based explanations of its decisions. These decisions are largely based on apparent personality predictions, which the system reports as intermediate results, but beyond that, the internal dynamics are not investigated for explainability. The proposed system ranked the first in both quantitative and qualitative stages of the official Challenge.

The scenario tackled in this chapter and in the related ChaLearn challenge is a limited case, where only passively recorded videos are available, as opposed to dyadic interactions. Subsequently, this scenario is more adequate to investigate first impression judgments, which are known to be very fast in their production, and very influential in behavior (Willis and Todorov 2006). There is a recent trend to ask job candidates to submit video resumes for job applications, and a widely held belief that such a format, being richer than a paper resume, will give a better leverage for the assessor to judge the personality of the candidate. Apers and Derous (2017) recently reported some results that illustrate that both paper resumes and video resumes are inadequate for judging the real personality of a candidate. But there is no doubt that they influence the recruiter's decisions, so the impact on the first impressions needs to be taken into account.

There is substantial research on first impressions, linking these to judgments of competence. Research on stereotype judgments put forward that two dimensions, posited as universal dimensions of human social cognition, particularly capture stereotype judgments, namely, warmth and competence (Fiske et al. 2002). These dimensions are, for instance, helpful to describe Western stereotypes against elderly (i.e. high warmth and low competence), or against Asians (i.e. high competence and low warmth). The warmth dimension predicts whether the interpersonal judgment is positive or negative (i.e. the valence of the impression), whereas the competence dimension quantifies the strength of this impression (Fiske et al. 2007). In an interesting study, Agerström et al. (2012) investigated 5636 job applications by Swedish and Arab applicants, and found substantial discrimination where Arab applicants receive fewer invitations to job interviews. The authors used the warmth-competence model to suggest that the Arab applicants need to "appear warmer and more competent than Swedish applicants to be invited equally often," but how exactly this can be achieved is an open question. Automatic analysis tools, if they can properly quantify such perceived qualities, can act as useful training tools.

There is further research on stigmatizing features that give the applicant a distinct disadvantage during a job interview. Examples of such features include obesity (Agerström and Rooth 2011), physical unattractiveness (Dipboye 2005), and visible disabilities (Hayes and Macan 1997). An automatic system that can accurately predict how such biases will effect decisions can be a useful tool in combatting these biases.

One of the limitations of the automatic job assessment task is that it considers only the applicant. However, any biases that exist on the interviewer's side are also essential in assessing the quality of this process (Dipboye et al. 2012). Future work should therefore ideally capture both the interviewer and the applicant during interactions. In particular, both the expertise and the confidence of the interviewer in their hiring decision need to be recorded to properly analyze the strength of the biases in the assessment.

Acknowledgements This work is supported by Boğaziçi University Project BAP 16A01P4 and by the BAGEP Award of the Science Academy.

References

Agerström J, Rooth DO (2011) The role of automatic obesity stereotypes in real hiring discrimination. Journal of Applied Psychology 96(4):790–805

Agerström J, Björklund F, Carlsson R, Rooth DO (2012) Warm and competent Hassan= cold and incompetent Eric: A harsh equation of real-life hiring discrimination. Basic and Applied Social Psychology 34(4):359–366

Alam F, Riccardi G (2014) Predicting personality traits using multimodal information. In: Proceedings of the 2014 ACM Multi Media on Workshop on Computational Personality Recognition, ACM, pp 15–18

Alam F, Stepanov EA, Riccardi G (2013) Personality traits recognition on social network-Facebook. WCPR (ICWSM-13), Cambridge, MA, USA

Almaev TR, Valstar MF (2013) Local Gabor binary patterns from three orthogonal planes for automatic facial expression recognition. In: Humaine Association Conference on Affective Computing and Intelligent Interaction, IEEE, pp 356–361

Apers C, Derous E (2017) Are they accurate? recruiters' personality judgments in paper versus video resumes. Computers in Human Behavior 73:9–19

Barezi EJ, Kampman O, Bertero D, Fung P (2018) Investigating audio, visual, and text fusion methods for end-to-end automatic personality prediction. arXiv preprint arXiv:180500705

Breiman L (2001) Random forests. Machine learning 45(1):5–32

Celiktutan O, Gunes H (2016) Automatic prediction of impressions in time and across varying context: Personality, attractiveness and likeability. IEEE Transactions on Affective Computing 8(1):29–42, https://doi.org/10.1109/TAFFC.2015.2513401

Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, vol 1, pp 886–893

Dipboye RL (2005) Looking the part: Bias against the physically unattractive as discrimination issue. Discrimination at work: The psychological and organizational bases pp 281–301

Dipboye RL, Macan T, Shahani-Denning C (2012) The selection interview from the interviewer and applicant perspectives: Can't have one without the other. The Oxford handbook of personnel assessment and selection pp 323–352

Escalante HJ, Ponce-López V, Wan J, Riegler MA, Chen B, Clapés A, Escalera S, Guyon I, Baró X, Halvorsen P, et al (2016) Chalearn joint contest on multimedia challenges beyond visual analysis: An overview. In: Pattern Recognition (ICPR), 2016 23rd International Conference on, IEEE, pp 67–73

Escalante HJ, Guyon I, Escalera S, Jacques J, Madadi M, Baró X, Ayache S, Viegas E, Güçlütürk Y, Güçlü U, et al (2017) Design of an explainable machine learning challenge for video interviews. In: Neural Networks (IJCNN), 2017 International Joint Conference on, IEEE, pp 3688–3695

Escalante HJ, Kaya H, Salah AA, Escalera S, Gucluturk Y, Guclu U, Baro X, Guyon I, Junior JJ, Madadi M, et al (2018) Explaining first impressions: Modeling, recognizing, and explaining apparent personality from videos. arXiv preprint arXiv:180200745

Eyben F, Wöllmer M, Schuller B (2010) OpenSMILE: the Munich versatile and fast open-source audio feature extractor. In: ACM International Conference on Multimedia, pp 1459–1462

Farnadi G, Sushmita S, Sitaraman G, Ton N, De Cock M, Davalos S (2014) A multivariate regression approach to personality impression recognition of vloggers. In: Proceedings of the 2014 ACM Multi Media on Workshop on Computational Personality Recognition, ACM, pp 1–6

Fernando T, et al (2016) Persons' personality traits recognition using machine learning algorithms and image processing techniques. Advances in Computer Science: an International Journal 5(1):40–44

Fiske ST, Cuddy AJ, Glick P, Xu J (2002) A model of (often mixed) stereotype content: competence and warmth respectively follow from perceived status and competition. Journal of personality and social psychology 82(6):878–902

Fiske ST, Cuddy AJ, Glick P (2007) Universal dimensions of social cognition: Warmth and competence. Trends in cognitive sciences 11(2):77–83

Gievska S, Koroveshovski K (2014) The impact of affective verbal content on predicting personality impressions in youtube videos. In: Proceedings of the 2014 ACM Multi Media on Workshop on Computational Personality Recognition, ACM, pp 19–22

Goodfellow IJ, Erhan D, Carrier PL, Courville A, Mirza M, Hamner B, Cukierski W, Tang Y, Thaler D, Lee DH, et al (2013) Challenges in representation learning: A report on three machine learning contests. In: International Conference on Neural Information Processing, Springer, pp 117–124

Güçlütürk Y, Güçlü U, van Gerven M, van Lier R (2016) Deep impression: Audiovisual deep residual networks for multimodal apparent personality trait recognition. In: ChaLearn Looking at People Workshop on Apparent Personality Analysis, ECCV Workshop Proceedings, pp 349–358

Güçlütürk Y, Güçlü U, Baro X, Escalante HJ, Guyon I, Escalera S, van Gerven MAJ, van Lier R (2017) Multimodal first impression analysis with deep residual networks. IEEE Transactions on Affective Computing, online https://doi.org/10.1109/TAFFC.2017.2751469

Gürpınar F, Kaya H, Salah AA (2016a) Combining deep facial and ambient features for first impression estimation. In: ChaLearn Looking at People Workshop on Apparent Personality Analysis, ECCV Workshop Proceedings, pp 372–385

Gürpınar F, Kaya H, Salah AA (2016b) Multimodal Fusion of Audio, Scene, and Face Features for First Impression Estimation. In: ChaLearn Joint Contest and Workshop on Multimedia Challenges Beyond Visual Analysis, Collocated with ICPR 2016, Cancun, Mexico

Hayes TL, Macan TH (1997) Comparison of the factors influencing interviewer hiring decisions for applicants with and those without disabilities. Journal of Business and Psychology 11(3):357–371

Huang GB, Zhu QY, Siew CK (2004) Extreme Learning Machine: a new learning scheme of feedforward neural networks. In: IEEE International Joint Conference on Neural Networks, vol 2, pp 985–990

Junior JCSJ, Güçlütürk Y, Perez M, Güçlü U, Andujar C, Baro X, Escalante HJ, Guyon I, van Gerven MAJ, van Lier R, Escalera S (2018) First impressions: A survey on computer vision-based apparent personality trait analysis. arXiv preprint arXiv:180408046 URL https://arxiv.org/abs/1804.08046

Kaya H, Salah AA (2014) Continuous mapping of personality traits: A novel challenge and failure conditions. In: Proceedings of the 2014 ICMI Workshop on Mapping Personality Traits Challenge, ACM, pp 17–24

Kaya H, Gürpınar F, Salah AA (2017a) Multi-modal Score Fusion and Decision Trees for Explainable Automatic Job Candidate Screening from Video CVs. In: CVPR Workshops, Honolulu, Hawaii, USA, pp 1651–1659

Kaya H, Gürpınar F, Salah AA (2017b) Video-based emotion recognition in the wild using deep transfer learning and score fusion. Image and Vision Computing 65:66–75, DOI http://dx.doi.org/10.1016/j.imavis.2017.01.012

Larkin JC, Pines HA (1979) No fat persons need apply: experimental studies of the overweight stereotype and hiring preference. Sociology of Work and Occupations 6(3):312–327

Lopez VP, Chen B, Places A, Oliu M, Corneanu C, Baro X, Escalante HJ, Guyon I, Escalera S (2016) Chalearn lap 2016: First round challenge on first impressions - dataset and results. In: ChaLearn Looking at People Workshop on Apparent Personality Analysis, ECCV Workshop Proceedings, Springer, pp 400–418

Lowe DG (2004) Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2):91–110

Madzlan N, Han J, Bonin F, Campbell N (2014) Towards automatic recognition of attitudes: Prosodic analysis of video blogs. Speech Prosody, Dublin, Ireland pp 91–94

Nowson S, Gill AJ (2014) Look! who's talking?: Projection of extraversion across different social contexts. In: Proceedings of the 2014 ACM Multi Media on Workshop on Computational Personality Recognition, ACM, pp 23–26

Ojala T, Pietikainen M, Maenpaa T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7):971–987

Pan SJ, Yang Q (2010) A survey on transfer learning. IEEE Transactions on knowledge and data engineering 22(10):1345–1359

Parkhi OM, Vedaldi A, Zisserman A (2015) Deep face recognition. In: British Machine Vision Conference

Qin R, Gao W, Xu H, Hu Z (2016) Modern physiognomy: An investigation on predicting personality traits and intelligence from the human face. arXiv preprint arXiv:160407499

Sarkar C, Bhatia S, Agarwal A, Li J (2014) Feature analysis for computational personality recognition using youtube personality data set. In: Proceedings of the 2014 ACM Multi Media on Workshop on Computational Personality Recognition, ACM, pp 11–14

Schuller B, Steidl S, Batliner A, Nöth E, Vinciarelli A, Burkhardt F, Van Son R, Weninger F, Eyben F, Bocklet T, et al (2012) The INTERSPEECH 2012 speaker trait challenge. In: INTERSPEECH, pp 254–257

Schuller B, Steidl S, Batliner A, Vinciarelli A, Scherer K, Ringeval F, Chetouani M, Weninger F, Eyben F, Marchi E, Mortillaro M, Salamin H, Polychroniou A, Valente F, Kim S (2013) The INTERSPEECH 2013 Computational Paralinguistics Challenge: Social Signals, Conflict, Emotion, Autism. In: INTERSPEECH, Lyon, France, pp 148–152

Sidorov M, Ultes S, Schmitt A (2014) Automatic recognition of personality traits: A multimodal approach. In: Proceedings of the 2014 Workshop on Mapping Personality Traits Challenge and Workshop, ACM, pp 11–15

Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556

Valente F, Kim S, Motlicek P (2012) Annotation and recognition of personality traits in spoken conversations from the ami meetings corpus. In: INTERSPEECH, pp 1183–1186

Willis J, Todorov A (2006) First impressions: Making up your mind after a 100-ms exposure to a face. Psychological science 17(7):592–598

Xiong X, De la Torre F (2013) Supervised Descent Method and Its Application to Face Alignment. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 532–539

Zhang CL, Zhang H, Wei XS, Wu J (2016) Deep bimodal regression for apparent personality analysis. In: ChaLearn Looking at People Workshop on Apparent Personality Analysis, ECCV Workshop Proceedings, pp 311–324

On the Inherent Explainability of Pattern Theory-Based Video Event Interpretations

Sathyanarayanan N. Aakur, Fillipe D. M. de Souza, and Sudeep Sarkar

Abstract The ability of artificial intelligence systems to offer explanations for its decisions is central to building user confidence and structuring smart human-machine interactions. Expressing the rationale behind such a system's output is an important aspect of human-machine interaction as AI continues to be prominent in general, everyday use-cases. In this paper, we introduce a novel framework integrating Grenander's pattern theory structures to produce inherently explainable, symbolic representations for activity interpretations. These representations provide semantically rich and coherent interpretations of video activity using connected structures of detected (grounded) concepts, such as objects and actions, that are bound by semantics through background concepts not directly observed, i.e. contextualization cues. We use contextualization cues to establish semantic relationships among concepts to infer a deeper interpretation of events than what can be directly sensed. We propose the use of six questions that can be used to gain insight into the models ability to justify its decision and enhance its ability to interact with humans. The six questions are designed to (1) build an understanding of how the model is able to infer interpretations, (2) enable us to walk through its decision-making process, and (3) understand its drawbacks and possibly address them. We demonstrate the viability of this idea on video data using a dialog model that uses interpretations to generate explanations grounded in both video data and semantics.

Keywords Explainability · Activity interpretation · ConceptNet · Semantics

S. N. Aakur (✉) · F. D. M. de Souza · S. Sarkar
University of South Florida, Department of Computer Science and Engineering, Tampa, FL, USA
e-mail: saakur@mail.usf.edu; fillipe@mail.usf.edu; sarkar@usf.edu

H. J. Escalante et al. (eds.), *Explainable and Interpretable Models in Computer Vision and Machine Learning*, The Springer Series on Challenges in Machine Learning, https://doi.org/10.1007/978-3-319-98131-4_11

1 Introduction

Intelligent agents, especially those based on deep learning, have evolved tremendously and have achieved significant milestones approaching human capabilities in some domains (Kheradpisheh et al. 2016). Such massive strides in machine learning hold much promise in the development of autonomous agents that are capable of learning, perceiving and acting on their own. However, despite these performance gains, their ability to *explain* their decision appears to be constrained.

The strength of artificial intelligence systems to offer explanations for its decisions is central to building user confidence and structuring smart human-machine interactions. Understanding the rationale behind such a system's output helps in making an informed action based on a model's prediction. It becomes even more vital to understand the model's decision process when dealing with uncertainty in the input. This need for understanding is especially real as uncertainty can arise at various levels. It can occur at a lower level, at the input level, such as due to sensor degradation. It can happen at the higher level of abstraction such as at the concept detection stage, due to low visibility, occlusion, lower quality, or at semantic levels, due to ambiguity in establishing semantic relationships.

Ideally, any intelligent system must account for uncertainty at any level and possess sufficient flexibility to handle uncertainty at any of these levels. For example, when taking vital decisions in high-risk areas like medical diagnosis (Caruana et al. 2015; Linder et al. 2014) and surveillance (Mahadevan et al. 2010; Junior et al. 2010), the level of interaction between the human and a model is of high importance. It has also been established that a model with higher explainability is more likely to be trusted (Ribeiro et al. 2016) than a model with limited or no explainability.

A model's ability to provide sufficient justification for its decision requires in-depth knowledge about various concepts and the semantic relationships that they share with other concepts. This use of prior knowledge can be considered to be analogous to how humans correlate the presence of certain concepts to aid in the current task. For example, in medical diagnosis (Ledley et al. 1959), it has been noted that the reasoning process used by doctors requires the establishment of correlation between symptoms (logical concepts) and probabilities to aid their diagnosis. Each symptom adds a certain value to the overall diagnosis and hence influences the direction of the reasoning process. This prior knowledge can be particularly helpful in identifying *how* two concepts can be related and *why* that relationship can contribute to the overall goal of the model.

Explainable models have been explored to some extent in literature. Spanning a variety of application domains such as medical diagnosis (Shortliffe and Buchanan 1975), modeling and recognizing personality (Escalante et al. 2018), activity simulations such as those in the military (Core et al. 2006; Lane et al. 2005) and robotics (Lomas et al. 2012), these approaches have advocated models that are able to explain the approach undertaken to arrive at decisions but were not able to *justify* their decision to the user. There also have been *model-agnostic* approaches such as Baehrens et al. (2010), Ribeiro et al. (2016) that attempt to explain the decision of machine learning models while treating them to be a black-box. However, some approaches, such as those advocated in Biran and McKeown (2014), Hendricks et al. (2016), are able to support their decisions with explanations justifying them with evidence from visual and semantic cues.

Extending the concept of explainability to event interpretation from videos, we consider an explanation to be a description that explains and justifies the rationale of a model's decision process. In addition to defending with respect to feature-level evidence, it should elucidate the semantic correlations among concepts that make up an activity (actions and objects). Unraveling these relationships is a fundamental aspect of explainability in event interpretations. In open, uncontrolled environments, establishing justifiable semantic correlation is integral to a model's success since the training data may not always be representative of all viable event that one may encounter.

Current deep learning-based methods, that model intelligence as a label mapping problem, adopt an entirely retrospective approach to the problem explainability. They look into techniques to visualize the learned network and imposes human interpretations on various aspects of what it is accomplishing. This visualization of learned weight works well with lower layers of the network, where low-level features are detected, but the interpretations get speculative and, many times anthropocentric, as one tries to explain higher levels of the network. We adopt an entirely different approach. We accept the success of deep learning to assign putative labels to objects and actions in the video but form higher-level semantic interpretation about the event using the probabilistic symbolic approach of pattern theory in light of commonsense knowledge. We develop a flexible architecture that is both introspective, expressing the mechanics behind the model's decision-making process, as well as retrospective in the model's ability to justify its decision. Based on Grenander's Pattern Generator Theory, we aim to capture the underlying structure of patterns and produces inherently explainable, semantically coherent interpretations of video activity.

In this paper, we propose a novel framework that leverages Grenander's Pattern Theory structures (Grenander 1996) to infer semantically coherent interpretations of video activity. An interpretation is defined as a semantically linked structure of concepts. It is an intermediate representation that can be considered to be the underlying source of knowledge for more expressive representations such as sentence-based descriptions and/or question and answers systems. In pattern theory language, concepts are represented by basic elements called generators with their semantic relationships represented by connections called bonds. Some concepts in this representation possess direct evidence from video, i.e. grounded concepts, while some are inferred concepts called contextualization cues. As defined by Gumperz (1992), primarily for linguistics, contextualization refers to the use of knowledge acquired from past experience to retrieve *presuppositions* required to maintain involvement in the current task. It has also been observed that providing

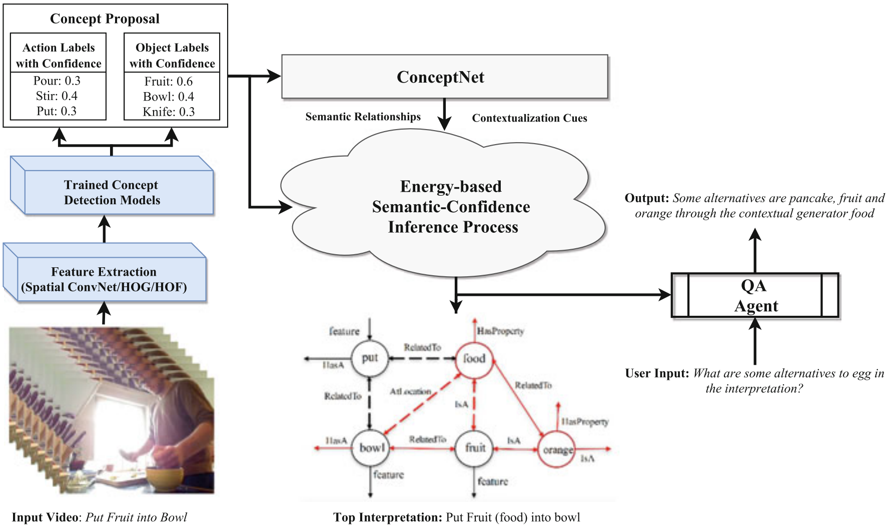

Concept Proposal
Action Labels with Confidence
Pour: 0.3
Stir: 0.4
Put: 0.3
Object Labels with Confidence
Fruit: 0.6
Bowl: 0.4
Knife: 0.3
Trained Concept Detection Models
Feature Extraction (Spatial ConvNet/HOG/HOF)
Input Video: Put Fruit into Bowl
ConceptNet
Semantic Relationships
Contextualization Cues
Energy-based Semantic-Confidence Inference Process
feature
put
HasA
RelatedTo
food
HasProperty
AtLocation
IsA
RelatedTo
HasProperty
HasA
bowl
RelatedTo
fruit
IsA
orange
IsA
feature
feature
Top Interpretation: Put Fruit (food) into bowl
QA Agent
Output: Some alternatives are pancake, fruit and orange through the contextual generator food
User Input: What are some alternatives to egg in the interpretation?

Fig. 1 Overall architecture. Deep learning or machine learning-based approaches hypothesize multiple object and action labels. Pattern theory formalism disambiguates knowledge using ConceptNet to generate an interpretation. An interactive agent then uses this as a source of knowledge for conversation about the inference process

contextualization cues often result in increase in acceptance of decisions made by automated systems (Herlocker et al. 2000; Martens and Provost 2013). In addition to providing additional knowledge to the model that aids in establishing semantic relationships among concepts, it also allows us to help address dataset shift to a certain extent. Dataset shift (Quionero-Candela et al. 2009) refers to the possible disconnect that can exist between training data and test data.

The overall architecture of the proposed approach is shown in Fig. 1. Given an input video, individual, atomic concepts such as actions and objects are hypothesized using machine-learning or deep learning approaches. The resulting, multiple putative labels per object instance are then used to generate interpretations using an Markov Chain Monte-Carlo (MCMC) based simulated annealing process. The most likely interpretations are then used as the source of knowledge for generating explanations for human interaction via a dialog model.

The contribution of this paper is threefold: (1) we are, to the best of our knowledge, among the first to address the issue of explainability in video activity interpretation; (2) the use of contextualization cues allows us to generate interpretations that is able to provide sufficient information to generate explanations at different levels of abstraction—from feature-level evidence (through grounded generators) to semantic relations (via bonds); and (3) we are able to show, through a dialog model, that the proposed framework is capable of generating explanations for its decision making process that is both introspective and retrospective.

2 Explainable Model for Video Interpretation

Grenander's formalism allows us to express interpretations in an inherently explainable manner facilitating better human interaction. We begin with discussion about how concepts (such as actions and objects) are represented as *generators* and the different types of generators that can exist. We follow with discussion on how the detected concepts are grounded with semantic provenance using contextualization cues generated from a commonsense knowledge base known as ConceptNet (Liu and Singh 2004). We, then, follow with discussion on how generators are connected together using connections called *bonds* to form video interpretations called *configurations*. Finally, we discuss the Monte-Carlo based inference process.

2.1 Symbolic Representation of Concepts

Following Grenander's notations (Grenander 1996), we represent each concept using atomic components called `generators` $g_i \in G_S$ where G_S is called the `generator space`. The generator space represents a finite collection of all possible generators that can exist in a given environment.

The generator space (G_S) consists of three disjoint subsets that represent three kinds of generators—feature generators (F), grounded concept generators (G) and ungrounded context generators (C) such that

$$G_S = \{F \cup G \cup C; \; F \cap C \cap S = \emptyset\}. \tag{1}$$

Feature generators ($g_{f_1}, g_{f_2}, g_{f_3}, \ldots, g_{f_q} \in F$) correspond to the features extracted from videos and are used to infer the presence of the basic concepts (actions and objects) called `grounded concept generators` ($\underline{g}_1, \underline{g}_2, \underline{g}_3, \ldots, \underline{g}_k \in G$). Individual units of information that represent the background knowledge of these grounded concept generators are called `ungrounded context generators` ($\bar{g}_1, \bar{g}_2, \bar{g}_3, \ldots, \bar{g}_q \in C$). The term *grounding* is used to distinguish between generators with direct evidence in the video data and those that define the background knowledge of these concepts.

Each type of generator is a source of knowledge for generating explanations and hence contributes to the overall interpretation's inherent explainability. For example, the feature generators allow the model to establish and express provenance for grounded concept generators in the actual input data. Hence, the model is able to provide direct video evidence for the presence of the grounded concept generators in the final configuration. The ungrounded context generators represent the additional, background knowledge that allow us to *semantically* correlate the presence of the grounded concept generators and hence help provide semantic justification for the presence of a concept in the final interpretation.

The semantic content of an interpretation is expressed through the presence of grounded concept generators. Continuing the analogy from Sect. 1, we can consider the grounded concept generators to be the actual diagnosis whereas the ungrounded context generators represent symptoms that the patient possesses. The feature generators represent the actual, physical attributes that give support to the diagnosis.

2.2 Constructing Contextualization Cues

In the context of video activity recognition, we propose the use of a commonsense knowledge base as a source of contextualization cues for establishing semantic relationships among concepts. ConceptNet, proposed by Liu and Singh (2004) and expanded to ConceptNet5 (Speer and Havasi 2013), is a knowledge source that maps concepts and their semantic relationships in a traversable semantic network structure. Spanning more than three million concepts, the ConceptNet framework serves as a source of cross-domain semantic information from general human knowledge while supporting commonsense knowledge as expressed by humans in natural language. Technically, it encodes and expresses knowledge in a hypergraph, with the nodes representing concepts and edges representing semantic assertions.

There are more than 25 relations (also referred to as assertions) by which the different nodes are connected, with each of these relations contributing to the semantic relationship between the two concepts, such as `HasProperty`, `IsA`, and `RelatedTo`. The validity of each assertion in ConceptNet is quantified by a weighted score and is representative of the semantic relation between concepts. Positive values indicates assertions and negative values indicates the opposite.

2.3 Expressing Semantic Relationships

Each generator g_i has a fixed number of bonds called the `arity` of a generator ($w(g_i) \forall g_i \in G_S$). These bonds are symbolic representations of the semantic relationships shared between generators. Bonds are differentiated at a structural level by the direction of information flow that they represent—*in-bonds* and *out-bonds*. Each bond is identified by a unique coordinate and bond value such that the j^{th} bond of a generator $g_i \in G_S$ is denoted as $\beta^{j}_{dir}(g_i)$, where dir denotes the direction of the bond. A bond is said to be `open` if it is not connected to another generator through a complementary bond. For example, in Fig. 2 there exist a bonded generator pair $\{$*pour* and *liquid*$\}$. The bonds representing *HasProperty* and *HasA* are *open*, whereas the bond labeled *RelatedTo* represents a *closed* bond between the generators “pour” and “liquid”. The closed bond indicates that there exists a quantifiable, semantic relationship between two generators and hence contributes to the overall semantic content of the interpretation.

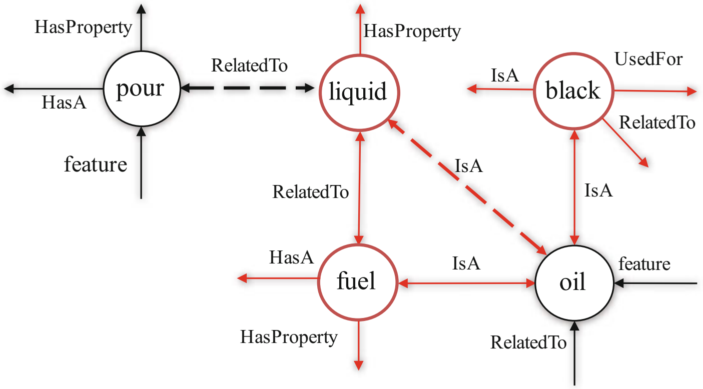


Fig. 2 Representation of an interpretation using pattern theory. Black circles are generators that represent grounded concepts and red generators represent ungrounded concepts i.e. contextualization cues. The red links represent contextual bonds. Dashed links represent the optimal relationship between concepts

2.3.1 Bond Compatibility

The viability of a *closed* bond between two generators is determined by the `bond relation function` ρ. This function determines whether two bonds $\beta(g_i)$ and $\beta(g_j)$ between two generators, g_i and g_j, are compatible and is denoted by $\rho[\beta(g_i), \beta(g_j)]$ or simply as $\rho[\beta, \beta]$. This function is used to determine whether a given bond $\beta^j_{dir}(g_i)$ is either *closed* or *open*. The bond relation function is given by

$$\rho[\beta(g_i), \beta(g_j)] = \{TRUE,\ FALSE\}; \forall g_i, g_j \in G_S \tag{2}$$

If the two bonds are determined to be compatible, then the bond is established and quantified using a bond energy function. If the bonds are not compatible, then an connection is not established and the bonds are left *open*.

2.3.2 Types

There exist two types of bonds—`semantic bonds` and `support bonds`. Each *closed* bond (both semantic and support) is a symbolic representation of a concept's ties to the interpretation and are used as guiding cues for generating explanations. The direction of *semantic bonds* signify the semantics of a concept and the type of relationship a particular generator shares with its bonded generator or concept. These bonds are analogous to the assertions present in the ConceptNet framework. For example, Fig. 2 illustrates an example configuration with *pour*, *oil*, *liquid*, etc. representing generators and the connections between them, given by *RelatedTo*, *IsA*, etc., representing the semantic bonds. *Support bonds* connect (grounded) concept generators to feature generators and are representative of direct image evidence for the grounded concept generator. These bonds are quantified using confidence scores from classification models.

2.3.3 Quantification

The bonds between the generators are quantified using the strength of the semantic relationships between generators. This allows to quantify the amount of contribution the generator provides to the interpretation. The bond energy is quantified by the bond energy function:

$$a(\beta'(g_i), \beta''(g_j)) = q(g_i, g_j)\,\tanh(f(g_i, g_j)). \tag{3}$$

where $f(.)$ is the weight associated with the relation in ConceptNet between concepts g_i and g_j through their respective bonds β' and β''. The tanh function normalizes the score output by $f(.)$ to range from -1 to 1. $q(g_i, g_j)$ weights the score output by the $tanh$ function according to the bond connection type (e.g., semantic or support) β' and β'' formed. This helps us tradeoff between direct evidence and prior knowledge, i.e. context. If we want direct evidence to play a larger role in the inference we choose a larger value for $q(g_i, g_j)$ for support bonds than semantic bonds, and vice-versa.

The decision about the bond type parameters $q(g_i, g_j)$ can be as simple as defining number between zero and one or higher to accommodate the strengths and weaknesses of decision making based on the priors or observations. The parameters can be trained using a grid search for varying the parameters and either cross-validation or a test and training sets approach for evaluation. In the past (Souza et al. 2015), the choice of these parameters were evaluated and shown to have considerable impact on the final interpretation when observations and priors are not equally reliable. If decisions based on observations are reliable, one can safely choose high values for $q(g_i, g_j)$ for support bonds. Otherwise, semantic bonds can have more influence on evaluating the semantic quality of the interpretation. In such cases, however, the energy from support bonds would still be used and helps reduce the inherent bias from the prior knowledge.

2.4 Constructing Interpretations

Generators can be combined together through their local bond structures to form composite structures called *configurations* c, which, in our case, represent semantic interpretations of video activities. Each configuration has an underlying graph topology, specified by a connector graph σ. The set of all feasible connector graphs σ is denoted by Σ, also known as the connection type. Formally, a configuration c is a connector graph σ whose sites $1, 2, \ldots, n$ are populated by a collection of generators $g_1, g_2, \ldots, g_n$ expressed as $\sigma(g_1, g_2, \ldots, g_i)$. The collection of generators $g_1, g_2, \ldots, g_i$ represents the semantic content of a given configuration c. For example, the collection of generators from the configuration in Fig. 2 gives rise to the semantic content "*pour oil (liquid) (fuel) (black)*".

2.4.1 Probability

The probability of a particular configuration c is determined by its energy as given by the relation

$$P(c) \propto e^{-E(c)} \tag{4}$$

where $E(c)$ represents the total energy of the configuration c. The energy $E(c)$ of a configuration c is the sum of the bond energies formed by the bond connections that combine the generators in the configuration, as described in Eq. (3).

$$E(c) = -\sum_{(\beta',\beta'')\in c} a(\beta'(g_i), \beta''(g_j)) + k \sum_{\bar{g}_i \in G'} \sum_{\beta^j_{out} \in \bar{g}_i} [D(\beta^j_{out}(\bar{g}_i))] \tag{5}$$

where the first term is a summation over all *closed* bonds in the configuration and the second term is a summation over all the *open* bonds of all ungrounded concept generators; G' is a collection of ungrounded contextual generators present in the configuration c, β_{out} represents each *out-bond* of each generator g_i and D(.) is a function that returns true if the given bond is open; k is an arbitrary constant that quantifies the extent of the detrimental effect that the ungrounded context generators have on the quality of the interpretation.

The second term of Eq. (5) is a quality cost function for a given configuration c. The cost factor restricts the inference process from constructing configurations with degenerate cases such as those composed of unconnected or isolated generators that do not have any closed bonds and as such do not connect to other generators in the configuration. It also prevents the inference from spawning generators that do not add semantic value to the overall quality of the configuration thereby reducing the search space to arrive at an optimal interpretation. The parameter k determines how much penalty the interpretation pays for allowing ungrounded generators with many open bonds to be present the interpretation. Open bonds indicate lower semantic compatibility with semantics of the interpretation. The optimal value for the k was chosen using a simple grid search from having no impact ($k = 0$) to having higher restriction ($k = 10$). Very low values tend to produce configurations with a high number of ungrounded generators, while very high values tend to favor configurations with no ungrounded generators to produce configurations whose grounded generators possess direct semantic relationships. In our experiments, we keep the value of k was kept constant at 1.

2.4.2 Inherent Explainability

The generated interpretations possess a level of self-explainability. The presence of a grounded concept generator can be explained directly in the data with the feature generators. Additionally, any established semantic correlations between grounded concept generators are explained through the presence of the ungrounded concept generators. For example, consider an example interpretation for a video with groundtruth "*Pour oil*" in Fig. 2, the presence of the grounded concept generators "pour" and "oil" are supported by the presence of the features "HOF" and "HOG" respectively. This can easily be traced back to direct video data evidence. The presence of the two grounded concept generators is further correlated by the presence of ungrounded context generator "liquid". Hence representing interpretations using Grenander's Pattern Theory framework provides a sense of inherent explainability further augmented by the use of contextualization cues.

2.5 Inference

Searching for the best semantic description of a video involves minimizing the energy function $E(c)$ and represents the inference process. The solution space spanned by the generator space is very large as both the number of generators and structures can be variable. For example, the combination of a single connector graph σ and a generator space G_S give rise to a space of feasible configurations $C(\sigma)$. While the structure of the configurations $c \in C(\sigma)$ is identical, their semantic content is varied due to the different assignments of generators to the sites of a connector graph σ. A feasible optimization solution for such exponentially large space, is to use a sampling strategy. We follow the work in de Souza et al. (2016) and employ a Markov Chain Monte Carlo (MCMC) based simulated annealing process. The MCMC based simulation method requires two types of proposal functions—global and local proposal functions.

A connector graph σ is given by a global proposal function which makes structural changes to the configuration that are reflected as jumps from a subspace to another. A swapping transformation is applied to switch the generators within a configuration to change of semantic content of a given configuration c. This results in a new configuration c', thus constituting a move in the configuration space $C(\sigma)$.

Initially, the global proposal function introduces a set of grounded concept generators derived from machine learning classifiers. Then, a set of ungrounded context generators, representing the contextualization cues, are populated for each grounded concept within the initial configuration. Bonds are established between compatible generators when each generator is added to the configuration. Each jump given by the local proposal function gives rise to a configuration whose semantic content represents a possible interpretation for the given video. Interpretations with the least energy are considered to have a higher probability of possessing a more semantic coherence.

There may arise certain cases in which for a given interpretation, many other plausible alternatives with same probability but different semantic content exist. These are critical cases and can aid in human-machine interaction to find a suitable resolution. In such cases, it is often necessary to quantify the certainty with which the model has arrived at a given interpretation. Given the search nature of the proposed algorithm, there can be multiple alternatives to a given interpretation. There are two ways to quantify the certainty: (1) a local conditional probability and (2) a global semantic certainty score.

The local conditional probability gives the degree of certainty of the model given alternative interpretation(s). For example, given two interpretations, c and an alternative c', the model's confidence in c is given by.

$$\hat{P}(c|c') = \frac{P(c)}{P(c')} \tag{6}$$

The global semantic certainty score provides the degree of certainty with which the model has arrived at the given interpretation given all the possible interpretations. This can be achieved through the normalization of the probability of an interpretation with all sampled interpretations. Hence, the semantic certainty score for a given interpretation c is given by,

$$\hat{P}(c) = \frac{exp[-E(c)]}{\sum\limits_{c' \in c_{sampled}} exp[-E(c')]} \tag{7}$$

where $c_{sampled}$ is the collection of all sample configurations in the inference process. The denominator term provides an approximation of the partition function Z, which provides the total energy of all possible global structures and all possible generator combination for any given global structure. The MCMC-based inference process allows us to sample the interpretations that provide an efficient approximation of the partition function. This normalizes the probability of a given configuration with respect to all possible configurations given the hypothesized (grounded) concepts and the background (ungrounded) concepts and hence provides an accurate portrayal of the model's certainty (or uncertainty) in the given interpretation c.

3 Generating Explanations

We are able to walk through the decision making process and express *why* it arrived at the interpretation as the most likely one. This allows the human to understand the reasoning behind the interpretation and provide a deeper understanding about how the interpretation is a viable explanation for the given video activity. In our current implementation, we allow for six questions that can be used to gain explanatory insight. A more general framework will be focus of future work. These questions allow us to evaluate the model's ability to *justify* its decision as well as enhance its ability to interact with humans. The six questions are designed to (1) build an understanding of how the model is able to infer interpretations for a given video, (2) enable us to walk through each aspect of its decision-making process, and (3) understand its drawbacks and possibly address them. They are the following:

1. *How did you arrive at the interpretation?* The model walks through the inference process starting at the feature level. The response enables the human to understand how each factor contributes to the interpretation and determine the point of failure (if any) for improving the performance.
2. *What are alternatives to the interpretation?* The answer to this question provides alternative interpretations. This allows the human to pick the best possible interpretation from the model. In a critical scenario, a human may need to choose an alternative interpretation rather than blindly trusting the model's top prediction.
3. *Why is <concept> in the interpretation?* The model looks for cues to justify the presence of a concept within its final interpretation. This provides a detailed justification for including a concept in the interpretations at both levels of abstraction—feature level and semantic level.
4. *What are alternatives to <concept> in the interpretation?* To answer this, we walk through the inference process to bring alternatives to the specific concept in the interpretation. This allows for better understanding of the inference process while providing an ideal point of interaction for understanding the model's capability to semantically associate different concepts in a coherent manner.
5. *Why not use <concept1> instead of <concept2>?* To answer this, we have to reason about alternatives. This interaction allows us to understand how the semantics influence its inference process.
6. *The correct interpretation is <interpretation>. Why did you not get there?*: This prompts the model to continue reasoning about its inference process and provide a concise argument about its choices.

Considered together, these questions cover various aspects about the decision making process and explain the rationale behind the output. An important observation to be noted is that these questions require the model to be able to relate concepts together beyond what may be visible in the video data and/or training data. Hence models is able to generate semantically coherent interpretations as well as provide semantic justification for the presence of concepts beyond feature-level evidence.

3.1 Understanding the Overall Interpretation (Q1)

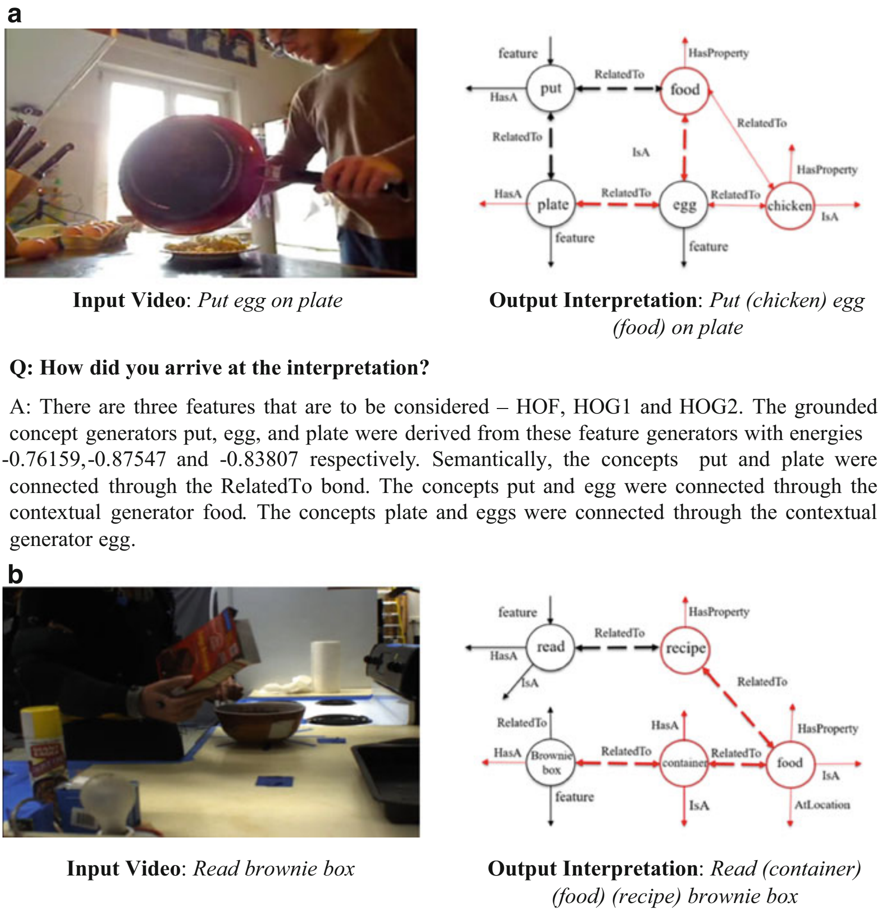
The first question, *How did you arrive at the interpretation?*, is an expression of the model's ability to express rationale behind its decision making process. Such explanations, through meaningful interactions, can aid in understanding the system's overall strengths and weaknesses and convey an understanding of how the system will behave in the future.

Algorithm 1: The algorithm for generating an explanation for the model's final (top) interpretation. Each call to the function returns two sets of tuples: one for data-based explanations $E_{support}$ and semantic bonds ($E_{semantic}$) for a given configuration c. The function $get\text{-}generators(\beta_j(g_i))$ returns a tuple consisting of the corresponding generator to a given bond $\beta_j(g_i)$ and the bond energy

```
ExplainInterpretation ($c$);
  $E_{support} \leftarrow \emptyset$
  $E_{semantic} \leftarrow \emptyset$
  $F_S \leftarrow \{g_{f_i} \in c\}$
  for $g_{f_i} \in F_S$: do
      $B' \leftarrow \{\beta_j(g_{f_i})\} \quad \forall \quad D(\beta_j(g_{f_i})) = 1$
      for $\beta_j(g_{f_i}) \in B'$: do
          $E_{support} \cup get\text{-}generators(\beta_j(g_{f_i}))$
      end
  end
  for $\underline{g}_i \in E_{support}$: do
      $B' \leftarrow \{\beta_j(\underline{g}_i)\} \quad \forall \quad D(\beta_j(\underline{g}_i)) = 1$
      for $\beta_j(\underline{g}_i) \in B'$: do
          $E_{semantic} \cup get\text{-}generators(\beta_j(\underline{g}_i)$
      end
  end
  return $E_{support}, E_{semantic}$
```

The algorithm for generating an explanation for the model's final interpretation is shown in Algorithm 1. Each call to the function returns two sets of tuples: one for data-based explanations $E_{support}$ and semantic bonds ($E_{semantic}$) for a given configuration c. The function $get\text{-}generators(\beta_j(g_i))$ returns a tuple consisting of the corresponding generator to a given bond $\beta_j(g_i)$ and the bond energy. The tuple is then iterated through and included into a template-based sentence generation model for generating the dialogues for interaction.

For example, consider Fig. 3a, where the model is prompted to explain its decision making process in the first interaction. It can be seen that the model begins with the factors that were considered in generating the interpretation—namely the feature generators HOF, HOG1 and HOG2 and continues with the labels chosen to represent these feature generators in the final interpretation as grounded concept generators as well as expressing the confidence levels in its choice of labels. Another, similar interaction is shown in Fig. 3b, where the model is presented with an input video with groundtruth "*Read brownie box*". It is interesting to note that the model is also able to justify the interpretation's overall meaning through the presence of the ungrounded context generators *container, food and recipe*; thus covering all aspect's of the model's inference process.



Input Video: *Put egg on plate*

Output Interpretation: *Put (chicken) egg (food) on plate*

Q: How did you arrive at the interpretation?

A: There are three features that are to be considered – HOF, HOG1 and HOG2. The grounded concept generators put, egg, and plate were derived from these feature generators with energies -0.76159,-0.87547 and -0.83807 respectively. Semantically, the concepts put and plate were connected through the RelatedTo bond. The concepts put and egg were connected through the contextual generator food. The concepts plate and eggs were connected through the contextual generator egg.

Input Video: *Read brownie box*

Output Interpretation: *Read (container) (food) (recipe) brownie box*

Q: How did you arrive at the interpretation?

A: There are two feature generators that are to be considered – CNN and CNNFlow. The grounded concept generators *read* and *brownie box* were derived from these feature generators with energies -0.03862 and -0.92633 respectively. Semantically, the concepts *read* and *brownie box* are connected through the contextual generators *container, food* and *recipe*.

Fig. 3 An illustration of example interactions with the proposed model is shown when provided with the question *"How did you arrive at the interpretation"* for two input videos (**a**) "*Putt egg on plate*" and (**b**) "*Read brownie box*". The model was able to walk through the interpretation and justify the presence of each concept in the interpretation at both semantic and data levels

3.2 Understanding Provenance of Concepts (Q3)

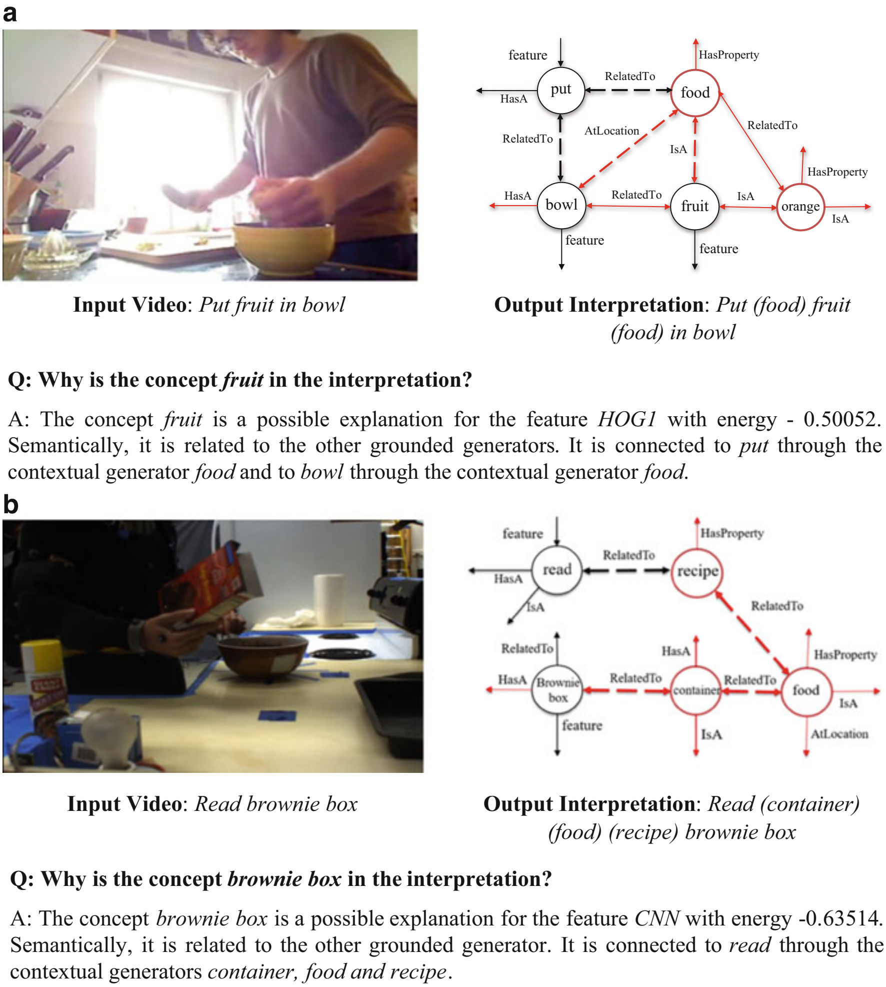
When dealing with complex video activities, understanding the rationale behind the presence of individual concepts in an interpretation is essential and requires meaningful explanations that explain the provenance of concepts. Often, it involves explanations that provide justification for the concepts that are both grounded with direct evidence from data as well as meaningful explanations that are not obvious in the video alone. In the proposed framework, interpretations are used as a source of knowledge to generate explanations for a concept's provenance. For example, consider the interpretation in Fig. 2 whose semantic content is *Pour oil*. The presence of the concept *oil* can be explained through the presence of its corresponding feature generator connected through the bond labeled *feature* as well as the ungrounded context generator *liquid*. Hence the resulting semantic explanation can be constructed as *"Oil can be poured because it is a liquid."*

Direct data evidence for the presence of the concept is provided through the presence of feature generators while semantic justification is derived using the bonds connecting ungrounded context generators. The algorithm for understanding the provenance of a concept is shown in Algorithm 2. Each call to the function returns a tuple for data-based provenance $E_{support}$ and semantics-based provenance ($E_{semantic}$) for a given configuration c and a concept g_i. We begin with the given concept generator. We traverse through the configuration based on the active, closed bonds present for the given generator. Then, the corresponding generator is taken and a tuple of consisting of the confidence score, semantic relationship and the generator is constructed and classified into either data-based or semantics-based explanation. The resulting tuple is then fed onto a template-based sentence generator for dialogue-based user interaction.

Algorithm 2: The algorithm for understanding the provenance of a concept. Each call to the function returns a tuple for data-based provenance $E_{support}$ and semantics-based provenance ($E_{semantic}$) for a given configuration c and a concept g_i

```
getProvenance ($c, g_i$);
  $E_{support} \leftarrow \emptyset$
  $E_{semantic} \leftarrow \emptyset$
  $G' \leftarrow \emptyset$
  $B' \leftarrow \{\beta_j(g_i)\} \quad \forall \quad D(\beta_j(g_i)) = 1$
  for $\beta_j(g_i) \in B'$: do
      $G' \cup get\text{-}generators(\beta_j(g_i))$
  end
  for $(\beta_j(g_i), g_j) \in G'$: do
      if $g_j \in F$ then
          $E_{support} \cup get\ \text{-}generator(\beta_j(g_i))$
      end
      else if $g_j \in G$ then
          $E_{support} \cup get\ \text{-}generator(\beta_j(g_i))$
      else if $g_j \in C$ then
          $E_{semantic} \cup get\ \text{-}generator(\beta_j(g_i))$

  end
  return $E_{support}, E_{semantic}$
```



Input Video: *Put fruit in bowl*

Output Interpretation: *Put (food) fruit (food) in bowl*

Q: Why is the concept *fruit* in the interpretation?

A: The concept *fruit* is a possible explanation for the feature *HOG1* with energy - 0.50052. Semantically, it is related to the other grounded generators. It is connected to *put* through the contextual generator *food* and to *bowl* through the contextual generator *food*.

Input Video: *Read brownie box*

Output Interpretation: *Read (container) (food) (recipe) brownie box*

Q: Why is the concept *brownie box* in the interpretation?

A: The concept *brownie box* is a possible explanation for the feature *CNN* with energy -0.63514. Semantically, it is related to the other grounded generator. It is connected to *read* through the contextual generators *container, food and recipe*.

Fig. 4 An illustration of example interactions with the proposed model is shown when provided with the question *"Why is <concept> in the interpretation?"* for two input videos (**a**) "*Put fruit into bowl*" and (**b**) "*Read brownie box*". The model was able to justify the presence of the specified concept in the interpretation at both semantic and data levels

Example interactions with the model when posed with the question "*Why is the concept egg in the interpretation*" for the video, "*put fruit into bowl*" in Fig. 4a. The model is able to contend the presence of the concept "*fruit*" by reasoning through different levels of provenances, such as the feature generator *HOG1* and its corresponding confidence level indicated by the energy *-0.50052*. The semantic justification for the concept is provided through the presence of the ungrounded concept generators *food* which allows it to establish a semantic connection to other grounded concept generator *put*.

Another interaction is shown in Fig. 4b, where the model is presented with an input video with groundtruth "*Read brownie box*". It is interesting to note that the model is also able to infer semantic provenance for the concept "*brownie box*" through the presence of the multiple ungrounded context generators *container, food and recipe*. This highlights the advantage of using ConceptNet as a semantic knowledge source; concepts with a more abstract semantic connection can be inferred given the depth of common-sense knowledge encoded within the ConceptNet knowledge base.

3.3 Handling What-Ifs

Perhaps the most important aspect of explainability is a model's ability to handle "What-if" scenarios posed by the user. As the final decision maker, the human may have some insight that the model does not possess such as intuition and experience. The model must be able to handle such queries and justify its inference process based on its experience and the resulting knowledge. For example, while an interpretation made by the model may hold semantic meaning, the context may not be correct and hence a exchange of concepts is required for better performance. This requires the model to have a deep understanding of the domain concepts and their applicability in the current interpretation. This is perhaps where the advantage of using an external knowledge-base in ConceptNet and integrating it with the Pattern Theory formalism is best highlighted.

The process of establishing and quantifying semantic relationship relies on completeness of the ConceptNet framework and can be analyzed for extant semantic relationships in the knowledge-base to understand why certain concepts were or were not linked semantically in a given interpretation. The inherently explainable nature of the proposed approach allows the model to interact with the user to unravel the shortcomings of its current semantic knowledge-base. It also allows for the human to understand *why* certain semantics were ignored, either completely or in favor of others. It is important to note that such interactions can easily be extended into a form of active learning model that successfully transfers knowledge from the human user to its existing knowledge base.

3.3.1 Alternatives to Grounded Concept Generators

One of the proposed questions for explainability in Sect. 3 is designed to allow the model to walk through its knowledge base and inference process to reason possibility of the presence of a proposed concept generator in its final interpretation. The question "*Why not <concept> instead of <concept> in the interpretation?*" allows the user to understand why certain concepts were not chosen by the model in its inference process. The algorithm for generating explanations for a possible alternative to a particular grounded concept is shown in Algorithm 3.

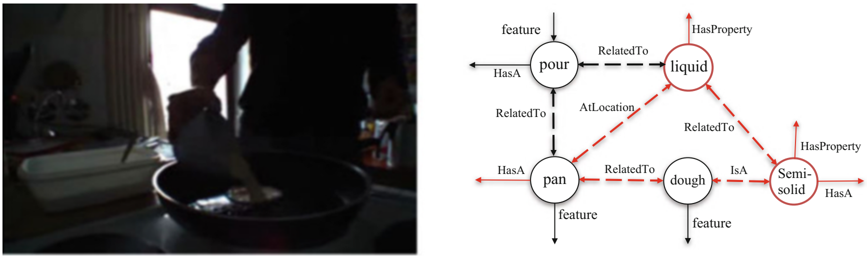
Algorithm 3: The algorithm for understanding the provenance of a concept. Each call to the function returns a tuple for data $E_{support}$ and semantics-based ($E_{semantic}$) explanations for the probability of an alternative generator (g_i') for a given configuration c, a grounded concept g_i and a knowledge-base C_N, which, in this case, is ConceptNet. The function call to $get\text{ -}semantics(g_i, C_N)$ returns a set of all semantic relationship for a concept g_i in the knowledge-base C_N

```
alternateConcept ($c, g_i, g_i', C_N$);
    $E_{support} \leftarrow \emptyset$
    $E_{semantic} \leftarrow \emptyset$
    $G' \leftarrow \emptyset$
    $B' \leftarrow \{\beta_j(g_i)\} \quad \forall \quad D(\beta_j(g_i)) = 1$
    for $\beta_j(g_i) \in B'$: do
        $G' \cup get\text{-}generators(\beta_j(g_i))$
    end
    for $(\beta_j(g_i), g_j) \in G'$: do
        if $g_j \in F$ then
            $E_{support} \cup (g_j, a(\beta'(g_i), \beta''(g_j)))$
        end
        else if $g_j \in G$ then
            $E_{support} \cup (get\text{ -}semantics(g_i', C_N) \cap get\text{ -}semantics(g_i, C_N))$

    end
    return $E_{support}, E_{semantic}$
```

Each call to the function returns a tuple for data-based probability and explanation $E_{support}$ and semantics-based explanation ($E_{semantic}$) for a given configuration c, a grounded concept g_i, a knowledge-base C_N and a possible alternate concept g_i'. We begin with the given concept generator. We traverse through the configuration based on the active, closed bonds present for the given generator. We compute the data-based explanation is constructed by getting the confidence scores from underlying machine learning classifiers with the corresponding feature generator. Finally, the semantics of the proposed concept and the reminder of the configuration, consisting of only the other grounded concept generators are analyzed using the $get\text{-}semantics(.)$ function. The $get\text{-}semantics(g_i, C_N)$ function returns a set of all concepts semantically related to the given concept g_i is returned. The semantic explanation for the presence of a particular concept within the configuration c can be obtained through recursive calls to the function for all grounded concept generators in the configuration. The resulting tuples are then fed onto a template-based sentence generator for dialogue-based user interaction.

One such explanation is shown in Fig. 5 where the prompt by the user posed an alternative concept *bun* in the place of the existing concept *dough*. The model was able to reason through the semantic relationships in ConceptNet and able to justify its choice due to the lack of semantic concepts that allowed for semantic relationships with the other grounded concept generator *pour*. Additionally, the



Input Video: *Pour dough into pan*

Output Interpretation: *Pour (liquid) (semisolid) dough into pan*

Q: Why not the concept *bun* instead of *dough* in the interpretation?
A: The concept generator *bun* can be explained through the feature generator *HOG1* with energy -0.26827. No semantic relationships could be established between concept generators *bun* and *pour*.

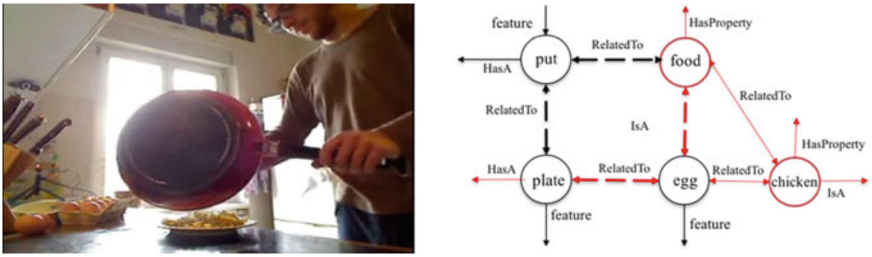
Fig. 5 An illustration of example interactions with the proposed model is shown when provided with the question *"Why not <concept> instead of <concept> in the interpretation?"* for the input video "*Pour dough into pan*". The model is able to reason through the proposed concept "*bun*" and infer why the concept was not chosen in its final interpretation

energy of the proposed configuration is also presented to the user which points to the reduced probability of presence of the grounded generator due to reduced confidence scores from its corresponding the feature generator *HOG1*.

3.3.2 Alternative Activity Interpretations

The question *"What are some alternatives to the interpretation?"* is one of the proposed questions for explainability in Sect. 3 that is designed to allow the model to walk through its inference process to provide alternative interpretations for a given video input. It is to be noted that the proposed approach using contextualization produces several, semantically coherent interpretations for a given video as its output. This is designed such that it takes multiple possible labels as input for each of the detected action and object feature generators. This is illustrated in Fig. 1, where the overall architecture is shown. Hence, there can exist multiple, plausible interpretations that are ranked based on their energy, given in Eq. (5) and hence their resulting probability (Eq. (4)). The probability and energy of a given configuration *c* are indicative of their semantic coherence and plausibility.

One such explanation is shown in Fig. 6 where the prompt by the user allowed the model to provide alternative interpretations to the input video "*Put egg on plate*". The original inference process (Sect. 2.5), produces plausible interpretations, which are stored in its internal model, are iterated through and the top interpretations are returned. In this case, the other plausible interpretations that were returned were "*put pancake on plate*", "*put from pan to plate*" and "*Pour eggs into bowl*". It is to be noted that these interpretations are driven primarily by the confidence score and



Input Video: *Put egg on plate*

Output Interpretation: *Put (chicken) egg (food) on plate*

Q: What are some alternatives to the interpretation?

A: Some alternatives are: (1) Put pancake on plate; (2) Put from pan to plate; (3) Pour eggs into bowl

Fig. 6 An illustration of example interactions with the proposed model is shown when provided with the question *"What are some alternatives to the interpretation?"*. The model is able to produce other plausible, semantically coherent alternative interpretations for a given video

concept proposals provided by the underlying concept detector models and further explanations for the provenance for plausible concepts can be derived by algorithms provided in prior sections.

3.3.3 Why Not a Given Interpretation?

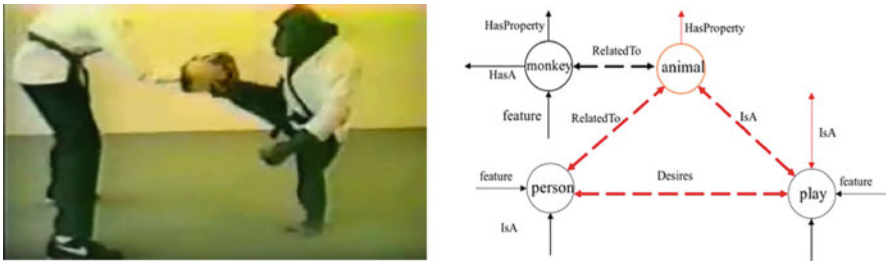
The final question in the proposed set is perhaps one of the most important one. It allows the model to express *why* it failed to arrive at the expected interpretation and hence allows the user to understand the point of failure. For example, the model might have not inferred the correct interpretation due to ambiguity in the underlying machine learning models for action and object label detection or there might be gaps in the knowledge base that did not allow for the correct inference. Such explanations are prompted by the question "*The correct interpretation is <interpretation>. Why did you not get there?*".

The response to this question are generated by using Algorithm 1 described in Sect. 3.1. However, the algorithm's input are given by constructing a new configuration by running through the inference process by combining the existing feature generators, the proposed grounded concept generators and the knowledge base in ConceptNet. This is illustrated in Algorithm 4, which returns a tuple for data-based provenance $E_{support}$ and semantics-based provenance ($E_{semantic}$) for the alternate configuration c'.

One such explanation is shown in Fig. 7 where the prompt by the user allows the model to provide justification for not inferring the correct interpretation to the input video "*Monkey fighting with a man*". It is interesting to note that the shortcomings of the ConceptNet knowledge-base in allowing the inference process to establish semantic relationships between the proposed concept generators {*person, monkey* and *fight*}.

Algorithm 4: The algorithm for understanding the reasoning behind the model's decision to ignore the correct interpretation. The function call to *construct -config*(.) runs the inference process again for a fixed set of grounded concept generators, feature generators and the knowledge-base C_N

```
alternateInterpretation $(c, c', C_N)$;
    $E_{support} \leftarrow \emptyset$
    $E_{semantic} \leftarrow \emptyset$
    $G' \leftarrow \{g_i \in c'\}$
    $F' \leftarrow \{g_{f_i} \in c'\}$
    $c_{alt} \leftarrow construct\text{-}config(F', G', C_N)$
    return $ExplainInterpretation(c_{alt})$
```



Input Video: *Monkey is karate kicking at a man's gloved hand*

Top Interpretation: *A monkey is playing with a person*

Q: **Correct interpretation is *Monkey fighting with person.* Why did you not get there?**

A: No semantic relationships could be established between concept generators *monkey, person* and *fight*. Hence, it has higher energy compared to "*Monkey is playing with person*" and resulting in lower probability of selection.

Fig. 7 An illustration of example interactions with the proposed model is shown when provided with the question *"The correct interpretation is <interpretation>. Why did you not get there?"*. The model is able to generate an explanation why the proposed interpretation was not arrived at for the given video "*Monkey is fighting with a man.*"

4 Conclusion and Future Work

In this paper, we explored the aspect of explainability in intelligent agents that generate activity interpretations through the inherently explainable nature of Grenander's pattern theory structures and contextualization cues constructed from ConceptNet. We demonstrated that, when combined, the proposed approach can be used to naturally capture the semantics and context in ConceptNet and infer rich interpretative structures. The inference process allows for multiple putative objects and action labels for each video event to overcome errors in classification. Along with a dialog system, this allows us to interact with the model to understand the rationale behind its inference process. We demonstrate that the proposed approach naturally captures the semantics in ConceptNet to infer rich interpretations. We have so far evaluated the outputs primarily on the Breakfast Actions dataset (Kuehne et al. 2014) which has over 5000 videos, but mostly qualitatively and visually, ourselves. We plan to conduct a structured study of the quality of the Q&A using human subjects. For further work, we look to expand the concept of contextualization to include temporal and spatial correlations for better visual understanding in videos that are longer in duration and with varying spatial constraints such as multiple video sources. Another possible direction is to build a life long learning system that can learn from its interactions from the user and update its knowledge base to adapt accordingly.

Acknowledgements This research was supported in part by NSF grants IIS 1217676 and CNS-1513126. The authors would also like to thank Daniel Sawyer for his invaluable insights during discussion.

References

Baehrens D, Schroeter T, Harmeling S, Kawanabe M, Hansen K, MÃžller KR (2010) How to explain individual classification decisions. Journal of Machine Learning Research 11(Jun):1803–1831

Biran O, McKeown K (2014) Justification narratives for individual classifications. In: Proceedings of the AutoML workshop at ICML, vol 2014

Caruana R, Lou Y, Gehrke J, Koch P, Sturm M, Elhadad N (2015) Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp 1721–1730

Core MG, Lane HC, Van Lent M, Gomboc D, Solomon S, Rosenberg M (2006) Building explainable artificial intelligence systems. In: AAAI, pp 1766–1773

Escalante HJ, Kaya H, Salah AA, Escalera S, Gucluturk Y, Guclu U, Baro X, Guyon I, Junior JJ, Madadi M, Ayache S, Viegas E, Gurpinar F, Sukma Wicaksana A, Liem CCS, van Gerven MAJ, van Lier R (2018) Explaining First Impressions: Modeling, Recognizing, and Explaining Apparent Personality from Videos. ArXiv e-prints 1802.00745

Grenander U (1996) Elements of pattern theory. JHU Press

Gumperz JJ (1992) Contextualization and understanding. Rethinking context: Language as an interactive phenomenon 11:229–252

Hendricks LA, Akata Z, Rohrbach M, Donahue J, Schiele B, Darrell T (2016) Generating visual explanations. In: European Conference on Computer Vision, Springer, pp 3–19

Herlocker JL, Konstan JA, Riedl J (2000) Explaining collaborative filtering recommendations. In: Proceedings of the 2000 ACM conference on Computer supported cooperative work, ACM, pp 241–250

Junior JCSJ, Musse SR, Jung CR (2010) Crowd analysis using computer vision techniques. IEEE Signal Processing Magazine 27(5):66–77

Kheradpisheh SR, Ghodrati M, Ganjtabesh M, Masquelier T (2016) Deep networks can resemble human feed-forward vision in invariant object recognition. Scientific Reports 6:32,672

Kuehne H, Arslan A, Serre T (2014) The language of actions: Recovering the syntax and semantics of goal-directed human activities. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 780–787

Lane HC, Core MG, Van Lent M, Solomon S, Gomboc D (2005) Explainable artificial intelligence for training and tutoring. Tech. rep., DTIC Document

Ledley S, Lusted LB, Ledley RS (1959) Reasoning foundations of medical diagnosis. In: Science, Citeseer

Linder N, Turkki R, Walliander M, Mårtensson A, Diwan V, Rahtu E, Pietikäinen M, Lundin M, Lundin J (2014) A malaria diagnostic tool based on computer vision screening and visualization of plasmodium falciparum candidate areas in digitized blood smears. PLoS One 9(8):e104,855

Liu H, Singh P (2004) Conceptnet' a practical commonsense reasoning tool-kit. BT Technology Journal 22(4):211–226

Lomas M, Chevalier R, Cross II EV, Garrett RC, Hoare J, Kopack M (2012) Explaining robot actions. In: Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction, ACM, pp 187–188

Mahadevan V, Li W, Bhalodia V, Vasconcelos N (2010) Anomaly detection in crowded scenes. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, pp 1975–1981

Martens D, Provost F (2013) Explaining data-driven document classifications. MIS Quarterly

Quionero-Candela J, Sugiyama M, Schwaighofer A, Lawrence ND (2009) Dataset shift in machine learning. The MIT Press

Ribeiro MT, Singh S, Guestrin C (2016) 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, ACM, pp 1135–1144

Shortliffe EH, Buchanan BG (1975) A model of inexact reasoning in medicine. Mathematical biosciences 23(3–4):351–379

Souza F, Sarkar S, Srivastava A, Su J (2015) Temporally coherent interpretations for long videos using pattern theory. In: CVPR, IEEE, pp 1229–1237

de Souza FD, Sarkar S, Srivastava A, Su J (2016) Spatially coherent interpretations of videos using pattern theory. International Journal on Computer Vision pp 1–21

Speer R, Havasi C (2013) Conceptnet 5: A large semantic network for relational knowledge. In: The People's Web Meets NLP, Springer, pp 161–176