OTTO-FRIEDRICH UNIVERSITY BAMBERG

DOCTORAL THESIS

Enhancing Explanatory Interactive Machine Learning – A Generalization of the CAIPI Algorithm

---

Author:
Emanuel SLANY

Supervisor:
Prof. Dr. Ute SCHMID

Committee:
Prof. Dr. Sven OVERHAGE (*head of committee*)
Prof. Dr. Ute SCHMID (*first examiner*)
Prof. Dr. Roman KLINGER (*second examiner*)

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Natural Sciences (Dr. rer. nat.)

at the

Faculty for Information Systems and Applied Computer Science
Otto-Friedrich University Bamberg

A thesis written in cooperation

with the

Research Group Comprehensible Artificial Intelligence
Fraunhofer Institute for Integrated Circuits IIS

Bamberg 2025

URN: urn:nbn:de:bvb:473-irb-1073760
DOI: https://doi.org/10.20378/irb-107376

Diese Arbeit hat der Fakultät Wirtschaftsinformatik und Angewandte Informatik der Otto-Friedrich-Universität Bamberg als Dissertation vorgelegen.

Erste Gutachterin: Prof. Dr. Ute Schmid
Zweiter Gutachter: Prof. Dr. Roman Klinger
Tag der mündlichen Prüfung: 14.02.2025

"Sag', wie weit noch? Wie weit
Muss man gehen, damit man alles erreicht?"

Montez, Casper – 7 Leben

Zusammenfassung

Bei statistischen Verfahren des Maschinellen Lernens besteht die Möglichkeit falscher Vorhersagen. Vor allem im Hinblick auf sensitive Modellentscheidungen ist es wichtig sicherzustellen, dass falsche Vorhersagen korrigiert werden können. Algorithmen der Erklärbaren Künstlichen Intelligenz ermöglichen es, falsche Entscheidungsgründe aufzuzeigen. Die Möglichkeit, falsche Entscheidungsmechanismen aufzudecken, verhindert diese jedoch nicht. Das Forschungsgebiet Erklärbares Interaktives Maschinelles Lernen gibt Forschenden die Möglichkeit, Erklärungen, die den Entscheidungsmechanismus repräsentieren, anzupassen. Die Algorithmen unterscheiden sich in ihrer Art, das menschliche Feedback in das Modell einzubetten. Einige Algorithmen nutzen Modell-Spezifika, wie zum Beispiel das Bestrafen von Gradienten in irrelevanten Regionen. Andere verfolgen eine modell-agnostische Herangehensweise, die die Korrelation von menschlichem Feedback zu Erklärungen und der Zielvariable als zusätzliche Trainingsinstanzen, sogenannte Gegenbeispiele, induzieren.

Der CAIPI-Algorithmus gilt als Begründer der zweiten Kategorie. CAIPI ist eine Art algorithmischer Baukasten, der Komponenten zur Vorhersage, Erklärungsgebung, Interaktion und Gegenbeispielgenerierung iterativ verbindet. Gegenbeispiele werden für Instanzen generiert, die auf Basis der falschen Gründe korrekt vorhergesagt wurden. Obwohl CAIPI ein modell-agnostischer Algorithmus ist, lässt er sich aufgrund seines größtenteils theoretischen Konzepts schwer auf andere Domänen abseits der Bild- und Textklassifikation übertragen. Die existierende Formalisierung von CAIPI erschwert zudem die Untersuchung theoretischer Fragestellungen wie der nach dem Einfluss von Gegenbeispielen auf den Optimierungsprozess.

Das primäre Ziel dieser Doktorarbeit ist die Verbesserung von CAIPI, einerseits durch eine mathematische Analyse der Frage, wie sich Gegenbeispiele auf den Optimierungsprozess auswirken, andererseits durch Abänderungen oder Erweiterungen des ursprünglichen Algorithmus. Jeder Beitrag dieser Arbeit wird entweder formal diskutiert oder experimentell evaluiert. Dies gelingt durch folgende Forschungsfragen, die die Forschungsziele in wissenschaftlich messbare Fragestellungen überführen:

R1 Welchen Einfluss haben Gegenbeispiele auf den Optimierungsprozess von maschinell gelernten Modellen?

R2 Welchen Einfluss haben Änderungen der Komponenten von CAIPI, im Spezifischen Modifikationen der Vorhersage-, Erklärungs-, Interaktions- und Gegenbeispielgenerierungs-Komponente, auf die Anwendungsbereiche von CAIPI und die damit einhergehende Qualität?

Die kombinierte Antwort beider Forschungsfragen leistet einen wichtigen Beitrag auf dem Forschungsgebiet Erklärbares Interaktives Maschinelles Lernen: Während der Einfluss von Gegenbeispielen formal analysiert wird, sind die theoretischen Ergebnisse Motivation für Modifikationen von CAIPI. Die modifizierten Algorithmen verbessern entweder CAIPI im Kontext der Bildklassifikation oder erweitern den Anwendungsbereich von CAIPI. Die wissenschaftlichen Beiträge dieser Arbeit können in zwei Punkten zusammengefasst werden:

1) CAIPI und alle anderen enthaltenen Methoden dieser Arbeit werden formal beschrieben. So kann der Einfluss von Gegenbeispielen auf den Optimierungsprozess systematisch analysiert werden. In diesem Zusammenhang beinhaltet diese Arbeit eine neue Methode, die mit Hilfe von Erklärbaren Interaktiven Maschinellen Lernen mit Gegenbeispielen Random Forests optimiert. Die theoretische Analyse zeigt, dass Gegenbeispiele die Wahrscheinlichkeit der Anpassung des Entscheidungsmechanismus erhöhen. Eine angepasste Entscheidungsgrenze von Modellen ist jedoch nicht gleichbedeutend mit Vorteilen in der Vorhersagequalität. Ergebnisse aus einer Simulationsstudie mit einem aus den theoretischen Ergebnissen abgeleiteten Gegenbeispiel-Filter zeigen, dass vorteilhafte von nicht vorteilhaften Gegenbeispielen unterschieden werden können. Dies gelingt dadurch, dass die vorhergesagte Instanz dahingehend überprüft wird, ob sie charakteristisch für Instanzen aus dem Datensatz ist. Ist dies nicht der Fall, handelt es sich um einen Ausreißer, für den keine Gegenbeispiele generiert werden.

2) Zusätzlich werden durch diese Doktorarbeit zahlreiche neue Verfahren basierend auf CAIPI entwickelt. Unter anderem verwenden vorgeschlagene Varianten von CAIPI generative Modelle, um Gegenbeispiele zu generieren. Dies steigert die Plausibilität von Gegenbeispielen. Diese Modifikation steigert sowohl die Performanz in der Bildklassifikation, ermöglicht ferner aber auch die Modelloptimierung für die Klassifikation tabellarischer Daten. Vor allem im neu erschlossenen Anwendungskontext Klassifikation von Tabellendaten werden durch Teile dieser Doktorarbeit bedeutende neue Methoden entwickelt: (i) FAIRCAIPI ist eine Variante von CAIPI, die die maschinell aus Daten gelernten Verzerrungen in Modellen abschwächt. Durch FAIRCAIPI werden maschinell gelernte Klassifikationsmodelle fairer. (ii) HYXIML kombiniert probabilistisch logische Inferenzen und statistische Vorhersagen aus maschinell gelernten Modellen, um Catastrophic Feedback Forgetting zu verhindern. Bei Catastrophic Feedback Forgetting handelt es sich um ein erstmals in dieser Arbeit beschriebenes Phänomen, bei dem die Wahrscheinlichkeitsmasse von Nutzerannotationen nicht ausreicht, um von optimierten Modellen berücksichtigt zu werden.

Darüber hinaus erlauben User Interfaces auch Personen ohne Expertise im Maschinellen Lernen, maschinell gelernte Modelle in einem erklärbaren und interaktiven Setting zu optimieren. Die User Interfaces wurden jedoch nicht systematisch evaluiert. Zudem beinhaltet diese Arbeit neue Algorithmen des Erklärbaren Maschinellen Lernens für metrische Zielvariablen und Clustering mit arbiträren Datentypen. Kombiniert mit den beschriebenen Interaktionsmechanismen können weitere Anwendungsbereiche für CAIPI erschlossen werden.

Diese Doktorarbeit zeichnet sich durch einen formalen und algorithmischen Fokus aus. Zahlreiche Varianten von CAIPI erschließen neue Anwendungsbereiche. Die starke Ähnlichkeit aller Algorithmen trägt zu einer Generalisierung von CAIPI bei. Zudem leistet die Untersuchung von induzierten Gegenbeispielen einen wichtigen Beitrag im Forschungsgebiet. Nicht detailliert thematisiert werden in dieser Doktorarbeit alternative Feedback-Induktionsverfahren wie zum Beispiel die inkrementelle Abänderung von Instanzgewichten. In diesem Zusammenhang könnte eine Generalisierung der mathematischen Ergebnisse die Arbeit aufwerten. Gemeinsam mit hybriden Verfahren im Bereich der Bildklassifikation, die Catastrophic Feedback Forgetting begegnen, sind dies wesentliche Anknüpfungspunkte für künftige wissenschaftliche Arbeiten.

Abstract

An inherent challenge of statistical machine learning models is the possibility of erroneous outcomes, which underscores the importance of corrigibility, particularly in sensitive domains. Advances in the field of explainable Artificial Intelligence have revealed that machine learning models might conduct correct decisions based on incorrect decision-making mechanisms. The possibility of detecting erroneous decision making does not prevent incorrect decision-making mechanisms, which is why the research field explanatory interactive machine learning has equipped researchers with the opportunity to revise explanations. Explanatory interactive machine learning procedures vary in their feedback injection mechanism: Some utilize model-specific internals, e.g., they penalize gradients in indecisive regions. Others are model-agnostic and induce additional training data, termed counterexamples, which only contain the relation of human explanation revisions and the target.

CAIPI, the origin of the latter category, is an algorithmic framework to optimize machine learning models that iteratively combines components to predict, explain, and interact with instances. It generates counterexamples in iterations with correct prediction and erroneous decision-making mechanism. CAIPI, despite being claimed to be model-agnostic, has been proposed as a theoretical concept, which does not transfer well to practical application scenarios beyond image and text classification. Moreover, its formalization requires refinement, particularly when evaluating the impact of counterexamples on the optimization framework.

This thesis aims to address CAIPI's limitations through a formal analysis of how counterexamples influence the iterative optimization of machine learning models, together with algorithmic modifications of the original CAIPI framework. Each contribution is formally assessed or experimentally evaluated. Consider the following two research questions that operationalize the research objectives:

R1 How do counterexamples affect the optimization of machine learning models?

R2 How do modifications in the prediction, explanation, interaction, and counterexample generation components affect CAIPI's applicability to machine learning tasks?

In combination, the answers to both research questions advance explanatory interactive machine learning by introducing theoretically motivated, adapted versions of CAIPI that enhance its performance in image classification or extend its range of applications. This thesis makes two key contributions:

1) This thesis formalizes CAIPI and each proposed method with mathematically grounded definitions, enabling a formal evaluation of how an iterative counterexample induction affects the machine learning optimization process. In this regard, a novel method is proposed to optimize random forests by adding counterexamples constructed to adjust the model's decision-making mechanism. Findings indicate that while counterexamples increase the probability of adjusting the random forest's decision boundary, the decision boundary modification and the predictive quality are decoupled, meaning that counterexamples do not constantly shift the decision-making mechanism into the beneficial

direction. A simulation study demonstrates that a proposed counterexample filtering step can distinguish between beneficial and non-beneficial counterexamples. The counterexample filter assesses the representativeness of instances and prevents the generation of counterexamples for outliers.

2) Additionally, this thesis proposes several adaptations of CAIPI, each modifying one or more of its components. For example, incorporating generative models for the generation of counterexamples increases the plausibility of counterexamples, enhancing CAIPI's performance in image classification and extending its applicability to tabular data classification – a novel domain for CAIPI. Especially for tabular data classification, two important CAIPI adaptations are proposed: (i) FAIRCAIPI, which optimizes bias detection metrics to improve the fairness of classification models that might reproduce data-inherent biases, and (ii) HYXIML, which accommodates probabilistic logic inferences and machine learning predictions to counteract catastrophic feedback forgetting. This vulnerability of explanatory interactive machine learning, discussed by this thesis, describes user annotations with insufficient probability mass, preventing user feedback from being reflected by the optimized model in the prediction phase.

Although not explicitly evaluated, this thesis proposes user interfaces for binary image and tabular data classification tasks. Finally, it derives and evaluates explanatory machine learning approaches combined with feedback injection mechanisms, which potentially expand CAIPI's application spectrum to numeric target variables and clustering with mixed data types.

This doctoral thesis is driven by a strong formal and algorithmic focus. It generalizes the CAIPI algorithm by transferring it to diverse machine learning tasks. The thorough derivation of CAIPI's components causes highly similar algorithms, despite being applied to various tasks and data types. Its formal focus reveals theoretical insights into model optimization with counterexamples, ultimately enhancing explanatory interactive machine learning as a research area. The thesis proposes mostly algorithms, which induce user feedback into the model by counterexamples, neglecting alternative feedback injection mechanisms, e.g., incremental weight adaptations, which is a promising first future research direction. Alternative feedback injection mechanisms would benefit from a formal refinement and generalization – a second research topic to explore. Finally, developing hybrid approaches against catastrophic feedback forgetting for image classification defines a potentially impactful future research avenue.

Acknowledgements

While in candidature for the doctoral degree, my position was funded by the German Federal Ministry of Education and Research. I worked on the project "Human-Centered Artificial Intelligence in the Chemical Industry" (short and German: hKI-Chemie, grant number: 01IS21023G).

While writing this thesis, I have always been looking forward to finally write the acknowledgements section to express my gratitude to the persons that have contributed to my academic success. Many persons mentioned in this section might not even know that I am thankful about how they influenced my academic career.

First of all, I want to thank my math teacher – I omit his name here –, who told my parents that I would be too stupid for math. A sincere "thank you". Knowing this has given me the drive to work hard, in particular when my predisposition might not be the best. In this regard, I also want to thank my parents, who told me that I should not care about other people's destructive opinions as long as I am humble and hardworking. Thank you also to my parents-in-law, who have taught me that it is okay to take risks if you follow your dreams.

The first academic thank you note is directed to Prof. Dr. Thomas Saalfeld, who gave me my first student assistant job during my BA studies in Political Science. This position has awakened my academic interest. An important person in this time was Carsten Schwemmer – in the meantime Prof. Dr. Carsten Schwemmer –, who has introduced me to the world of Python and recognized my potential in statistics and programming.

Carsten was the main reason why I approached a MSc degree in Survey Statistics. During my MSc studies, I was and today I still am especially looking up to two Survey Statistics chair members at the University of Bamberg: Dr. Florian Meinfelder has shown me that statistical programming can solve problems, which otherwise would be infeasible, and Prof. Dr. Christian Aßmann has taught me that computational methods – no matter how complex they are – can be expressed in mathematical language. Christian Aßmann also gave me the opportunity to write my MSc thesis in cooperation with the Fraunhofer Institute for Integrated Circuits IIS on the topic "Approximation Methods for Bayesian Deep Neural Networks in Image Recognition" – back then, an unusual thesis topic for Survey Statistics.

Simultaneously to starting my MSc studies, I started a student assistant job at Fraunhofer IIS, where I was responsible for software security and sales of the SHORE software. Thank you, Sabine Stigler, for hiring me without software development skills. Today, Sabine still has an open ear for everybody in the office, including me. I shared my first office in Tennenlohe with Ines Rieger, who supervised my MSc thesis together with Andreas Foltyn. Thank you, Ines and Andreas; you are true experts on your fields. Thank you for guiding me through my first baby steps in Applied Informatics.

After my MSc thesis, I was working for HUK-Coburg. I want to thank two persons who encouraged me to pursue a doctoral degree: Dr. Steve Grehl and Dr. Matthias Ring. In this context, I also want to thank Dr. Jens-Uwe Garbas and again Ines, who helped me with my way back into academia to Fraunhofer IIS.

Back at Fraunhofer, Stephan Gick and Dr. Volker Bruns both granted me the freedom to work on this thesis. I had many great student assistants: I am especially grateful for the theses of Yannik Ott, Louisa Heidrich, and Jonas Amling. All of which were the basis for publications. Not to forget Maren Stümke, who helped me with the literature review for this thesis. Over the years, I have worked with many researchers from academia and industry. Prof. Dr. Jan Paulus and Moritz Lang, I am glad that I had the opportunity to collaborate.

I was and still am part of the small project group "Comprehensible AI" led by Prof. Dr. Ute Schmid with her former deputy head, Dr. Stephan Scheele, now Prof. Dr. Stephan Scheele. Congratulations on this achievement, Stephan! Both are persons who have had a significant impact on my academic career. First, Prof. Dr. Ute Schmid who selected me as one of her doctoral candidates. Ute, it still is an honor. Ute has the impressive skill of distinguishing good from bad research practices in seconds. Even if your feedback immediately destroyed some of my ideas, I always left your office with new directions to explore. I frequently faced doubts that my topics were not important enough for the research community or that the research volume I produced was insufficient for a doctoral degree. You encouraged me to pursue my path. That is why I am thankful that you are my supervisor. Moreover, I am grateful for the opportunity to participate in the "Nordic Probabilistic AI" summer school in Trondheim, Norway, in June 2023, which has sharpened my research profile and influenced many parts of this thesis. This was the single most influential event during my doctoral studies. Second, I want to thank Prof. Dr. Stephan Scheele. In fact, if I only could thank one person, I would thank him. Stephan is a mentor for me. For countless hours, we discussed paper ideas and their mathematical formalization. Stephan has co-authored all of my publications and has extensively proofread this thesis. Your advice had an incredible impact on this thesis. Sometimes, I even wonder if I could have finished this thesis if I had not had you. Thank you, Stephan.

I had my colloquium in June 2024, where I first met Prof. Dr. Sven Overhage and Prof. Dr. Roman Klinger, the head of my doctoral committee and the co-assessor. There, I realized why professors are professors. A short presentation of my work was sufficient for them to pinpoint the weaknesses of my research. This feedback was precious and I have done my best to include it in this thesis. Thank you for this discussion, and thank you also for encouraging me to write a monograph. I enjoyed writing it.

It is important to me to thank Ute and Roman for providing feedback to this thesis before its submission. I am aware that this has been an unusual opportunity, yet it has let me appreciate your comments even more. I have included most of your suggestions, even if it meant to rewrite parts I considered final.

Having a child is sometimes incompatible with writing a doctoral thesis. Having a child like you, Frida, also exceeded the magnitude of love that I thought I could be feeling. You always show me what really is essential in life. Ina, I hope you know, you are the love of my life. Even if this section suggests that every stone of the path to the doctoral degree has magically fallen into its place, you know better than everybody else that there have been dark times for me – darker than just quitting this work. No words can express how thankful I am that you are on my side no matter what. I sincerely apologize for every situation where I have put my personal success over our marriage or our family. I love you, Ina and Frida.

Contents

Zusammenfassung **v**

Abstract **vii**

Acknowledgements **ix**

Contents **xi**

List of Figures **xiii**

List of Tables **xv**

List of Algorithms **xvii**

List of Abbreviations **xix**

List of Symbols **xxi**

1 Introduction **1**
1.1 CAIPI and its Limitations . . . 2
1.2 Objectives, Contributions, and Research Questions . . . 5
1.3 Related Work . . . 8
1.4 Synopsis . . . 12

2 Technical Foundations **17**
2.1 Derivation of CAIPI Components . . . 17
2.2 The Generalized CAIPI Algorithm . . . 21
2.3 Chapter Summary . . . 22

3 The Role of Counterexamples **23**
3.1 Extending the Mathematical Formalization . . . 25
3.1.1 Foundations of Random Forests . . . 25
3.1.2 Revisable Counterfactual Explanations . . . 27
3.1.3 Counterexample Generation by Subset Sampling . . . 30
3.2 Theoretical Implications . . . 32
3.3 Experimental Evidence for a Counterexample Filter . . . 33
3.4 Chapter Summary . . . 38

4 CAIPI Component Adaptations **41**
4.1 Tabular-data-specific Adaptations . . . 42
4.1.1 Counterexamples by Constrained Large Language Models . . . 42
4.1.2 Optimization for Fairness . . . 54
4.2 Image-specific Adaptations . . . 66
4.2.1 User Interaction and Data Augmentation . . . 66
4.2.2 Counterexamples by Variational Autoencoders . . . 74
4.3 Potential CAIPI Extensions . . . 85

4.3.1 Regression and Optimization . . . 85
4.3.2 Mixed-Data Clustering . . . 93
4.4 Chapter Summary . . . 103

5 Catastrophic Feedback Forgetting **105**
5.1 Hybrid Explanatory Interactive Machine Learning . . . 108
5.2 Experimental Evidence for a Hybrid Approach . . . 110
5.3 Chapter Summary . . . 113

6 Conclusion **117**
6.1 Summary . . . 117
6.2 Discussion . . . 118
6.3 Open Potentials . . . 120

A Mathematical Appendix **121**
A.1 Proof of Lemma Decision Trees and Decisive Features . . . 121
A.2 Proof of Proposition Decision Trees and Counterexamples . . . 121
A.3 Proof of Theorem Generalization Error and Counterexamples . . . 123

B Additional Results **125**

C Peer-Reviewed Publications **133**
C.1 Counterexamples from Constrained Large Language Models . . . 133
C.2 FairCAIPI . . . 133
C.3 CAIPI in Practice . . . 134
C.4 Bayesian CAIPI . . . 134
C.5 Cluster XAI . . . 135
C.6 Hybrid Explanatory Interactive Machine Learning . . . 135

D Scientific Activities **137**
D.1 Non-Peer-Reviewed Publications . . . 137
D.2 Talks . . . 137
D.3 Reviews . . . 137
D.4 Advised Thesis . . . 138
D.5 Proposals . . . 138

Bibliography **139**

List of Figures

1.1 Research area . . . 2
1.2 Original CAIPI algorithm . . . 4
1.3 Related work . . . 9
1.4 Visual synopsis . . . 13
1.5 CAIPI modifications per chapter . . . 15

3.1 Explanatory interactive machine learning on tabular data . . . 24
3.2 Counterfactual explanation . . . 28
3.3 User interface for tabular data . . . 30
3.4 Optimization outcome on synthetic Credit data . . . 36

4.1 LLMXIML overview . . . 43
4.2 LLMXIML component adaptations . . . 43
4.3 LLM interaction . . . 49
4.4 LLMXIML optimization on tabular data . . . 52
4.5 Gender distribution on Credit data set . . . 55
4.6 FAIRCAIPI overview . . . 56
4.7 FAIRCAIPI component adaptations . . . 56
4.8 Graphical representation of FAIRCAIPI . . . 60
4.9 Predictive performance during FAIRCAIPI optimization . . . 62
4.10 FAIRCAIPI optimization for fairness metrics . . . 63
4.11 Fairness of predictions and explanations in FAIRCAIPI . . . 63
4.12 Practical CAIPI component adaptations . . . 67
4.13 Data augmentation . . . 69
4.14 Image-specific user interface . . . 70
4.15 Problematic data augmentation . . . 75
4.16 Bayesian CAIPI component adaptations . . . 75
4.17 Variational Autoencoder . . . 77
4.18 User interface for Bayesian CAIPI . . . 80
4.19 Bayesian CAIPI predictive performance . . . 82
4.20 Bayesian CAIPI explanatory quality . . . 82
4.21 PHAL overview . . . 86
4.22 Explanatory interactive regression outline . . . 93
4.23 clusterExplainR overview . . . 94
4.24 clusterExplainR rule selection heuristic . . . 97
4.25 clusterExplainR rule-based cluster explanations . . . 99
4.26 clusterExplainR global feature importance score SHAP comparison . . . 100
4.27 Explanatory interactive clustering outline . . . 102
4.28 Summary of CAIPI adaptations . . . 102

5.1 HYXIML problem statement . . . 106
5.2 HYXIML overview . . . 107
5.3 HYXIML component adaptations . . . 107
5.4 HYXIML experimental setup . . . 111

5.5 ML predictions and logical inferences on Diabetes 112
5.6 HYXIML CAIPI comparison on Diabetes 112
5.7 Extension of CAIPI adaptations summary 114

B.1 Plausibility post-processing for counterfactual explanations 128

List of Tables

1.1 Explanatory interactive machine learning methods 11
1.2 Summary of contained publications 14

2.1 Outcome cases 20

3.1 Data-generating process 35
3.2 Baseline results on synthetic data 35
3.3 Experimental results on synthetic data 37

4.1 Baseline on tabular data 51
4.2 Ratio of valid counterexamples 51
4.3 LLMXIML results on tabular data 51
4.4 Bias detection metrics 57
4.5 FAIRCAIPI predictive performance results 62
4.6 FAIRCAIPI fairness results 63
4.7 CAIPI predictive performance 72
4.8 CAIPI explanatory performance 72
4.9 Bayesian CAIPI experimental results 82
4.10 PHAL experimental results 91
4.11 clusterExplainR rule fidelity evaluation 99
4.12 clusterExplainR rule complexity evaluation 100

5.1 ML predictions and logical inferences 112

B.1 Quality metrics for counterfactual explanations 129
B.2 Evaluation of plausibility post-processor 129

List of Algorithms

2.1 CAIPI 21

4.1 LLMXIML 48
4.2 FAIRCAIPI: GEN 59
4.3 FAIRCAIPI 60
4.4 BAYESIANCAIPI: GEN 79
4.5 BAYESIANCAIPI 80
4.6 PHAL 89
4.7 clusterExplainR: RULESEARCH 98

5.1 HYXIML: TRAIN 109
5.2 HYXIML: PREDICT 109

B.1 Plausibility of counterfactual explanations: POSTPROCESSOR 127

List of Abbreviations

ClAMP	**C**luster **A**nalysis with **M**ultidimensional **P**rototypes
CT	**C**omputer **T**omography
Corr. Expl.	occasionally: ratio of **Corr**ect **Expl**anations
Def.	occasionally: **Def**inition
DP	**D**eprived (unprivileged) group with **P**ositive (favorable) label
DN	**D**eprived (unprivileged) group with **N**egative (unfavorable) label
e.g.	**e**xempli **g**ratia
ELBO	**E**vidence **L**ower **BO**und
et al.	**et al**ia
FP	**F**avored (privileged) group with **P**ositive (favorable) label
FN	**F**avored (privileged) group with **N**egative (unfavorable) label
GPT-x	**G**enerative **P**re-trained **T**ransformer (- version number)
HyXiml	**HY**brid e**X**planatory **I**nteractive **M**achine **L**earning
KL	**K**ullback-**L**eibler
LLM	**L**arge **L**anguage **M**odel
LIME	**L**ocal **I**nterpretable **M**odel-agnostic **E**xplanations
LLMXiml	**L**arge **L**anguage **M**odel e**X**planatory **I**nteractive **M**achine **L**earning
ML	**M**achine **L**earning
NLP	**N**atural **L**anguage **P**rocessing
Phal	**P**ost-**H**oc model **A**pproximation with **L**ogic
SHAP	**SH**apley **A**dditive ex**P**lanations
Re.	occasionally: **Re**mark
RRR	**R**ight for the **R**ight **R**easons
RWR	**R**ight for the **W**rong **R**easons
W	**W**rong (for the wrong reasons)
wrt.	**w**ith **r**espect **t**o
XAI	e**X**plainable **A**rtificial **I**ntelligence
XIML	e**X**planatory **I**nteractive **M**achine **L**earning

List of Symbols

The symbols are chronologically listed wrt. the chapter or section of their first occurrence and the notational domain. This overview targets self-defined symbols rather than known mathematical operators. Note that symbols might be inconsistent between domains. The important goal is to preserve consistency within a notational domain. Occasionally, notation will be shared between domains. Then, inter-domain consistency is ensured. This will be made clear from context.

Tabular Data Classification

Chapter 2

$(x_1, ..., x_i, ..., x_n)^T \in \mathcal{X}$	feature
$\mathcal{F} = \{1, ..., i, ..., n\}$	identifier set
$v \subseteq \mathcal{F}, \bar{v} = \mathcal{F} \setminus v$	subset of (in)decisive feature identifiers
$\mathcal{Y} = \{0, 1\}$	target
$(x^{(n)}, y^{(n)}) \in \mathcal{X} \times \mathcal{Y}$	n-th feature target instance
$f : \mathcal{X} \to \mathcal{Y}$	binary classification model
$y = f(x)$	inference
$l : \mathcal{X} \to \mathcal{Y}$	binary labeling function
$(x_{\mathcal{L}}^{(n)}, y_{\mathcal{L}}^{(n)}) \in \mathcal{L} \subseteq \mathcal{X} \times \mathcal{Y}$	n-th labeled instance
$x_{\mathcal{U}}^{(n)} \in \mathcal{U} \subseteq \mathcal{X}$	n-th unlabeled instance
$(x', y') \in \mathcal{X}' \times \mathcal{Y}'$	counterexample
c	number of counterexamples
m	most-informative instance
α	prediction or split threshold

Chapter 3

$\mathcal{X}_v, \mathcal{X}_{\bar{v}}$	subset of (in)decisive features
$y = h(x, \Theta_d)$ with $d \in \{1, ..., D\}$	d-th decision tree inference
$(i, \alpha) = \theta \in \Theta$	decision tree split parameters
$\mathbf{y}$ with $\lvert\mathbf{y}\rvert = D$	set of decision tree inferences
$\mathbf{Y}$	decision tree inference sets of all instances
$\mathcal{Q} \subseteq \mathcal{X} \times \mathcal{Y}$	partition
$imp(...)$	impurity function
$gini(...)$	Gini index
$\mathcal{I}_{[...]}$	indicator function
$Pr(...)$	probability of random sequence
$mr(...)$	margin function
PE^*	upper bound of generalization error
$\bar{\rho}$	inter-tree correlation
str	strength
$p(...)$	probability distribution
$(\bar{x}, \bar{y}) \in \bar{\mathcal{X}} \times \bar{\mathcal{Y}}$	counterfactual instance
$dist(...)$	distance function
$div(...)$	diversity function

$U((a,b)^n)$	n-dimensional uniform distribution
a, b	scalars for lower and upper bound
$u \subseteq \mathcal{F}$	subset of altered feature identifiers
$q_{x_i}(...)$	quartile value wrt. i-th feature
λ	step size
x^*	emphasizing of an instance
$\mathcal{B} = \{B^{(1)}, ..., B^{(j)}, ..., B^{(k)}\} \subseteq \mathcal{X}$	partitioning / clustering
$\{z^{(1)}, ..., z^{(j)}, ..., z^{(k)}\} = \mathcal{Z}$	cluster prototypes
$B^* \in \mathcal{B}^*, z^* \in \mathcal{Z}^*$	optimal clusters and prototypes
$\|\|...\|\|_2$	l2-norm
δ	distance threshold
w	scalar
tn, fn, tp, fp	true/false negatives, true/false positives

Section 4.1.1

$r = p::H:\text{-}B \in R$	probabilistic logic rule
p	probability scalar
$H = \phi, B = b_1, ..., b_n$	rule head (target) and body
$Pr_S(q\|R)$	success probability
L_R	ground facts in R
$e \in E$	probabilistic example
$t(...)$	translation function
$\mathcal{X}^* \times \mathcal{Y}^* \subseteq \mathcal{X} \times \mathcal{Y}$	subset of valid instances
x_{spurious}	spurious correlation feature

Section 4.1.2

S	identifier of protected attribute
$s, \bar{s}$	(un)privileged group
$d, \bar{d}$	(un)favorable label
$\tilde{z} \subseteq \tilde{x}$ with $\|\tilde{z}\| = M$	subset of simplified input
$g(...)$	approximation function
$\overrightarrow{h}(...)$	transformation function
$\overleftarrow{h}(...)$	reverse transformation function
$\bar{\psi} = (\psi_1, ..., \psi_n)^T$	attribution vector
exp	SHAP explanation
β	attribution threshold

Chapter 5

$\mathcal{M} \subseteq \mathcal{F}, \mathcal{C} = \mathcal{F} \setminus \mathcal{M}$	metric and categorical identifiers
x^{other}	other instance than x

Appendix B

$\mathcal{X}_{Exp} \subseteq \mathcal{X}$	explanation subset
$\gamma \in \Gamma$	plausibility constraint
$r_\Gamma \in R_\Gamma$	constrained rule

Image Classification

Section 4.2.1

$X \in \mathcal{X} \subseteq \mathbb{R}^{W \times H}$	feature with dimensions
$\mathcal{F} = \{1, ..., i, ..., WH\}$	identifier set
$\mathcal{Y} = \{0, 1\}$	target
$(X^{(n)}, y^{(n)}) \in \mathcal{X} \times \mathcal{Y}$	n-th feature target instance

$f : \mathcal{X} \to \mathcal{Y}$	binary classification model
$y = f(X)$	inference
$l : \mathcal{X} \to \mathcal{Y}$	binary labeling function
$(X_{\mathcal{L}}^{(n)}, y_{\mathcal{L}}^{(n)}) \in \mathcal{L} \subseteq \mathcal{X} \times \mathcal{Y}$	n-th labeled instance
$X_{\mathcal{U}}^{(n)} \in \mathcal{U} \subseteq \mathcal{X}$	n-th unlabeled instance
$D \in \{0,1\}^{W \times H}$	(in)decisive features
$\xi(...)$	LIME loss function
$\tilde{Z} \sim \tilde{X} \in \tilde{\mathcal{Z}}$	sample of simplified input
$g(...)$	approximation function
$\overrightarrow{h}(...)$	transformation function
$\overleftarrow{h}(...)$	reverse transformation function
$\tau(...)$	data augmentation
$(X', y') \in \mathcal{X}' \times \mathcal{Y}'$	counterexample
c	number of counterexamples
$IoU(...)$	intersection over union

Section 4.2.2

m	most-informative instance
α	prediction threshold
IG_i	Integrated Gradient at pixel i
X_B	baseline image
$k \in K$	Riemman summation approximation step
M	explanation mask
β	attribution threshold
$vae : \mathcal{X} \to \hat{\mathcal{X}}$	Variational Autoencoder
$enc : \mathcal{X} \to \mathcal{Z}$	encoder
$dec : \mathcal{Z} \to \hat{\mathcal{X}}$	decoder
$\hat{\mathcal{X}} = \{\hat{X}^{(1)}, ..., \hat{X}^{(n)}\}$	reconstructed images
$\mathcal{Z} = \{z^{(1)}, ..., z^{(n)}\}$	latent space representations
$p(x)$	marginal distribution
$p(z)$	prior distribution
$p(X\|z, \theta)$	likelihood distribution
$p(z\|X)$	posterior distribution
$q_\theta(z\|X)$	approximated posterior distribution
θ	parameters of vae
KL [...]	Kullback-Leibler divergence
ELBO(...)	evidence lower bound
$f' : \mathcal{Z} \to \mathcal{Y}$	classification model from latent space
$y = f'(z)$	inference from latent space
$z_{\mathcal{L}} \in \mathcal{Z}_{\mathcal{L}}, z_{\mathcal{U}} \in \mathcal{Z}_{\mathcal{U}}$	(un)labeled latent space encodings
$f'(enc(...))$ and $f' \circ enc$	nested notation

Tabular Data Regression

Section 4.3.1

$(x_1, ..., x_i, ..., x_n)^T \in \mathcal{X}$	feature
$\mathcal{F} = \{1, ..., i, ..., n\}$	identifier set
$y \in \mathcal{Y} \subseteq \mathbb{R}$	target
$f : \mathcal{X} \to \mathcal{Y}$	regression model
$f_*\|\mathcal{X}, \mathcal{Y}, x$	predictive distribution
$y = (y_{mean}, y_{cov}) = f(x)$	inference

$\mathcal{X}_{Exp} \subseteq \mathcal{X}$	explanation subset
$y_{prob} = inv\_cv(y)$	inverse coefficient of variation
$\overrightarrow{h}(...)$	transformation function
$\overleftarrow{h}(...)$	reverse transformation function
$\tilde{x}, \tilde{y}$	abstracted feature and target
p	probability scalar
$\phi \in \Phi$	target predicate
$R \in \mathcal{R}$	rule set from set of rule sets
$t(...)$	translation function
$e \in E$	probabilistic example
$Pr_S(q\|R)$	success probability
α	prediction threshold
$S(\mathcal{P}^{(t)}, \mathcal{P}^{(t')})$	stability between to sets of predicates

Mixed-Data Clustering

Section 4.3.2

$(x_1, ..., x_i, ..., x_n)^T \in \mathcal{X}$	feature
$\mathcal{F} = \{1, ..., i, ..., n\}$	identifier set
$\mathcal{M} \subseteq \mathcal{F}, \mathcal{C} = \mathcal{F} \setminus \mathcal{M}$	metric and categorical identifiers
$\mathcal{B} = \{B^{(1)}, ..., B^{(k)}\} \subseteq \mathcal{X}$	partitioning / clustering
$\mathbf{X} = (X_1, ..., X_i, ..., X_n)$	sequence of random processes
$z \in Z$	outcome from possible outcome set
$p = Pr(...)$	probability scalar from random sequence
$p(...)$	probability distribution
$H(...)$	entropy
$lFIS(...)$	local feature importance score
$gFIS(...)$	global feature importance score
$ECMS(...)$	entity cluster matching score
$m(...)$	matching coefficient
$r \in R$	conjunctive rule from set of rules
$r(...)$	application of rule
F1(...)	calculation of F1-score

Dedicated to the ones who always believed in me.
Dedicated to Frida and Ina Marie.

Chapter 1

Introduction

Data-driven, statistical machine learning (ML) models aid humans in many domains, either by entirely automating processes, or by providing auxiliary information learned from data: Examples include the automatic detection of defective components for industrial quality control (Müller et al., 2022), ML-aided parameter estimation for complex engineering mechanisms (Wirth, Schmid, and Voget, 2022), or the identification of cancer tissue to support physicians (Kourou et al., 2015). All of the prior use cases have in common that they require a high degree of expert knowledge. Naturally, extraordinary high expertise is expected to be rare throughout domains, putting ML models into the position that they may outperform the majority of users, while domain experts are still superior in their respective task.

This circumstance can lead to the chance that domain experts induce their knowledge into ML models, which are then exploited to guide and potentially educate novices (Wirth, Schmid, and Voget, 2022). Furthermore, the former use cases might evolve: For instance, installing new production facilities might cause novel types of damages, mechanical instruments might behave differently when transferred into regions with different humidity or temperature, or novel types of cancer might occur. In all examples, ML models improve if they are re-trained, which requires accumulated expert advice over time.

Crucial for the adaptation of ML models is their trustworthiness, as domain experts need to be certain that the ML model's decision boundary reflects the correct decision-making mechanism (Thaler and Schmid, 2021; Wirth, Schmid, and Voget, 2022), which can be revealed by explanatory ML (or more broadly Explanatory Artificial Intelligence (XAI)) techniques (e.g., Schwalbe and Finzel, 2023, and references therein). At this point, this thesis restricts itself to *local* explanatory ML approaches that illustrate the reasons for a specific prediction given a specific instance.

Explanatory Interactive Machine Learning (XIML) (Teso and Kersting, 2019) enables users to iteratively revise local explanations of ML models to optimize ML models and refine their decision-making mechanism. Specifically relevant to the understanding of XIML for this thesis is the *iterative interactivity on explanations* with a *model optimization objective*. Interacting with explanations of ML models has been shown to strengthen the users' trust in ML models (Teso and Kersting, 2019). Hence, XIML is suitable for both a continuous human knowledge induction into ML models and their trustworthy application.

Broadly, there exist two categories of XIML methods: Model-specific algorithms are either tailored to a specific ML model type, such as the interactive loss function adaptation in the deep learning context (Schramowski et al., 2020), or entirely novel algorithms like interactive self-explainable neural networks (Teso, 2019). Model-agnostic methods are mostly variations of the CAIPI algorithm (Teso and Kersting,

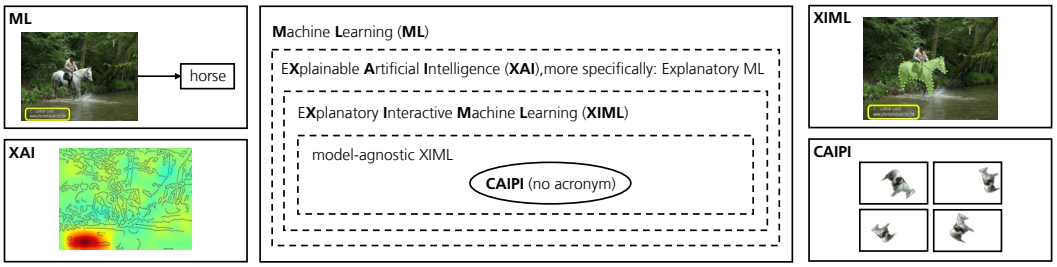

Figure 1.1: Research area. This figure situates CAIPI in the research area using *Clever-Hans predictions* (Lapuschkin et al., 2019). CAIPI overweighs feedback by counterexamples.

2019)[1]. CAIPI enables human users to refine the decision-making mechanism of ML models by an iterative interaction with their local explanations. In iterations, where the prediction is correct but the explanation reveals an incorrect perception of the decision boundary, CAIPI generates counterexamples – novel training instances that enforce the relation between decisive features and the target. Counterexamples move the decision boundary into the presumably correct direction.

The overarching objective of this thesis is an algorithmic enhancement of CAIPI that ultimately leads to an improved applicability of CAIPI across diverse ML tasks. The remainder of this chapter is organized into four sections: Section 1.1 will review the original formalization and evaluation of CAIPI (Teso and Kersting, 2019) to identify its major limitations. Section 1.2 will translate CAIPI's limitations into the research objectives of this thesis and list the contributions of this work. It will conclude with the formulation of research questions to support the contributions by scientific evidence. The XIML research area, in general, and CAIPI as a specific representative and central element of this thesis will be set into the broader context of related publications in Section 1.3. Finally, Section 1.4 will outline the structure of this thesis, emphasizing the complementary role of each chapter. This thesis consolidates and expands on the results of publications by the author during his doctoral candidature. These will be summarized in Section 1.4, which will also describe how the publications affect certain parts of this thesis.

1.1 CAIPI and its Limitations

This section contains two paragraphs: The first one locates CAIPI in the ML research domain and briefly discusses the formalization and evaluation strategy of the original publication (Teso and Kersting, 2019). The algorithmic discussion will be more technical than ordinary for introductions. However, the discussion will be as shallow as possible to spare out non-essential technical details. The second paragraph lists the major limitations of CAIPI, which this thesis aims to overcome.

CAIPI algorithm Figure 1.1 situates CAIPI in the research area (center) using *Clever-Hans predictions* (Lapuschkin et al., 2019) on the Pascal VOC data set[2], where the class label *horse* correlates with a watermark. Because traditional ML has, depending on the specific algorithm, a more or less opaque input output relation between the image and the class label (top left), local XAI techniques

[1] CAIPI is not an acronym. It inherits its name from its underlying local explanation procedure Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro, Singh, and Guestrin, 2016). The authors write: "CAIPIrinhas are made out of LIMEs." (Teso and Kersting, 2019)

[2] `http://host.robots.ox.ac.uk/pascal/VOC/voc2007/`, 21 May 2024.

highlight the decisive features for the model's decision – in this case, the watermark (bottom left). XIML enables users to revise the explanations, e.g., by annotating the supposably decisive features – here, the horse itself (top right). This figure leaves the translation from human revision to feedback induction into the model unspecified, as algorithms belonging to the XIML category follow heterogeneous strategies. CAIPI is an explain-to-revise loop using counterexamples, e.g., augmented cropped decisive features (bottom right). CAIPI is also the most prominent representative of the category model-agnostic XIML algorithms (Section 1.3).

CAIPI (Teso and Kersting, 2019) iteratively aligns components (i) to train a ML model, (ii) to select the most-informative instance, (iii) to obtain a ML prediction (iv) as well as a corresponding explanation, (v) and to generate counterexamples based on a human explanation revision (Figure 1.2, left). In each optimization iteration, a pre-trained ML model (line 1) selects the most-informative instance from an unlabeled data set (line 3). If a human user evaluates the prediction of the most-informative instance (line 4) to be correct, a local explanation depicts the reasons leading to the prediction (line 5). CAIPI uses Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro, Singh, and Guestrin, 2016) as a local explainer. LIME is a two-stage algorithm, where the instance is first segmented and the relevance for the segments is determined by a subsequent model. If both prediction and decision-making mechanism are correct (line 6) – the prediction is **R**ight for the **R**ight **R**easons (**RRR**) –, the most-informative instance is added to the labeled data. If the prediction is **W**rong (for the Wrong Reasons) (**W**), the labeled data set is appended by the corrected most-informative instance. If, however, the prediction is correct but not the reasons revealed by the explanation – the prediction is **R**ight for the **W**rong **R**easons (**RWR**) –, a human user revises the explanation (line 7) and the generated counterexamples overweigh the human feedback (lines 8 and 9). Finally, the most-informative instance is removed from the unlabeled data set (line 10) before the model is retrained (line 11) to prepare the subsequent optimization iterations.

While CAIPI's original local explanation method – LIME – in a more recent article has been associated with faithfulness drawbacks, as the authors have found a high variance among the explanations generated after the application of different image segmentation algorithms (Schallner et al., 2019), the algorithm has also been shown to be an inferior choice on tabular data (Lundberg and Lee, 2017). Both findings hint that LIME restricts the applicability of CAIPI, as the CAIPI framework is built around an explanatory ML technique, which is beneficial for some data types, mostly images and text. Yet, CAIPI's model invariance property also involves the local explanation component. In practice, each component of CAIPI can be modified or substituted to satisfy specific use cases or data type demands.

Figure 1.2 (right) proposes an abstracted conceptual CAIPI overview: The model component unifies the pre- and re-training as well as the most-informative instance selection procedure. The interaction component illustrates that human annotation is required for the prediction and the explanation. In the **RWR** case, the human feedback serves as input for the counterexample generation procedure.

Figure 1.2 (center) illustrates the basic idea behind CAIPI's original experimental setup for the image classification task (Teso and Kersting, 2019). The authors have induced decoy pixels into the FashionMNIST data set[3] such that the pixel color corresponds to a class label. The optimization objective is to unlearn the spurious correlation caused by the decoy pixels. LIME highlights the decisive regions. A prediction is categorized as **RWR** if the decoy pixels are the reasons that the classifier

[3] `https://github.com/zalandoresearch/fashion-mnist`, 22 May 2024.

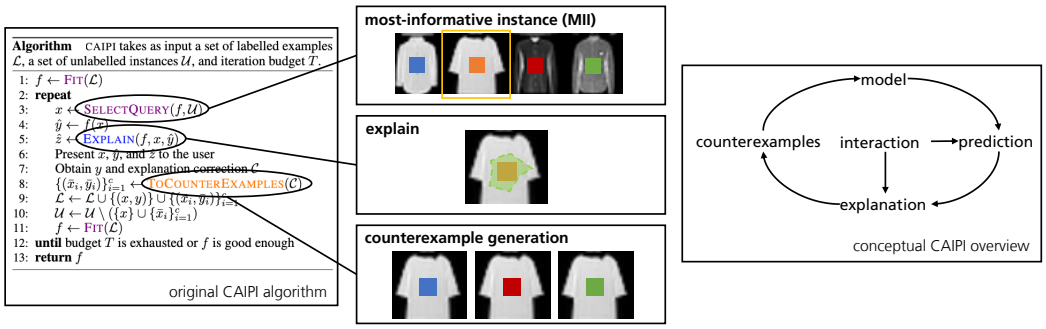

Figure 1.2: Original CAIPI algorithm (Teso and Kersting, 2019).

obtains the correct class label. In such cases, counterexamples mitigate the influence of the spurious correlation. Counterexamples are identical images but with randomized decoy pixel color. The authors compare the effects of inducing $\{0, 1, 3, 5\}$ counterexamples, where zero counterexamples equals coactive learning (Shivaswamy and Joachims, 2015). They infer an improved predictive performance on test data without decoy pixels for five counterexamples per **RWR** iteration.

Teso and Kersting (2019) also provide experimental evidence for text data classification, yet with fewer reimplementation details. Their GitHub repository[4] also contains some implementations for CAIPI and tabular data, which, however, is restricted in the sense that it is constructed to classify grid patterns. Therefore, this thesis will refer to image classification as CAIPI's primary application domain and supposes the application of CAIPI on tabular data as unsolved. This thesis explicitly excludes text classification with CAIPI, which has been part of Sebastian Kiefer's dissertation (Kiefer, 2023). In summary, this thesis particularly targets CAIPI variations in the image and tabular data classification domains.

Limitations Various limitations directly follow from the formalization and the experimental setup of the original CAIPI algorithm (Teso and Kersting, 2019):

1) CAIPI is formulated and evaluated as a theoretical concept, which cannot be put into practice without further extensions such as user interfaces.

2) Even if CAIPI is extended such that human users can operate it, the selection of components restricts CAIPI to image and text classification tasks.

3) The formalization of CAIPI is incomplete, as it contains no outcome case distinction. Nor are the procedures built on formal definitions,

4) which causes the inability to target theoretical research questions.

Limitations 1) and 2) restrict CAIPI's practical applicability. Limitations 3) and 4) confine the possibility of deriving theoretical implications from CAIPI.

The next section will summarize the scientific contributions of this thesis, beginning with the translation of CAIPI's limitations into specific research objectives from which key contributions will emerge. Each contribution is rigorously evaluated at a formal or experimental level. Accordingly, the section will conclude with the research questions that structure this thesis.

[4] `https://github.com/stefanoteso/caipi/blob/master/caipi/`, 14 August 2024.

1.2 Objectives, Contributions, and Research Questions

The previous section has outlined CAIPI and identified its algorithmic and experimental limitations. This section shifts focus to discuss the primary research objectives of this work and highlights its scientific contributions. To substantiate these contributions with formal, mathematical, or experimental evidence, this thesis reframes them as specific research questions.

Objectives To overcome the preliminary formulated limitations, this thesis will enable users to train ML models for image and tabular data classification with CAIPI. It will modify CAIPI's components to utilize CAIPI for the priorly mentioned ML tasks while increasing CAIPI's predictive and explanatory performance. The thesis will derive a unified algorithmic formalization for CAIPI with a minimal divergence between ML tasks and mathematically defined procedures to formally explore how counterexamples affect the CAIPI optimization cycle.

Contributions The research objectives can be condensed into two key contributions, which emanate from the specific contributions from each chapter or section:

1) This thesis contributes a unified, mathematically grounded algorithmic notation for CAIPI (Chapter 2) that is exploited for a theoretical discussion on the effects of counterexamples in model-agnostic XIML (Chapter 3):

   - In this regard, this thesis mathematically derives a novel XIML method for tabular data. It is built upon a random forest (Breiman, 2001) and counterfactual explanations (Wachter, Mittelstadt, and Russell, 2017) and generates counterexamples by subset sampling from clusters obtained by k-means (Lloyd, 1982) (Section 3.1).

   - This thesis proves that an induction of counterexamples into the derived setting is not necessarily beneficial for the predictive performance of the ML model nor for the model's ability to follow the correct decision-making mechanism (Section 3.2).

   - Moreover, it proposes a filtering step to distinguish beneficial from non-beneficial counterexamples and demonstrates its viability in a simulation study (Section 3.3).

2) Furthermore, this thesis contributes CAIPI variations in the sense that either the original performance for image classification is improved or novel application areas are disclosed. This thesis will have its focus on image and tabular data classification, where it will also contribute user interfaces. Nevertheless, this thesis will provide suggestions on how to expand CAIPI to mixed-data clustering, regression, and optimization.

   - Specifically, this thesis proposes a CAIPI variant for tabular data that generates counterexamples using Large Language Models (LLMs) (e.g., Brown et al., 2020). The validity of counterexamples is preserved by probabilistic logic constraints (Raedt, Kimmig, and Toivonen, 2007). Using LLMs iteratively as a semantic translator between probabilistic logic predicates and natural language, LLMs report validity violations to users, users can refine the validity constraints, and the LLM can revise its own generation instruction (Section 4.1.1).

- This thesis utilizes CAIPI as a bias mitigation in-processing method for tabular data. The evaluation demonstrates superiority to a state-of-the-art bias mitigation method (Kamiran and Calders, 2011) (Section 4.1.2).
- On image data, this thesis overcomes the predominantly theoretical evaluation concept of the original CAIPI publication (Teso and Kersting, 2019). It proposes a user interface and a counterexample generation procedure based on data augmentation, allowing human users to optimize ML models for binary image classification tasks (Section 4.2.1).
- Furthermore, this thesis identifies plausibility issues for image counterexamples generated by data augmentation. It proposes a novel counterexample generation procedure leveraging Variational Autoencoders (Kingma and Ba, 2015) and infers an improved representation of the model's decision boundary (Section 4.2.2).
- Targeting CAIPI's applicability beyond tabular data and image classification, this thesis contributes a probabilistic logic surrogate model (Raedt et al., 2015) for regression and optimization tasks (Section 4.3.1) as well as an entropy-based model-agnostic explanatory ML method for clustering (Section 4.3.2). Both sections contain XIML architectures that incorporate the derived approaches.
- Finally, motivated by Contribution 1), Chapter 5 describes the propensity that user feedback during the XIML optimization cycle is not reflected in subsequent optimization iterations, which this thesis terms catastrophic feedback forgetting. This thesis combines ML predictions and probabilistic logic inferences (Raedt et al., 2015) to a hybrid XIML approach, which is shown to counteract catastrophic feedback forgetting more effectively compared to CAIPI.

Research Questions Contributions 1) and 2) can be seen as this thesis's formal and algorithmic contribution. Both map into two distinct research questions to underpin the contributions by either mostly formal or experimental evidence. On a formal level, this thesis asks the following research question:

R1 How do counterexamples affect the optimization of ML models?

To assess **R1**, Chapter 2 will formalize the components of CAIPI (Teso and Kersting, 2019) transferred to a tabular data classification scenario. Chapter 3 will derive a XIML method to optimize a random forest (Breiman, 2001). Despite the XIML framework itself will still be model-agnostic, the formal insights into the optimization process with counterexamples will be centered explicitly around random forests. Random forests are comparatively simple to derive, yet are shown to have a relatively high predictive ability, especially on tabular data (Breiman, 2001). Hence, random forests are an appropriate choice for the first mathematical investigation of a XIML optimization setting with counterexamples. The subsequent chapters will account for the specificity of the results of Chapter 3, but still exploit the theoretical insights for their conducted CAIPI component variations.

Whereas research question **R1** will be addressed formally specifically for random forests and will only be supplemented by a simulation study, **R2** will be answered by experimental evidence. This thesis asks the following algorithmic research question:

R2 How do modifications in the prediction, explanation, interaction, and counterexample generation components affect CAIPI 's applicability to ML tasks?

Hereby, *modifications* refer to the exchange of either a single or multiple components of CAIPI, e.g., a replacement of the local explanation method. *Applicability* will be operationalized by predictive and explanatory performance metrics. Occasionally, other metrics will be used, such as bias assessments in the fairness context.

The goal of the first research question is to formalize CAIPI on a mathematical basis to foster a general theoretical understanding of the effects of counterexamples in ML training. The second research question will be applied to multiple domains. The goal is to either improve CAIPI for its original primary proposed objective – image classification – or to expand its application area. Research question **R2** aggregates modifications of each CAIPI component and targets multiple ML settings. Therefore, **R2** will be specified in the subsequent sections – specifically:

Section 4.1.1

R2.1 Does reasoning enhance the correctness of counterexamples generated by LLMs?
R2.2 How do LLM-generated counterexamples with and without reasoning affect the predictive and explanatory quality of XIML optimization?

Section 4.1.2

R2.3 Does the correction of explanations for fairness lead to fairer models?
R2.4 Does correcting explanations for fairness lead to fairer explanations?
R2.5 Does correcting for fair explanations have a negative impact on the predictive performance of the model?
R2.6 Which is superior, FAIRCAIPI (a CAIPI variant optimizing for fairness metrics) or the state-of-the-art Reweighing strategy?

Section 4.2.1

R2.7 Do explanation revisions improve the predictive quality?
R2.8 Do explanation revisions lead to an improved explanatory quality?
R2.9 Does the predictive and explanatory quality benefit from explanation revisions for wrong predictions?
R2.10 Which is superior, CAIPI or default deep learning?

Section 4.2.2

R2.11 Do counterexamples improve the model's predictive quality?
R2.12 Do counterexamples improve the model's ability to follow the correct decision-making mechanism?
R2.13 Which is superior, Bayesian CAIPI, CAIPI, or default deep learning?

Chapter 5

R2.14 Which is superior for unlearning a spurious correlation, HYXIML (Hybrid Explanatory Interactive Machine Learning) or CAIPI?
R2.15 Does HYXIML compromise the predictive performance?

Specific variations of **R2** ensure accurate assessments of specific CAIPI modifications. The refined applicability research questions will be answered in the respective sections. These will be aggregated into a generalized answer for **R2** in Chapter 5. Although this thesis proposes user interfaces, which increase CAIPI's applicability in the sense that they enable users without ML expertise to operate CAIPI, the user interfaces are not systematically evaluated. The assessment of human interaction, e.g., by user studies, is not subject of this thesis and excluded from **R2**.

After the presentation of research question **R2**, the meaning of the title of this work *Enhancing Explanatory Interactive Machine Learning – A Generalization of the CAIPI Algorithm* becomes clear: This thesis will present many variations of CAIPI, which together will increase the application bandwidth of model-agnostic XIML. This thesis generalizes the applicability of CAIPI. The generalization will start from a specified algorithmic formalization compared to the original publication (Teso and Kersting, 2019). This has the benefit that CAIPI can be transferred to novel application areas or ML tasks with minimal algorithmic modifications.

By now, the introduction has widely focused on the CAIPI algorithm (Teso and Kersting, 2019), its limitations, and the goals of the subsequent chapters on the enhancement of CAIPI. The next section will take a step back and present the results of a literature review in the research area XIML. It aims to situate the characteristics of CAIPI among related approaches in the XIML domain in a broader scope to account for the origins and the necessity of XIML.

1.3 Related Work

The motivational paragraphs of this introduction have briefly located CAIPI in the research field model-agnostic XIML. This section will discuss approaches related to CAIPI, whilst retaining its focus on XIML. CAIPI optimizes ML models by interacting with explanations (Teso and Kersting, 2019). It is thus a connection between the research areas XAI and interactive ML. Therefore, this section widens its focus to approaches that interact with explanations in a less strict iterative manner and without the model optimization objective. In the nature of a research field that itself is an intersection of two more general research areas lies that some approaches, which belong to one of the original areas, make substantial contributions to the combined field. Hence, this section will occasionally divide into either XAI or interactive ML for results that have vital implications on XIML.

Figure 1.3 summarizes the results of the literature review. It primarily contains XIML methods published until 31 May 2024. It also includes influential approaches to interact with explanations without an iterative model optimization. Figure 1.3 also lists survey, review, and overview articles as well as position papers, empirical evaluations, and user studies[5]. The remainder of this section will separately describe the survey and user study categories with their impact on XIML, before dividing into the category of technical contributions. The latter category will start with interactive ML methods, before incrementally zooming into the more specific niches interactive visualization and model-specific and model-agnostic XIML. For an intuitive distinction, the reviewed XIML methods are summarized in Table 1.1.

[5]Note that the citations in Figure 1.3 are color-coded: green: survey, review, and overview articles and position papers; orange: empirical evaluations and user studies; blue: technical contributions. The color coding, however, is not exact, as some technical contributions also contain user studies, e.g., Teso and Kersting (2019). The *technical contribution* encoding is preferred whenever appropriate.

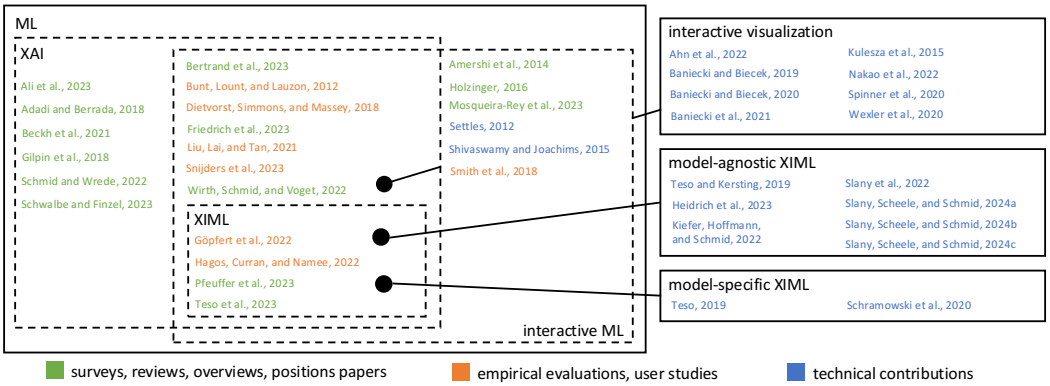

Figure 1.3: Related work.

Surveys, review, and overview articles and position papers Mosqueira-Rey et al. (2023) review interactive ML methods and identify intersections to XAI. They unify sometimes deviating definitions of both research areas. Beckh et al. (2021) emphasize the problem that existing explanatory ML methods allow users only insights into the model and the underlying data, where both might not reflect the actual dynamics of expert-knowledge-intensive use cases. The prior result indicates the necessity for users to revise explanations to refine the model. Human feedback on explanations is an important cornerstone for trustworthy ML (Adadi and Berrada, 2018; Ali et al., 2023; Teso et al., 2023). Interacting with explanations for model adaptation is considered to be a major not yet entirely solved challenge in the XAI community (Gilpin et al., 2018; Schmid and Wrede, 2022), which underpins the relevance of this thesis. Schwalbe and Finzel (2023) propose a taxonomy for XAI, while also discussing the intersection of explanatory and interactive ML. Hoffman et al. (2023) quantify the value of explanations: Not only do the models profit from the refinement by revised explanations, also humans rate explanations as more valuable if they are explorable and adaptable. Users prefer to interact with the decision-making mechanism of a model, rather than solely correcting its decision (Amershi et al., 2014), which can be taken as an additional motivation for this thesis.

The interaction between humans and models in the optimization phase also has algorithmic benefits: a reduction of the computational complexity of ML tasks (Holzinger, 2016) or the labeling effort (Teso et al., 2023). Recently, Friedrich et al. (2023) have proposed a typology for XIML methods. Pfeuffer et al. (2023) introduce an action design research process for XIML projects. A user study review concludes that interacting with explanations improves the joint human ML model task performance (Bertrand et al., 2023). For instance, parameterization tasks in the automotive sector benefit from an explainable human algorithm partnership with corrigibility of explanations (Wirth, Schmid, and Voget, 2022).

Empirical evaluations and user studies Humans refuse to use algorithms once they know that the algorithms' predictions are not necessarily correct (Dietvorst, Simmons, and Massey, 2018). However, the willingness of humans to use algorithms increases with interaction opportunities (Dietvorst, Simmons, and Massey, 2018). Hereby, the type of feedback plays an important role: Annotating spurious correlations is superior compared to annotating the decisive features (Hagos, Curran, and Namee, 2022). A user study of Snijders et al. (2023) assesses the task confidence of subjects who are asked to detect fake news. They conclude that participants, confident in their task, will be less likely to accept decisions of an algorithm. Algorithms

outperform humans in in-distribution tasks, but the joint task performance increases with an increasing amount of out-of-distribution instances (Liu, Lai, and Tan, 2021). The former user studies indicate that humans seek algorithmic contribution. Moreover, the predictive performance benefits from human revision.

Especially older findings, however, indicate the opposite: Bunt, Lount, and Lauzon (2012) evaluate diary entries of humans interacting with intelligent systems. Only a minority of participants seek additional explanations. In the context of topic modeling, Smith et al. (2018) find that interactivity over-amplifies the users' trust in the system. More recently, Göpfert et al. (2022) define *intuitiveness* as a concept that describes to which extent the algorithm's learning behavior aligns with the ability of humans to teach. Active learning of ML models by human users is limited by the intuitiveness of the respective model.

Relevant preliminary methods XIML combines the disclosure of the decision-making mechanism of ML models by explanatory ML techniques (e.g., Schwalbe and Finzel, 2023, and references therein) with the corrigibility of model outcomes by coactive learning (Shivaswamy and Joachims, 2015). Coactive learning iteratively queries user feedback on the prediction of the ML model, which can be seen as an active learning (Settles, 2012) modification that queries the label of instances that are expected to maximize the information gain of a ML model.

Interactive visualization Interactive visualization frameworks satisfy the interactive explanation component of XIML but do not have the primary purpose to optimize ML models or do not iteratively involve users. *TribalGram* (Ahn et al., 2022) is a visualization tool for subgroup analysis in data mining. It incorporates both explanatory and interactive ML concepts, without targeting ML model optimization. Wexler et al. (2020) provide the open-source *What-If?* tool. *exaplAIner* (Spinner et al., 2020) is a tool integrated in Tensorboard, which allows users to interact with local explanations. The *EluciDebug* (Kulesza et al., 2015) framework allows users to refine the model based on explanatory debugging. The explanation interaction is hereby not aligned iteratively. Another framework to visualize and interact with explanations is *modelStudio* (Baniecki and Biecek, 2019), which has been extended to a fairness objective (Baniecki et al., 2021). The latter is closely related to Nakao et al. (2022). The human algorithm interaction in *modelStudio* can even be formalized using a specific grammar (Baniecki and Biecek, 2020).

Model-specific XIML methods Model-specific methods are highly individualized and tailored to specific ML algorithms. For instance, Teso (2019) utilizes linear models that learn their weights with neural networks for active learning. Schramowski et al. (2020) integrate the Right for the Right Reasons loss (Ross, Hughes, and Doshi-Velez, 2017) into coactive learning, making their method suitable for deep learning and image data. The latter uses the Grad-CAM variant t-distributed Stochastic Neighbor Embedding (Maaten and Hinton, 2008) as an explainer.

Model-agnostic XIML methods CAIPI (Teso and Kersting, 2019) can be considered as the root of model-agnostic XIML methods. The model invariance property refers to the basic algorithmic framework of CAIPI with its distinct components (Figure 1.2, right). Specifying some or all of CAIPI's components might still restrict CAIPI to specific models or data. Other model-agnostic XIML approaches are essentially variations of CAIPI: For instance, Slany et al. (2022) propose a user interface that makes CAIPI operable by end-users and use data augmentation for the counterexample generation. Slany, Scheele, and Schmid (2024a) substitute the hand-crafted

Table 1.1: Explanatory interactive machine learning methods. Comparison of **Method**s wrt. their **Data** type, the **Explainer**, and whether they are **Model-agnostic** or model-specific. The presence and absence of a property is indicated by ✓ and ✗. The **-** sign indicates that the original publication contains no specification for the respective property.

Method	Model-agnostic	Explainer	Data
CAIPI (Teso and Kersting, 2019)	✓	LIME (Ribeiro, Singh, and Guestrin, 2016)	Image, Text
CAIPI for computer tomography scans (Slany et al., 2022)	✓	LIME	Image
Bayesian CAIPI (Slany, Scheele, and Schmid, 2024a)	✓	Integrated Gradients (Sundararajan, Taly, and Yan, 2017)	Image
FAIRCAIPI (Heidrich et al., 2023)	✓	SHAP (Lundberg and Lee, 2017)	Tabular
HYXIML (Slany, Scheele, and Schmid, 2024c)	✓	Counterfactuals (Wachter, Mittelstadt, and Russell, 2017)	Tabular
LLMXIML (Slany, Scheele, and Schmid, 2024b)	✓	Counterfactuals	Tabular
Semantic interactive learning (Kiefer, Hoffmann, and Schmid, 2022)	✓	CaSE (Kiefer, 2022)	Text
Active Learning in Self-Explainable Neural Networks (Teso, 2019)	✗	Linear model	-
Interactive learning with RRR loss (Schramowski et al., 2020)	✗	Grad-CAM by t-distributed Stochastic Neighbor Embedding (Maaten and Hinton, 2008)	Image

data augmentation procedure by generative models and infer an improved explanatory performance. CAIPI has also been extended to tabular data, exploiting logically constrained Large Language Models (LLMs) for the generation of counterexamples (Slany, Scheele, and Schmid, 2024b), or to reduce biases of ML models (Heidrich et al., 2023). Slany, Scheele, and Schmid (2024c) conserve the human feedback in probabilistic logic to overcome CAIPI's problem of catastrophic feedback forgetting. For text data, CAIPI has been exploited to semantically improve classification models (Kiefer, Hoffmann, and Schmid, 2022).

Naturally, the adaptation of CAIPI to other application areas requires an adaptation of its components – mostly the explanation component: So use Heidrich et al.

(2023) SHapley Additive exPlanations (SHAP) (Lundberg and Lee, 2017) and Slany, Scheele, and Schmid (2024b) and Slany, Scheele, and Schmid (2024c) counterfactual explanations (Wachter, Mittelstadt, and Russell, 2017) to transfer CAIPI to tabular data. Likewise, Kiefer, Hoffmann, and Schmid (2022) utilize the text-specific explainer Contextual and Semantic Explanations (CaSE) (Kiefer, 2022).

Implications for this thesis The literature review discloses that XIML has emerged from two broader research areas: XAI and interactive ML. Despite there exist some methods enabling users to interact with ML models (e.g., Ahn et al., 2022; Wexler et al., 2020; Spinner et al., 2020), such interaction frameworks are not the focus of this thesis. Instead, this thesis will mostly consider traditional algorithmic methods, which involve users in the optimization process (Settles, 2012; Shivaswamy and Joachims, 2015), and extend them towards the interaction with explanations. Hereby, it is important to understand that the purpose of explanation interactions is a refinement of a ML model wrt. pre-defined quality criteria assessing the model's predictive quality and its ability to follow the correct decision-making mechanism. The iterative optimization process proposed by coactive learning (Shivaswamy and Joachims, 2015) makes explaining specific instances – precisely, those that maximize the information gain of the ML model – mandatory. The explanation of specific instances is subject of local explanation algorithms (e.g., Schwalbe and Finzel, 2023, and references therein). Also, this thesis will exploit a small set of local explanatory ML algorithms depending on the ML task and data types, rather than systematically evaluating local XAI techniques in the context of XIML. All of which, as a consequence, narrows down the set of particularly related methods to the model-agnostic approaches contained by Table 1.1. Remarkable at this point is that the author has contributed most methods in the field (Slany et al., 2022; Slany, Scheele, and Schmid, 2024a; Heidrich et al., 2023; Slany, Scheele, and Schmid, 2024c; Slany, Scheele, and Schmid, 2024b). All of which evolve from Teso and Kersting (2019) and have in common that explanation revisions convey into additional training data, termed counterexamples. Kiefer, Hoffmann, and Schmid (2022) denotes the only exception. This thesis will continuously compare its findings to the methods in the research area in related results paragraphs at the end of each section or chapter. Moreover, certain parts of this thesis, particularly the Chapters 3 and 5, will put their findings into the broader context of increasing the training data set of ML models by counterexamples from explanation revisions.

1.4 Synopsis

This thesis advances CAIPI, the leading model-agnostic XIML method (Teso and Kersting, 2019), through formal, conceptual, and algorithmic enhancements. The thesis is organized into four main chapters, as visualized in Figure 1.4, which illustrates their interplay: The formal foundation will be developed in the Chapters 2 and 3: Chapter 2 will introduce a novel formalization for CAIPI on tabular data. Chapter 3 will investigate how counterexamples affect the optimization cycle in XIML settings. Chapter 4 will vary components of CAIPI to transfer CAIPI to tabular data classification scenarios (Section 4.1) or to increase CAIPI's performance in the image classification domain (Section 4.2). Section 4.3 will propose novel explanatory ML approaches, which, in combination with the proposed feedback injection mechanisms, expand CAIPI's applicability even further, precisely to regression, optimization, and clustering tasks. Chapter 5 will leverage the theoretical insights on XIML from previous chapters to identify and address cases where the weight of

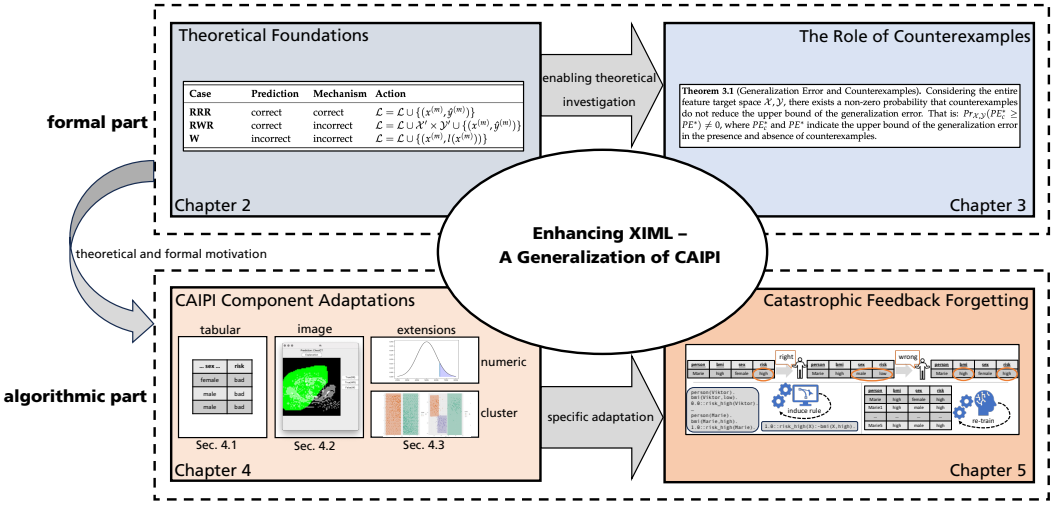

Figure 1.4: Visual synopsis.

user feedback might be insufficient or diminish over the optimization iterations – a phenomenon termed catastrophic feedback forgetting. This modification requires dedicated focus and will be discussed separately in Chapter 5. Chapters 4 and 5 will contain adapted versions of research question **R2**, which will be answered in the respective sections. Chapter 6 will summarize the findings, answer the primary research questions of this thesis, discuss the results, and conclude this thesis by listing open research questions and pointing out further research directories.

This thesis aspires to provide a formal derivation of methods and proofs whenever suitable and feasible. Sometimes, formalizations and proofs become lengthy. In this case, certain parts will be moved into Appendix A. Likewise, Appendix B will contain results that fit not perfectly into the contributions of this thesis but are still interesting within a broader scope of the research area. The Appendices C and D will provide formal information regarding the author's scientific contributions. Precisely, Appendix C will contain author contribution statements to the publications incorporated into this thesis. Appendix D will give an overview of the author's scientific activities, which are not explicitly part of this work.

Note that some of the contents of this thesis have already been accepted for publication at scientific venues. Table 1.2 summarizes the articles that contribute to this thesis, providing a brief summary of the contents and referencing the respective parts of occurrence. A challenge posed by a monograph that is built upon individual articles is to relate the individual scientific contributions to the broader contributions of this thesis. In the context of this thesis, Chapters 4 and 5 will target algorithmic modifications of a single or multiple CAIPI components: Precisely, the Sections 4.1.1 and 4.1.2 will present LLMXIML (Slany, Scheele, and Schmid, 2024b) and FAIRCAIPI (Heidrich et al., 2023). Both are variations of the CAIPI algorithm in the tabular data classification domain. The former generates counterexamples by logically constrained LLMs; the latter applies CAIPI to improve the fairness of ML models learned from data that contain biases in their distribution. The Sections 4.2.1 and 4.2.2, in contrast, will enhance CAIPI in its primary application area – image classification. Section 4.2.1 will propose a user interface for CAIPI enabling optimization experiments by human annotation (Slany et al., 2022). Section 4.2.2 will

Table 1.2: Summary of contained publications. Overview of the publications part of this thesis including the full **Reference**, a brief **Summary**, and the chapter or section (**Chapter/Section**).

Reference	Summary	Chapter/Section
Slany, Emanuel, Stephan Scheele, and Ute Schmid (2024). "Explanatory Interactive Machine Learning with Counterexamples from Constrained Large Language Models". In: KI 2024: Advances in Artificial Intelligence. Ed. by Andreas Hotho and Sebastian Rudolph. Cham: Springer Nature Switzerland, pp. 324–331. DOI: 10.1007/978-3-031-70893-0_26.	CAIPI variant for tabular data using LLMs to generate counterexamples	Section 4.1.1
Heidrich, Louisa, Emanuel **Slany**, Stephan Scheele, and Ute Schmid (2023). "FairCaipi: A Combination of Explanatory Interactive and Fair Machine Learning for Human and Machine Bias Reduction". In: *Machine Learning and Knowledge Extraction* 5.4, pp. 1519–1538. DOI: 10.3390/make5040076.	Bias-mitigating CAIPI variant for tabular data	Section 4.1.2
Slany, Emanuel, Yannik Ott, Stephan Scheele, Jan Paulus, and Ute Schmid (2022). "CAIPI in Practice: Towards Explainable Interactive Medical Image Classification". In: *Artificial Intelligence Applications and Innovations. AIAI 2022 IFIP WG 12.5 International Workshops - MHDW 2022, 5G-PINE 2022, AIBMG 2022, ML@HC 2022, and AIBEI 2022, Hersonissos, Crete, Greece, June 17-20, 2022, Proceedings.* Ed. by Ilias Maglogiannis et al. Vol. 652. IFIP Advances in Information and Communication Technology. Springer, pp. 389–400. DOI: 10.1007/978-3-031-08341-9_31.	Image-specific CAIPI variant proposing a user interface and a generic data augmentation procedure for the counterexample generation	Section 4.2.1
Slany, Emanuel, Stephan Scheele, and Ute Schmid (2024). "Bayesian CAIPI: A Probabilistic Approach to Explanatory and Interactive Machine Learning". In: *Artificial Intelligence. ECAI 2023 International Workshops*. Ed. by Sławomir Nowaczyk et al. Cham: Springer Nature Switzerland, pp. 285–301. DOI: 10.1007/978-3-031-50396-2_16.	CAIPI for image classification leveraging a Variational Autoencoder to generate counterexamples	Section 4.2.2
Amling, Jonas, Stephan Scheele, Emanuel **Slany**, Moritz Lang, and Ute Schmid (2024). "Explainable AI for Mixed Data Clustering". In: Explainable Artificial Intelligence. Ed. by Luca Longo, Sebastian Lapuschkin, and Christin Seifert. Cham: Springer Nature Switzerland, pp. 42–62. DOI: 10.1007/978-3-031-63797-1_3.	Entropy-based XAI method for clustering	Section 4.3.2
Slany, Emanuel, Stephan Scheele, and Ute Schmid (2024). "Hybrid Explanatory Interactive Machine Learning for Medical Diagnosis". In: Artificial Intelligence Applications and Innovations. Ed. by Ilias Maglogiannis et al. Cham: Springer Nature Switzerland, pp. 105–116. DOI: 10.1007/978-3-031-63211-2_9.	Hybrid XIML method to mitigate catastrophic feedback forgetting	Chapter 5

contribute Bayesian CAIPI (Slany, Scheele, and Schmid, 2024a) – a probabilistic variant of CAIPI that improves the plausibility of image counterexamples by building upon statistical generative models. Section 4.3 will contain novel explanatory ML algorithms that perspectively transfer CAIPI to regression, optimization, and clustering. Section 4.3.2, for instance, will present a model-agnostic post-hoc explainer for mixed-data clustering, which is built solely upon statistical information of the obtained clusters (Amling et al., 2024). Finally, Chapter 5 will propose HYXIML as a hybrid model-agnostic XIML method (Slany, Scheele, and Schmid, 2024c)– a combination of ML optimization with CAIPI and probabilistic logic – that overcomes catastrophic feedback forgetting, being a severe conceptual problem in XIML. Additionally, Figure 1.5 depicts an intuitive categorization of the modified components of the CAIPI variants proposed by the publications, which are part of Table 1.2, in relation to the original algorithm (Teso and Kersting, 2019). All of the published contents will be clearly marked at the beginning of the respective chapter or section,

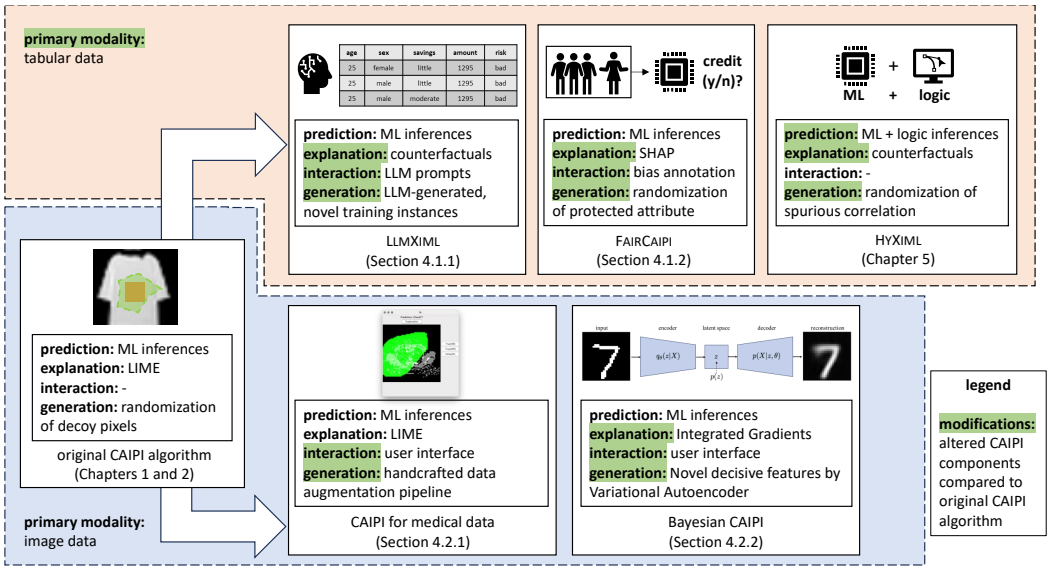

Figure 1.5: CAIPI modifications per chapter.

e.g., "the derivations and results in this section emanate from [Author] ([Year]) and will be paraphrased".

Missing from existing publications is a formal analysis of how an iterative induction of counterexamples affects ML model optimization and an extension of CAIPI to support numerical target variables. These concepts will be addressed in Chapter 3 and Section 4.3.1, respectively. This thesis is structured around scientific publications and complements the published results by novel findings. In summary, it contributes a well-grounded formal investigation of the research topic and a complete algorithmic adaptations part in both regards the CAIPI components as well as the ML tasks. Particular attention is attributed to a unified notation.

The next chapter will derive a novel formalization of CAIPI that is mathematically grounded and overcomes the limitations assigned to the original formalization (Figure 1.2, left). Moreover, Chapter 2 will contain mathematical definitions and derivations of methods that are used throughout this thesis.

Chapter 2

Technical Foundations

This chapter adopts the original idea of Teso and Kersting (2019) but derives a novel formalization, which distinguishes between XIML outcome cases and is mathematically grounded. The novel formalization enables subsequent sections to slightly vary the notation, definitions, and the algorithm itself to adopt CAIPI to various ML tasks (**R2**). Although CAIPI as depicted in Figure 1.2 has been formalized as model-agnostic XIML method (Teso and Kersting, 2019), its formalization leaves room for improvement: Precisely, (i) CAIPI contains no outcome case distinction, e.g., a distinction of how **RRR**, **RWR**, and **W** iterations differently affect the labeled data set size, and (ii) CAIPI's components are not built on mathematical definitions. Overcoming both shortcomings is a crucial first step on the way to answer **R1**.

2.1 Derivation of CAIPI Components

The notation, definitions, and the algorithm of this section have already been published with slight modifications (Heidrich et al., 2023; Slany, Scheele, and Schmid, 2024a; Slany, Scheele, and Schmid, 2024c). The goal of this section is to define a technical foundation for the rest of this work. The notation and some definitions might therefore occasionally differ from the one in the cited publications. Each mathematical component will contain a reference pointing to the theoretical origin.

Note also that the majority of definitions in this thesis concludes with a procedure that abstracts the underlying mathematical behavior. This style ensures precise mathematical definitions, while preserving algorithmic simplicity. Moreover, algorithms can be expressed in a generalized fashion – they are only varied to a minimal extent, even when the mathematical definitions change. This style of formalization is also in line with Teso and Kersting (2019). For now, look at the basic notation:

Notation 2.1 (Tabular Data Classification (Slany, Scheele, and Schmid, 2024c))**.** Let $f : \mathcal{X}^n \to \mathcal{Y}$ be a binary classification model from a feature set $(x_1, ..., x_i, ..., x_n)^T \in \mathcal{X}$ with identifier set $\mathcal{F} = \{1, ..., i, ..., n\}$ to a target set $\mathcal{Y} = \{0, 1\}$. An inference is given as $y = f(x)$. Let $x^{(n)}$ denote the n-th instance in $\mathcal{X}$ and $y^{(n)}$ the n-th instance in $\mathcal{Y}$. Further, let $l : \mathcal{X} \to \mathcal{Y}$ be a binary labeling function. Let $(x_{\mathcal{L}}^{(n)}, y_{\mathcal{L}}^{(n)}) \in \mathcal{L} \subseteq \mathcal{X} \times \mathcal{Y}$ be the n-th instance of the labeled set and $x_{\mathcal{U}}^{(n)}$ be the n-th instance of the unlabeled set $\mathcal{U} \subseteq \mathcal{X}$. A procedure FIT trains or updates f on $\mathcal{L}$.

CAIPI iteratively fits a ML model – for the majority of this thesis, a binary classification model – on a labeled data set for a pre-defined amount of optimization iterations. In each iteration, it selects the most-informative instance (Definition 2.1), which is the instance that is closest to the decision threshold. In the context of binary classifications, this thesis assumes a prediction score interval $[0, 1]$, e.g., obtained by a sigmoid activation function. The decision threshold is a scalar within the prediction score interval: Scores smaller or equal to the threshold yield to the one class and scores greater than the threshold to the other class. The most-informative instance

selection procedure obtains prediction scores for each unlabeled feature instance. It minimizes the absolute difference between the prediction score and the decision threshold.

Definition 2.1 (Most-Informative Instance (Slany, Scheele, and Schmid, 2024c))**.** Assume for this definition that $f(x)$ returns the prediction score. Then, let

$$m = \arg\min_{n} \left\{ |f(x_{\mathcal{U}}^{(n)}) - \alpha| | x_{\mathcal{U}}^{(n)} \in \mathcal{U} \right\}$$

be the index of the *most-informative instance* that is closest to a binary decision threshold $\alpha \in [0, 1]$. Suppose a procedure MII with inputs f and $\mathcal{U}$ that returns m.

This thesis distinguishes between a classification model and a labeling function. It assumes that the labeling function possesses knowledge about the true data-generating process connecting the feature with the target space. This property can be utilized to express the evaluation of the correctness of the classifier's prediction without the prerequisite to know the label (Definition 2.2). This property is especially important, as CAIPI iteratively assesses the correctness of predictions for unlabeled data and corrects the label in case of wrong predictions.

Definition 2.2 (Correct Prediction (Slany, Scheele, and Schmid, 2024c))**.** A *prediction* is *correct* if $f(x) = l(x)$.

The introductory motivation of this thesis has already pointed out that XIML goes beyond coactive learning (Shivaswamy and Joachims, 2015), where users iteratively correct predictions. XIML allows users to additionally interact with the decision-making mechanism (Teso and Kersting, 2019). The labeling function represents the data-generating process – it mimics the *true decision-making mechanism*. Hence, the labeling function is also part of the mathematical expression, which identifies the features of an instance that *are supposed to cause* (Pearl, 2009) a classifier's decision. Based on the labeling function, this thesis infers which features are decisive and which features are indecisive:

Definition 2.3 (Decisive Features (Slany, Scheele, and Schmid, 2024c))**.** Let $v \subseteq \mathcal{F}$ be the subset of *decisive feature* identifiers that cause a decision $l(x)$ either individually or in combination. The remaining features $\bar{v} = \mathcal{F} \setminus v$ are defined to be indecisive.

This thesis will use suitable local explanation methods (e.g., Schwalbe and Finzel, 2023, and references therein) for the respective ML models and tasks to reveal the model's decision-making mechanism – the features that *cause the prediction of the ML model*. The decision making of a ML model is defined to be correct if the decision is solely caused by (a subset of) decisive features:

Definition 2.4 (Correct Decision Making (Slany, Scheele, and Schmid, 2024c))**.** Let the *decision making* of f wrt. an instance x be *correct* if the decision is solely caused by features in v. A local explanation procedure EXP takes f and x (and auxiliary variables) as input and returns $\text{EXP}(f, x)_{\text{out}}$.

Remark 2.1. Note that the expression $\text{EXP}(f, x)_{\text{out}} = \text{EXP}(f, x, [...])$ serves as a placeholder that will be specified and varied throughout this thesis. This thesis will individually define suitable local explanation methods that operationalize the procedure EXP in specific sections. Hereby, the output placeholder $\text{EXP}(f, x)_{\text{out}}$ will be substituted with specific results of the utilized explainers.

At this point, it is extraordinarily important to understand the difference between the *correctness of the decision-making mechanism* and the *correctness of an explanation*. The former is directly connected to the causal structure – the true dynamics that connect features and target (Pearl, 2009). The latter is associated with specific quality criteria such as the fidelity that measures the approximation quality of explanation methods wrt. a ML model (Rudin, 2019). Whenever this thesis assesses the correctness of the decision making of a model, it refers to the former understanding. This thesis uses local explanation methods to reveal the decision-making mechanism of classification models and therefore assumes that the quality of the utilized explanation methods suffices. Thus, in the context of this thesis, explanations cannot be wrong. They can only depict erroneous decision making. The discussed distinction will be upheld in the mathematical parts of this thesis, but in favor of textual brevity, experimental results will contain terms such as *(in)correct explanation*, accurately meaning that the explanation depicts *(in)correct decision making*.

CAIPI (Teso and Kersting, 2019) generates a single or multiple counterexamples in iterations, where the prediction is correct, but a local explainer reveals erroneous decision making of the classification model. Counterexamples are additional, novel training instances that contain only the correlation between decisive features and the target. The decisive features are obtained by a revision of the local explanation. Counterexamples are supposed to refine the decision-making mechanism of a model into the presumably correct direction according to a human user; or, proceeding with the mathematical notation, incrementally align the decision-making mechanism of the ML model with the true decision-making mechanism of the labeling function.

The generated counterexamples satisfy three properties (Definition 2.5): (i) They enforce the correlation between (a subset of) decisive features and the target within a classification model. (ii) They restrict the correlation between indecisive features and the target within a classification model. And (iii) their label belongs to the same class as the original – most-informative – instance.

Definition 2.5 (Counterexamples (Slany, Scheele, and Schmid, 2024c))**.** Let $\mathcal{X}' \times \mathcal{Y}'$ be a *counterexample* set with $|\mathcal{X}'| = |\mathcal{Y}'| = c$. A counterexample feature $x' \in \mathcal{X}'$ emerges from x such that (i) the correlation between a single or multiple features from $\{x_i \text{ if } i \in v | x \in \mathcal{X}\}$ and $\mathcal{Y}$ wrt. a classifier f is enforced and (ii) the correlation between each feature in $\{x_i \text{ if } i \in \bar{v} | x \in \mathcal{X}\}$ and $\mathcal{Y}$ wrt. f is reduced under the constraint that (iii) $l(x') = l(x)$. The counterexample target $y' \in \mathcal{Y}'$ is the correct prediction: $y' = l(x)$. A procedure GEN with inputs (x, y), an at this point arbitrary explanation revision $\text{EXP}(f, x_{\mathcal{U}}^{(m)})_{\text{annot}}$, and c returns $(\mathcal{X}', \mathcal{Y}')$.

Remark 2.2. This thesis will define multiple counterexample generation procedures that precisely state the mechanism behind the procedure GEN.

The Introduction (Chapter 1) has narratively introduced different outcome cases. Table 2.1 uses the preliminary definitions of this section to precisely distinguish between possible XIML outcome cases and their impact on the labeled data set within the CAIPI optimization cycle. Depending on the correctness of the prediction and the decision-making mechanism of a model, CAIPI iterations belong to one of three XIML outcome cases: **R**ight for the **R**ight **R**easons (**RRR**), where prediction and decision-making mechanism are correct, **R**ight for the **W**rong **R**easons (**RWR**), where the prediction is correct but not the decision-making mechanism, and **W**rong (for the wrong reasons) (**W**), where the prediction is incorrect. The reasons yielding to the latter case are always assumed to be incorrect, as the decision-making mechanism must be erroneous for an incorrect outcome (Teso and Kersting, 2019).

Table 2.1: XIML outcome cases (Teso and Kersting, 2019; Slany, Scheele, and Schmid, 2024a). **Case** distinction based on the evaluation results of the **Prediction** and the decision-making **Mechanism** of a model and the associated algorithmic **Action**s for the labeled data set.

Case	Prediction	Mechanism	Action
RRR	correct	correct	$\mathcal{L} = \mathcal{L} \cup \{(x^{(m)}, \hat{y}^{(m)})\}$
RWR	correct	incorrect	$\mathcal{L} = \mathcal{L} \cup \mathcal{X}' \times \mathcal{Y}' \cup \{(x^{(m)}, \hat{y}^{(m)})\}$
W	incorrect	incorrect	$\mathcal{L} = \mathcal{L} \cup \{(x^{(m)}, l(x^{(m)}))\}$

In **RRR** iterations, the most-informative instance is added to the labeled data set without additional intervention (Table 2.1). CAIPI also adds the most-informative instance to the labeled data set in **W** iterations, however, with preliminary label correction. In **RWR** iterations, CAIPI generates counterexamples that append the labeled data set additionally to the most-informative instance.

In each iteration of the CAIPI optimization cycle, users provide feedback at two interaction points (Definition 2.6): they evaluate and, if necessary, correct the prediction and evaluate and, if necessary, revise the explanation. The purpose of the latter correction is two-fold: First, it serves as a subsequent distinction between the cases **RRR** and **RWR**, once the label has been evaluated as being correct. And second, the revised explanation is used as an input containing information about the decisive features for the counterexample generation procedure.

Definition 2.6 (Human Interaction (Slany, Scheele, and Schmid, 2024a))**.** Suppose a procedure INTERACT that indicates *human interaction*, e.g., annotation or evaluation.

Example 2.1. Let $y^{(m)} = \text{INTERACT}(x_{\mathcal{U}}^{(m)})$ denote a human user who labels the most-informative instance. The human interaction substitutes the labeling function.

Example 2.2. Suppose a local explanation for the prediction of the most-informative instance $f(x_{\mathcal{U}}^{(m)})$: Let

$$\text{EXP}(f, x_{\mathcal{U}}^{(m)})_{\text{out}} = \text{EXP}(f, x_{\mathcal{U}}^{(m)})$$
$$\text{EXP}(f, x_{\mathcal{U}}^{(m)})_{\text{annot}} = \text{INTERACT}\left(\text{EXP}(f, x_{\mathcal{U}}^{(m)})_{\text{out}}\right)$$

denote an explanation revision. The subscript annot indicates human annotation.

Remark 2.3. Depending on the ML task, the usage of the INTERACT procedure will differ in the remainder of this thesis. The precise specification will be made clear from its context. For some numerical experiments, the human interaction will be omitted and entirely be substituted by the labeling function.

What appears to be a contradiction at first is that CAIPI (Teso and Kersting, 2019) as an explanatory *interactive* ML method relies on human annotation. Nevertheless, the mathematical formalization introduces a labeling function that possesses knowledge about the data-generating process designed to substitute the human annotator. The contradiction can be solved once the labeling function is treated as an omniscient human user. The labeling function is exploited for an automatic and reliable mathematical and experimental evaluation of algorithms proposed by this thesis, which is crucial for their quantitative assessment. A substitution of the interaction procedure by the labeling function as input in the counterexample generation procedure will therefore be the preferred notation for major parts of this thesis. However, CAIPI being a concept proposed to involve human users, the foundational formalization (Algorithm 2.1) explicitly accounts for human annotation.

Algorithm 2.1: CAIPI($\mathcal{L}, \mathcal{U}, c, n$) (Teso and Kersting, 2019; Slany, Scheele, and Schmid, 2024a)

Input: Data sets $\mathcal{L}$ and $\mathcal{U}$, number of counterexamples c, iteration budget n
Output: Model f

1: **for** $1 : n$ **do**
2: $f \leftarrow \text{FIT}(\mathcal{L})$ ▷ *Notation 2.1*
3: $m \leftarrow \text{MII}(f, \mathcal{U})$ ▷ *Definition 2.1*
4: $\hat{y}^{(m)} \leftarrow f(x_{\mathcal{U}}^{(m)})$
5: $y^{(m)} \leftarrow \text{INTERACT}(x_{\mathcal{U}}^{(m)})$ ▷ *Definition 2.6, Example 2.1*
6: **if** $\hat{y}^{(m)} \neq y^{(m)}$ **then**
7: $\mathcal{L} \leftarrow \mathcal{L} \cup \{(x_{\mathcal{U}}^{(m)}, y^{(m)})\}$ ▷ *Case: **W***
8: **else**
9: $\text{EXP}(f, x_{\mathcal{U}}^{(m)})_{\text{out}} \leftarrow \text{EXP}(f, x_{\mathcal{U}}^{(m)})$ ▷ *Definition 2.4*
10: $\text{EXP}(f, x_{\mathcal{U}}^{(m)})_{\text{annot}} \leftarrow \text{INTERACT}\left(\text{EXP}(f, x_{\mathcal{U}}^{(m)})_{\text{out}}\right)$ ▷ *Def. 2.6, Ex. 2.2*
11: **if** $\text{EXP}(f, x_{\mathcal{U}}^{(m)})_{\text{out}} = \text{EXP}(f, x_{\mathcal{U}}^{(m)})_{\text{annot}}$ **then**
12: $\mathcal{L} \leftarrow \mathcal{L} \cup \{(x_{\mathcal{U}}^{(m)}, \hat{y}^{(m)})\}$ ▷ *Case: **RRR***
13: **else**
14: $\mathcal{X}', \mathcal{Y}' \leftarrow \text{GEN}(x_{\mathcal{U}}^{(m)}, \hat{y}^{(m)}, \text{EXP}(f, x_{\mathcal{U}}^{(m)})_{\text{annot}}, c)$ ▷ *Definition 2.5*
15: $\mathcal{L} \leftarrow \mathcal{L} \cup \{(x_{\mathcal{U}}^{(m)}, \hat{y}^{(m)})\} \cup \mathcal{X}' \times \mathcal{Y}'$ ▷ *Case: **RWR***
16: $\mathcal{U} \leftarrow \mathcal{U} \setminus x_{\mathcal{U}}^{(m)}$
17: **return** f

2.2 The Generalized CAIPI Algorithm

Algorithm 2.1 proposes an alternative formalization of CAIPI compared to Teso and Kersting (2019) depicted in Figure 1.2. It optimizes a model for a fixed amount of iterations, where each iteration starts with fitting the model on the labeled data set (line 2). Afterwards, CAIPI selects the most-informative instance (line 3), of which it obtains the prediction (line 4). The user provides the correct label (line 5) such that the prediction can be corrected in the **W** case, before the most-informative instance is added to the labeled data set (line 7). If the prediction is correct, a local explanation method reveals the decision-making mechanism (line 9). The user revises the explanation (line 10). If the annotated explanation is equal to the generated explanation – no revision was necessary, the explanation is correct –, the most-informative instance is added to the labeled data set (line 12). Otherwise, the revised explanation serves as input for the counterexample generation procedure (line 14). The generated counterexamples are added to the labeled data set together with the most-informative instance to refine the model (line 15). Finally, the most-informative instance is removed from the unlabeled data set to prepare the next iteration (line 16).

In a direct comparison to the original algorithm (Teso and Kersting, 2019), Algorithm 2.1 contains more details: It explicitly distinguishes between the XIML prediction outcome cases. It hereby depicts precisely which kind of user interaction is required in each outcome case. Furthermore, it precisely states how the labeled data set is affected in each outcome case. Moreover, every CAIPI component is based on a mathematical definition. Finally, Algorithm 2.1 does not require narrative annotation to describe its behavior (compare to Figure 1.2).

Note that Algorithm 2.1 and all algorithms that will be defined in this thesis spare out global hyperparameters such as the prediction threshold. CAIPI's primary application area is image classification (Teso and Kersting, 2019), yet the default notational domain is tabular data classification. The reasons are two-fold: First, the next section elaborates XIML on tabular data. Introducing CAIPI on tabular data minimizes notational breaks. Second, the majority of CAIPI variations proposed by this thesis is situated in the domain of tabular data classification, making Notation 2.1 to the most frequently used notation in this thesis.

2.3 Chapter Summary

Summary This chapter has built a theoretical foundation for the model-agnostic XIML method CAIPI (Teso and Kersting, 2019) on tabular data, both mathematically and algorithmically. CAIPI contains components to select the most-informative instance, to reveal the decision-making mechanism, to interact with the prediction as well as to revise the explanation, and to generate counterexamples. All of which have been derived from mathematical definitions. The proposed Algorithm 2.1 explicitly distinguishes between the prediction outcome cases. This chapter is an important foundation for both research questions. **R1** will directly build upon the definitions to mathematically assess the impact of counterexamples. **R2** will adapt the definitions and the algorithm to enhance CAIPI and expand its application spectrum.

Outlook A significant challenge for the notational clarity is posed by the circumstance that this thesis makes contributions in the image and tabular data classification domains as well as in regression, optimization, and clustering. All domains have their unique notational conventions. This thesis will define multiple notational domains to follow the notational conventions within the respective domain and to foster the notational clarity. Obviously, this might cause that some operations are overloaded between the notational domains. This compromise is well-knowingly taken into account as each chapter or section can be assigned to a specific notational domain, which will be defined explicitly or become clear from context. As each section is exclusively part of a single notational domain, inconsistent in-domain notation will be prevented whilst retaining a notational simplicity as notational conventions can be followed. Occasionally, notation will be shared across domains. This will be made explicit while in-domain consistency still is ensured.

Chapter 3

The Role of Counterexamples

The previous chapter has introduced CAIPI (Teso and Kersting, 2019) as a framework for human users to optimize a ML model by interacting with its decision-making mechanism. The human annotation of a local explanation that depicts the current decision-making mechanism of the model conveys into counterexamples that overweigh the human feedback. This chapter mathematically investigates the impact of counterexamples generated by subset sampling from a-priori optimized feature clusters by the k-means algorithm (Lloyd, 1982) on a random forest (Breiman, 2001). The decision-making mechanism of the random forest is retrieved by counterfactual explanations (e.g., Wachter, Mittelstadt, and Russell, 2017).

Random forests, as an ensemble of decision trees, consist of interpretable models (Breiman, 2001). Theoretically, the decision-making mechanism of random forests wrt. a specific prediction can be illustrated by the specific path of an instance given the most probable tree (e.g., Guidotti et al., 2018). CAIPI (Teso and Kersting, 2019) – or more broadly, model-agnostic XIML – is an iterative alignment of model-agnostic procedures. Choosing a random forest is a compromise between a comparatively high predictive quality whilst maintaining mathematical brevity (Chen and Guestrin, 2016). Using a model-agnostic counterfactual explainer (e.g., Guidotti, 2022, and references therein), the random forest can be substituted by any other suitable model in the future without changing the rest of the XIML components.

Problem Current publications on model-agnostic XIML methods investigate an induced amount of counterexamples ranging from zero to five (Teso and Kersting, 2019; Slany et al., 2022; Slany, Scheele, and Schmid, 2024a; Slany, Scheele, and Schmid, 2024c). They mostly put forward that ML models profit monotonously from an increasing amount of counterexamples. Following this assumption, an infinite amount of counterexamples would result in optimal models. The former assumption is exaggerated, as it can be shown that an infinite amount of counterexamples is likely to cause catastrophic forgetting:

Remark 3.1. The induction of an infinite amount of counterexamples causes catastrophic forgetting because $\lim_{c\to\infty}\mathcal{X}\cup\mathcal{X}'=\mathcal{X}'$.

Solution This chapter will answer **R1** – or precisely, provide mathematical evidence on how a reasonable, non-exaggerated amount of counterexamples affects the optimization of a random forest.

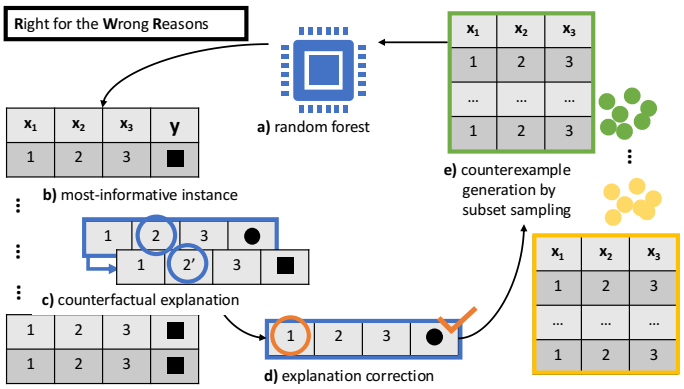

Figure 3.1: Explanatory interactive machine learning on tabular data. A pre-trained random forest (a) selects the most-informative instance (b). Erroneous decision making is revealed by a counterfactual explanation (c), which is corrected by a human user (d). Counterexamples are sampled from the cluster of the most-informative instance (e).

Contribution This section targets the first contribution of this thesis: the mathematical discussion of the effects of counterexamples in XIML. It, moreover, contains some subordinated contributions: (i) It proposes a XIML method (another CAIPI variation such as e.g., Heidrich et al. (2023), Slany, Scheele, and Schmid (2024c), and Slany, Scheele, and Schmid (2024b)) tailored for the binary classification of tabular data. This includes also (ii) a user interface that enables human users without ML expertise to interact with counterfactual explanations with the objective of generating counterexamples. Furthermore, (iii) this chapter builds upon the theoretical results to propose a counterexample filter that controls the amount of added counterexamples in **RWR** iterations. The utility of the latter (iv) is also underpinned by a simulation study – experimental evidence with a controlled data-generating process as common for statistical evaluations in active learning (e.g., Du and Ling, 2010; Chakraborty, 2020). All of which belong to the second contribution of this thesis: adapt and evaluate CAIPI components to expand CAIPI's application spectrum.

Figure 3.1 illustrates how this chapter approaches **RWR** iterations: A random forest (a) predicts the unlabeled data set, from which it selects the most-informative instance (b). A human user evaluates the prediction as correct. The subsequent counterfactual explanation (c) reveals erroneous decision making, which is corrected by the human annotator (d). The entire feature space (labeled as well as unlabeled features) were clustered prior to the optimization cycle with k-means. Counterexamples are samples from the cluster of the most-informative instance (e).

The remainder of this chapter can be split into three parts, before a final summary (Section 3.4) will aggregate the main findings and limitations of this chapter: Section 3.1 will derive the XIML approach proposed by this chapter. It will dedicate one subsection to each component: the random forest, the counterfactual explainer, and the counterexample generator. Section 3.2 will exploit the derivations to provide mathematical evidence on how counterexamples affect the optimization of random forests. A main finding will be: There exist cases where counterexamples are not beneficial for the optimization of random forests. Section 3.3 will deduce a counterexample filter that controls the amount of added counterexamples from the theoretical results. The counterexample filter will be evaluated experimentally using various tabular data sets with controlled data-generating process.

Despite this chapter contributes towards the enhancement of XIML – the optimization of ML models by *interacting* with their explanations –, this chapter does not target interactivity nor does it extend Algorithm 2.1. This chapter provides a mathematical perspective on a specific XIML arrangement to obtain general insights into the concept of *learning from counterexamples*.

3.1 Extending the Mathematical Formalization

During this section, the components of this chapter's XIML procedure for the binary classification of tabular data (Figure 3.1) will be derived in distinct subsections: starting with the random forest (Breiman, 2001), followed by the counterfactual explainer (Guidotti, 2022; Mothilal, Sharma, and Tan, 2020), and concluding with the counterexample generator, which is based on k-means (Lloyd, 1982). The purpose of this section is two-fold: First, all concepts are formalized using Notation 2.1. And second, the counterfactual explainer and the counterexample generator are modified to be integrated in a XIML optimization cycle.

3.1.1 Foundations of Random Forests

Random forests are ensembles of decision trees (Breiman, 2001). A single decision tree (Definition 3.1) obtains its split parameters by recursive partitioning (Definition 3.2). Each node, representing a subset of the feature target space, is split along a single feature such that the resulting subsets maximize (i) the heterogeneity of targets *between* the obtained subsets and (ii) the homogeneity of the target *within* the obtained subsets (Breiman, 2001; Fürnkranz, 2011). The latter is measured by the Gini impurity function (Definition 3.3), which is defined as the weighted sum of the Gini index (Definition 3.4) of each partition. The Gini index is the sum of the squared proportion of each target. The optimal split θ^* in each recursion is determined by $\theta^* = \arg\min_\theta imp(\mathcal{Q}_\theta)$ with θ being a placeholder for a proposal split. Practically, the decision trees within a random forest use a subset of the feature space to foster generalizability and prevent overfitting (Breiman, 2001).

Definition 3.1 (Decision Tree (Breiman, 2001; Fürnkranz, 2011))**.** Let $\theta = (i, \alpha) \in \Theta$ be a split of a *decision tree* h, where $i \in \mathcal{F}$ is a feature identifier and α is a split threshold. An inference for x is denoted as $y = h(x)$. The set of inferences is defined as $\mathcal{Y} = \{h(x, \Theta) | x \in \mathcal{X}\}$ with its shorthands $\mathcal{Y} = h(\mathcal{X}) = h(\mathcal{X}, \Theta)$. The latter notation is preferred whenever the split parameters are referenced explicitly.

Definition 3.2 (Partitioning[6])**.** Let $\mathcal{Q} \subseteq \mathcal{X} \times \mathcal{Y}$ be a subset of the feature target space. Further, let $\mathcal{Q}_\theta^{(left)} = \{(x, y) \text{ if } x_i < \alpha | (x, y) \in \mathcal{Q}\}$ and $\mathcal{Q}_\theta^{(right)} = \mathcal{Q} \setminus \mathcal{Q}_\theta^{(left)}$ be subsets *partitioned* by $\theta = (i, \alpha)$ in the continuous case. Categorical variables are split by sets of possible values.

Definition 3.3 (Gini Impurity)**.** Let the *Gini impurity* wrt. $\mathcal{Q}$ and θ be defined as:

$$imp(\mathcal{Q}_\theta) = \frac{|\mathcal{Q}_\theta^{(left)}|}{|\mathcal{Q}|} gini\left(\mathcal{Q}_\theta^{(left)}\right) + \frac{|\mathcal{Q}_\theta^{(right)}|}{|\mathcal{Q}|} gini\left(\mathcal{Q}_\theta^{(right)}\right),$$

where $gini(...)$ is the Gini index.

[6]Definitions 3.2 and 3.3 stem from `https://scikit-learn.org/stable/modules/tree.html` (31 May 2024). They have been reformulated according to Notation 2.1.

Definition 3.4 (Gini Index (Fürnkranz, 2011))**.** Let the indicator function $\mathcal{I}_{[...]}$ return 1 if the condition in its subscript is met and 0 otherwise. Then, let the *Gini index* measure the binary purity in a subset $\mathcal{Q}$:

$$gini(\mathcal{Q}) = \sum_{y^* \in \{0,1\}} Pr_{y^*}(1 - Pr_{y^*}) = 1 - \sum_{y^* \in \{0,1\}} Pr_{y^*}^2, \text{ where}$$

$$Pr_{y^*} = Pr(y = y^*) = \frac{1}{|\mathcal{Q}|} \sum_{y \in \mathcal{Y}} \mathcal{I}_{[y=y^*]}.$$

Inferences of random forests (Breiman, 2001), being ensembles of decision trees, are simple majority votes. The inference of the random forest is equal to the mode of the distribution over the set of inferences of all decision trees in the random forest:

Definition 3.5 (Random Forest (Breiman, 2001))**.** Let a *random forest* be an ensemble of D decision trees $h(\mathcal{X}, \Theta)$, where Θ is the set of split parameters. Then,

$$y = f(x) = \arg\max_y p(\mathbf{y} = y)$$

denotes an inference of the random forest for feature instance x, where $\arg\max_y p(\mathbf{y} = y)$ is the mode of the distribution over the set of inferences of all decision trees within the random forest, which is defined as:

$$\mathbf{y} = \{y_1, ..., y_D\} = \{h(x, \Theta_d) | d \in \{1, ..., D\}\}.$$

Definition 3.6 (Margin Function (Breiman, 2001))**.** Let Pr_Θ be a probability wrt. Θ. Then, let the *margin function* of any decision tree in the random forest be given as:

$$mr(\mathcal{X}, \mathcal{Y}) = Pr_\Theta(h(\mathcal{X}, \Theta) = \mathcal{Y}) - Pr_\Theta(h(\mathcal{X}, \Theta) \neq \mathcal{Y}).$$

Lemma 3.1 (Generalization Error (Breiman, 2001))**.** The *generalization error PE* of any decision tree within a random forest is said to almost surely converge to $Pr_{\mathcal{X},\mathcal{Y}}(mr(\mathcal{X}, \mathcal{Y}) < 0)$, where $Pr_{\mathcal{X},\mathcal{Y}}$ is a probability wrt. $\mathcal{X}$ and $\mathcal{Y}$.

The margin function (Definition 3.6) is the difference between the truth and the error probability of any decision tree in the random forest. Breiman (2001) proves that the generalization error of any decision tree of a random forest almost surely converges to the probability that the margin function is smaller than zero (Lemma 3.1). In his proof, he distinguishes between decision trees that converge after a finite number of optimization iterations and decision trees that do not. He shows that also the second group of decision trees is expected to converge after an infinite number of optimization iterations, which he terms as *almost sure convergence*.

Definition 3.7 (Inter-Tree Correlation (Breiman, 2001))**.** Let the expected *inter-tree correlation* between any two decision trees within a random forest be defined as

$$\bar{\rho} = \mathtt{E}_{\Theta,\Theta'} \left[\rho(h(\mathcal{X}, \Theta), h(\mathcal{X}, \Theta')) \right],$$

where $\mathtt{E}_{\Theta,\Theta'}$ is the expected correlation between a set of split parameters Θ and any other set of split parameters Θ'. Let $\mathcal{Y}$ be encoded as $\{-1, 1\}$ and $\rho(h(\mathcal{X}, \Theta), h(\mathcal{X}, \Theta'))$ be the correlation coefficient.

Definition 3.8 (Strength (Breiman, 2001))**.** Let $str = \mathtt{E}_{\mathcal{X},\mathcal{Y}} \left[Pr_\Theta(h(\mathcal{X}, \Theta) = \mathcal{Y}) \right]$ be the expected *strength* of any decision tree in the random forest, where $\mathtt{E}_{\mathcal{X},\mathcal{Y}}$ is the expected value wrt. $\mathcal{X}$ and $\mathcal{Y}$.

Lemma 3.2 (Upper Bound of the Generalization Error (Breiman, 2001))**.** The *upper bound of the generalization error* PE^* is said to be $\bar{\rho}(1 - str^2)/str^2$.

The upper bound of the generalization error of random forests (Lemma 3.2) is determined by the expected inter-tree correlation (Definition 3.7) between any two decision trees in the random forest and the expected strength (Definition 3.8) of any decision tree in the random forest (Breiman, 2001). The lower the inter-tree correlation and the greater the strength, the lower is the upper bound of the generalization error, which implies a higher expected predictive quality of the random forest. The proof that the upper bound of the generalization error is given by the expected inter-tree correlation and the expected strength is based on the Chebychev's inequality. Breiman (2001) shows that it evaluates to the outcome of Lemma 3.2.

Definition 3.9 (Most-Informative Instance for Random Forests)**.** Let $\hat{\mathbf{Y}}$ be the set of predictions of a random forest for all $x \in \mathcal{U}$ obtained by

$$\hat{\mathbf{Y}} = \{\hat{\mathbf{y}} = \{h(x, \Theta_d) | d \in \{1, ..., D\}\} | x \in \mathcal{U}\}.$$

The *most-informative instance for* a *random forest* is then defined as:

$$m = \arg\min_n \{\max(p(\hat{\mathbf{y}}^{(n)})) | \hat{\mathbf{y}}^{(n)} \in \hat{\mathbf{Y}}\},$$

where $\max(p(\hat{\mathbf{y}}^{(n)}))$ denotes the mode of the distribution of all decision tree inferences in the random forest for the n-th prediction.

There exist multiple approaches of how to select the most-informative instance with random forests. Suppose a set of random forest predictions, each being the distribution over the inferences of all decision trees in the random forest. Then, the most-informative instance is the prediction with the lowest distribution mode (Definition 3.9). A probabilistic alternative is to select the instance with the lowest second moment from the set of the random forest's predictive distributions.

Applying random forests to binary classification tasks with a decision threshold of 0.5 and assuming deterministic decision tree inferences in the sense that each leaf is a single-point distribution, the most-informative instance selected by Definition 3.9 will be equal to the one chosen by Definition 2.1. This is because, in binary classifications, a simple majority vote is equal to an absolute majority vote (abstention votes are not possible). The inverse of modes belonging to the negative class directly translates into a prediction score below the threshold.

3.1.2 Revisable Counterfactual Explanations

Counterfactual explanations (Wachter, Mittelstadt, and Russell, 2017; White and Garcez, 2020) illustrate the decision boundary of ML models. Take Marie as an example (Figure 3.2): Marie's credit request gets rejected. A counterfactual explainer alters her gender from female to male, which leads to a low credit risk and ultimately results in an acceptance of her request. Hence, counterfactual explanations modify the feature space and answer hypothetical *What if?*-questions to give users an intuition about the location of the decision boundary of the ML system. *Counterfactual explanations* are *counterfactual instances* with explanation purpose generated from a counterfactual explanation algorithm – a *counterfactual explainer*.

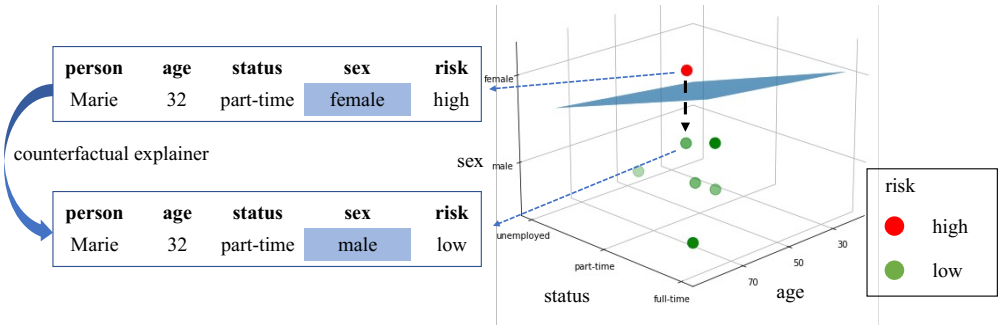

Figure 3.2: Counterfactual explanation. A classifier distinguishes between low and high credit risk along a decision boundary depicted as a blue hyperplane. A counterfactual explainer alters Marie's gender from female to male to cross the decision boundary, which causes a shift in the prediction outcome from high to low risk.

Definition 3.10 (Counterfactual Instance (Artelt, 2019; Mothilal, Sharma, and Tan, 2020))**.** Let $\bar{y}$ be the opposite class of a binary inference $y = f(x)$. A *counterfactual instance* $\bar{x}$ emerges from x such that (i) $f(\bar{x}) = \bar{y}$ and (ii) $\arg\min_{\bar{x}} dist(\bar{x}, x)$, where $dist(...)$ is a suitable distance metric.

Remark 3.2. Assume that $\bar{x} \in \bar{\mathcal{X}}$ is an instance from a set of counterfactual instances all emerging from x. Then, $\bar{\mathcal{X}}$ is supposed to be heterogeneous in the sense that $\arg\max_{\bar{\mathcal{X}}} div(\bar{\mathcal{X}})$, where $div(...)$ is a suitable diversity metric (Mothilal, Sharma, and Tan, 2020).

A counterfactual instance (Definition 3.10) in the classification context is supposed to alter the classification outcome with minimal feature modifications. The former property is known as validity and the latter as similarity (Guidotti, 2022). There exist counterfactual explainers that return multiple counterfactual explanations with a single query. In this case, the set of generated counterfactual explanations is additionally supposed to maximize a third property – heterogeneity – to mediate a broad understanding of the decision boundary (Guidotti, 2022)[7]. Minimality and heterogeneity oppose each other in the sense that increasing the difference among counterfactual explanations might produce indecisive modifications.

In this chapter, counterfactual explanations are generated by a two-stage brute force search consisting of a sampling and a post-processing stage (Definition 3.11). The sampling stage randomly alters a minimal subset of features to obtain the opposite outcome. The post-processing stage iteratively revokes modifications conducted to the feature space. The sampling stage preserves validity, while the post-processing stage accounts for similarity. The post-processor might not identify the optimal counterfactual instance in terms of similarity, but it improves the similarity.

[7]Note that similarity and heterogeneity can be measured by two different concepts: proximity and sparsity (Guidotti, 2022). Proximity approaches similarity by distance metrics and measures the magnitude of feature alternations, whereas sparsity accounts for the number of alternations.

Definition 3.11 (Two-Stage Brute Force Counterfactual Explainer (Guidotti, 2022; Mothilal, Sharma, and Tan, 2020)[8])**.** Let the *two-stage brute force counterfactual explainer* have a sampling and a post-processing stage:

Sampling stage:
Let there be a n-dimensional uniform distribution corresponding to the dimensionality of feature identifiers $\mathcal{F}$ with independent marginals

$$U_n((a,b)^n) = U(a_1,b_1),...,U(a_i,b_i),...,U(a_n,b_n),$$

where (a_i,b_i) are the lower and upper bounds of feature dimension i. Let $u \subseteq \mathcal{F}$ be the subset of modified features such that

$$\begin{aligned} a_i &= \min(\{x_i | x_i \in \mathcal{X}\}) \text{ and } b_i = \max(\{x_i | x_i \in \mathcal{X}\}) && \text{if } i \in u \text{ and} \\ a_i &= b_i = x_i && \text{otherwise.} \end{aligned}$$

Then, let $\bar{x} \sim U_n((a,b)^n)$ with the conditions (i) $\arg\min_u |u|$ and (ii) $f(\bar{x}) = \bar{y}$.

Post-processing stage:
The post-processing objective is defined to be: $\arg\min_{\bar{x}_i} |\bar{x}_i - x_i|$ for $i \in \mathcal{F}$ as long as $f(\bar{x}) = \bar{y}$. Let $q_{x_i}(...)$ return the quartile value wrt. x_i. Let λ indicate the step size. The post-processor repeats

$$\begin{aligned} \bar{x}_i^* &= \bar{x}_i - 10^{-\lambda}\bar{x}_i && \text{if } \bar{x}_i > x_i \text{ and } f(\bar{x}^*) = \bar{y} \text{ and} \\ \bar{x}_i^* &= \bar{x}_i + 10^{-\lambda}\bar{x}_i && \text{if } \bar{x}_i \leq x_i \text{ and } f(\bar{x}^*) = \bar{y} \end{aligned}$$

for a finite amount of iterations, where $i = \arg\max_i \{|q_{x_i}(\bar{x}_i) - q_{x_i}(x_i)| | i \in \mathcal{F}\}$.

Remark 3.3. If multiple counterfactual instances $\bar{x} \in \bar{\mathcal{X}}$ are queried, the heterogeneity of $\bar{\mathcal{X}}$ can be increased by the constraint: $\{(\bar{x} \sim U_n((a,b)^n))\} \cup \bar{\mathcal{X}}$ if $\bar{x} \notin \bar{\mathcal{X}}$.

This chapter utilizes a single counterfactual explanation per XIML iteration. If, however, it would query a set of counterfactual instances, its heterogeneity can be improved if the counterfactual explainer (Definition 3.11) is executed iteratively and novel counterfactual instances are only added if they are unique. Hence, the proposed counterfactual explainer could only find a set of optimal counterfactual instances in the case of complex decision boundaries – when multiple ways exist to alter the classification outcome.

The proposed counterfactual explanation procedure has two obvious limitations: First, numerically encoded originally nominal features, are encoded as natural numbers. The post-processor tries to revert the modifications of nominal-scaled features and treats hereby values of features with intermediate encoding as laying between original and counterfactual instances. And second, it might happen that a modification according to the currently highest quartile difference alters the model's decision. In this case, the respective feature is excluded from post-processing.

Definition 3.12 (Correct Decision Making for Counterfactual Explanations)**.** A *counterfactual explanation* is said to reveal *correct decision making* of a classification model f if (i) $f(\bar{x}) = l(\bar{x})$ and (ii) $\bar{x}_i = x_i$ if $i \notin v$ for $i \in \mathcal{F}$, where l is the labeling function (Notation 2.1) and $v \subseteq \mathcal{F}$ is the subset of decisive feature identifiers (Definition 2.3).

[8]Mothilal, Sharma, and Tan (2020) propose a counterfactual explanation algorithm for differentiable models. Their GitHub repository (`https://github.com/interpretml/DiCE/tree/main`, 04 June 2024) contains a model-agnostic counterfactual explainer, which is formalized by Definition 3.11. The repository nevertheless points to the cited reference. Guidotti (2022) uses the term *brute force* referencing counterfactual explainers that search counterfactual explanations by random sampling.

Figure 3.3: User interface for tabular data. Radar charts visualize the original (left, gray) and the counterfactual instance (right, orange overlay). The gauge charts depict the prediction scores and outcomes of the original (left) and the counterfactual instance (right).

Within the XIML optimization cycle, counterfactual explanations have the purpose to enable users to assess the correctness of the decision-making mechanism. A counterfactual explanation is said to reveal the correct decision-making mechanism if (i) the counterfactual feature instance truly alters the label and (ii) only truly decisive features are modified to obtain the counterfactual instance (Definition 3.12).

Even if not explicitly evaluated nor essential for the results of this chapter, Figure 3.3 showcases a possible user interface that lets users interact with the decision-making mechanism presented as a counterfactual explanation. On the left side, the original instance is displayed in gray together with the classification model's prediction below – a high credit risk. A single counterfactual explanation generated by Definition 3.11 is displayed as an orange overlay with its prediction score on the right side. This way, users can inspect the counterfactual modifications and their impact on the classification score. Users explore the decision boundary. They are enabled to evaluate and potentially revise the counterfactual explanation.

3.1.3 Counterexample Generation by Subset Sampling

In model-agnostic XIML procedures such as CAIPI (Teso and Kersting, 2019), an explanation revision leads to counterexamples. Counterexamples *bias* the training data set of the classification model to re-locate the decision boundary closer to the optimum according to a human annotator or the labeling function in the case of numerical evaluations. In the running example of this chapter, counterexamples are similar instances as Marie's but with random values except for the decisive features – in this case, the employment status.

This subsection proposes a novel counterexample generation procedure for the binary classification of tabular data. It is based on subset sampling from clusters obtained by k-means (Lloyd, 1982). Its formalization has been refined various times (e.g., Bock, 2007). This thesis combines the derivations of Lloyd (1982) and Bock (2007) and formalizes k-means according to Notation 2.1.

Clustering with k-means (Definition 3.13) divides a feature space into k partitions (clusters) with partition centers (cluster prototypes). Intuitively, k-means clustering combines two optimization objectives: It searches for optimal prototypes that minimize the distance to each of their cluster entities. Furthermore, it examines clusters whose entities are closest to the prototypes. The joint optimization function is solved iteratively by alternating the optimization objectives. There exist lower bounds for

each optimization objective in each iteration: The lower bound of the former objective is met for prototypes being the expected values of their clusters. The latter lower bound is reached if each instance is assigned to the cluster with the closest prototype. Practically, *k*-means starts with a random initialization of prototypes and continues the optimization of the alternated objectives for a finite number of iterations or until a stationary solution for clusters and prototypes is found (Bock, 2007).

Definition 3.13 (*k*-means Clustering (Lloyd, 1982; Bock, 2007))**.** Let the *k-means clustering* algorithm divide the feature space into k partitions (clusters) $\mathcal{B} = \{B^{(1)}, ..., B^{(j)}, ..., B^{(k)}\} \subseteq \mathcal{X}$ by:

$$g(\mathcal{B}, \mathcal{Z}) = \sum_{j=1}^{k} \int_{B^{(j)}} \left\| x - z^{(j)} \right\|_2 dp(x),$$

where $\{z^{(1)}, ..., z^{(j)}, ..., z^{(k)}\} = \mathcal{Z}$ with $\mathcal{Z}^n$ are the partition centers (prototypes), $p(x)$ is the distribution over the feature space $\mathcal{X}$ at x, and $\|...\|_2$ is the l2-norm, short for $\sqrt{\sum_{i=1}^{n}(x_i - \mathtt{E}[x_i])^2}$. Let the optimal prototypes for a set partitions $\mathcal{Z}$ wrt. $\mathcal{B}$ denoted as $\mathcal{Z}^*$ and the optimal partitions for a set of prototypes $\mathcal{B}$ wrt. $\mathcal{Z}$ termed $\mathcal{B}^*$ be:

$$\mathcal{Z}^* = (z^{*(1)}, ..., z^{*(j)}, ..., z^{*(k)}) \text{ with } z^{*(j)} = \mathtt{E}\left[\{(x | x \in B^{(j)}) | x \in \mathcal{X}\}\right] \text{ and}$$

$$\mathcal{B}^* = \{B^{*(1)}, ..., B^{*(j)}, ..., B^{*(k)}\} \text{ with}$$

$$B^{*(j)} = \{x | \left\| x, z^{(j)} \right\|_2 = \min(\{\|x, z\|_2 \, | z \in \mathcal{Z}\}) | x \in \mathcal{X}\}.$$

Minimizing $g(\mathcal{B}, \mathcal{Z})$ by altering the objectives has the lower bounds:

$$g(\mathcal{B}, \mathcal{Z}^*) = \sum_{j=1}^{k} \int_{B^{(j)}} \left\| x - \mathtt{E}\left[\{(x | x \in B^{(j)}) | x \in \mathcal{X}\}\right] \right\|_2 \text{ and}$$

$$g(\mathcal{B}^*, \mathcal{Z}) = \int_{\mathcal{X}} \min(\{\|x - z\|_2 \, | z \in \mathcal{Z}\}) dx \text{ for } B \in \mathcal{B}.$$

Definition 3.14 (Counterexample Generation by Subset Sampling)**.** Suppose a set of clusters $\mathcal{B}$ and their prototypes $\mathcal{Z}$ inherited from Definition 3.13. Let *counterexample generation by subset sampling* be executed in four chronological steps:
(i) Retrieve the index of the closest prototype to the feature instance x by

$$j = \arg\min_j \left\{ \left\| x, z^{(j)} \right\|_2 | z^{(j)} \in \mathcal{Z} \right\}.$$

(ii) Sample a counterexample feature set from the selected cluster:

$$\mathcal{X}' \sim B^{(j)} \text{ such that } |\mathcal{X}'| = c.$$

(iii) Obtain the corresponding counterexample target set by

$$\mathcal{Y}' = \{y' = l(x') | x' \in \mathcal{X}'\}.$$

(iv) Randomize indecisive features of each $x' \in \mathcal{X}'$ by

$$\mathcal{X}' \sim \{U_n((a, b)^n) | x' \in \mathcal{X}'\}, \text{ where}$$

$$a_i = \min(\{x'_i | x'_i \in \mathcal{X}'\}) \text{ and } b_i = \max(\{x'_i | x'_i \in \mathcal{X}'\}) \qquad \text{if } i \notin v \text{ and}$$

$$a_i = b_i = x'_i \qquad \text{otherwise.}$$

Counterexample features are sampled from the cluster of the most-informative instance (Definition 3.14). Practically, the proposed targets are the predictions of the classification model. They can be overruled by human annotators or the labeling function if necessary. Decisive features are retrieved from the explanation revision. Indecisive features are randomized by draws from a multivariate uniform distribution (Definition 3.11). Note that the proposed counterexample generator suffers

from the identical limitations in terms of numerically encoded originally nominal features as the counterfactual explainer derived in the previous subsection.

3.2 Theoretical Implications

This section investigates the assumption that more counterexamples are better. This assumption is put forward by evaluations of existing CAIPI variations (Teso and Kersting, 2019; Slany et al., 2022; Slany, Scheele, and Schmid, 2024a), which compare the predictive and explanatory performance of $\{0, 1, 3, 5\}$ counterexamples per **RWR** iteration. The goal of this section is to give a mathematically grounded answer to the question if more counterexamples are better, given the derived components of the previous section. The findings can then be taken as a foundation for algorithmic adaptations, which will be part of the next section, and further mathematical deductions, such as the identification of axioms for which the findings of this chapter hold. The latter is left for future work.

The theoretical implications are restricted to random forests being ensembles of decision trees (Breiman, 2001). Therefore, this section will first investigate how counterexamples affect a single decision tree. Afterwards, the findings will be utilized to examine the effects of counterexamples on the predictive quality and the correct decision making of random forests.

Lemma 3.3 (Decision Trees and Decisive Features)**.** Any split $\theta = (i, \alpha) \in \Theta$ of a decision tree $h(\mathcal{X}, \Theta)$ is more likely to be conducted along decisive features v than on indecisive features. That is: $\forall \theta \in \Theta, Pr_{\Theta}(i \in v) > Pr_{\Theta}(i \notin v)$.

The subsequent deductions are based on Lemma 3.3. Its proof can be found in Appendix A.1. The proof is based on two assumptions: First, indecisive features are assumed to be uncorrelated with the target which can be expressed by a uniform distribution. Second, the feature space is assumed to be dividable into a decisive and an indecisive feature set. Definition 3.3 states that splits of decision trees are determined by the impurity function – the lower, the higher the split probability. Then, it follows from the weak learner condition (Kearns and Valiant, 1994; Schapire, 1990) that learned models with correlation between features and target are superior to random guessing – the absence of correlations. Hence, splits of a decision tree are more likely to be conducted along decisive features.

Proposition 3.1 (Decision Trees and Counterexamples)**.** A positive number of counterexamples does not decrease the probability that a split of a decision tree is conducted along decisive features. Formally: $\forall c \in \mathbb{N}^{+}, Pr_{\Theta_c}(i \in v) \geq Pr_{\Theta}(i \in v)$, where Θ_c is a set of split parameters under the influence of c counterexamples.

Proposition 3.1 states that counterexamples enhance the chance that the correct decision-making mechanism is present in a decision tree. It is proven in Appendix A.2, which investigates the impurity function in extreme cases: when the targets are randomly or perfectly distributed. Perfect means that only instances with equal label are contained by a leaf. Each XIML outcome case is evaluated for each impurity function case. It can be inferred that counterexamples do not interfere with perfect splits but enforce the purity of random splits.

Theorem 3.1 (Generalization Error and Counterexamples)**.** Considering the entire feature target space $\mathcal{X}, \mathcal{Y}$, there exists a non-zero probability that counterexamples do not reduce the upper bound of the generalization error. That is: $Pr_{\mathcal{X},\mathcal{Y}}(PE_c^* \geq PE^*) \neq 0$, where PE_c^* and PE^* indicate the upper bound of the generalization error in the presence and absence of counterexamples.

The upper bound of the generalization error is a measure for the worst expectable predictive behavior of any decision tree in the random forest (Lemma 3.2). It is determined by the expected inter-tree correlation of any two decision trees in the random forest and the expected strength of any decision tree in the random forest. Theorem 3.1 states that there exist cases where the upper bound of the generalization error does not improve under the influence of counterexamples compared to the absence of counterexamples. Intuitively, the proof of Theorem 3.1 (Appendix A.3) shows that there exist cases where counterexamples do neither reduce the expected inter-tree correlation nor increase the expected strength. Those are the cases where Theorem 3.1 holds. Indeed, the expected inter-tree correlation does not decrease with counterexamples because counterexamples are essentially the identical sequence of decisive features. Furthermore, it can be shown that the expected strength only improves for counterexamples that have been generated from an instance, which has a sufficiently high similarity to the instances in the feature set.

Corollary 3.1 (Correct Decision Making and Counterexamples)**.** There exists a non-zero probability that counterexamples do not enhance the correct decision making of a random forest in the sense of Definition 3.12.

The correctness of the decision-making mechanism revealed by a counterfactual explanation is determined by two criteria: (i) The counterfactual feature truly alters the label given a labeling function and (ii) only truly decisive features are altered (Definition 3.12). The probability that the first criterion holds improves with a decreasing upper bound of the generalization error. A generalization error of zero for every decision tree in the random forest would indicate an equality of the random forest and the labeling function given an observed finite data space. It therefore suffices to revisit Theorem 3.1 that states that there exists a non-zero probability that the upper bound of the generalization error does not improve with added counterexamples. Hence, counterexamples do not ensure an improvement of the first criterion, which is a sufficient argumentation for Corollary 3.1. Counterexamples, however, improve the second criterion, as they enhance the probability that the decisive features are present in a decision tree (Proposition 3.1), which can be transferred to the random forest as a sequence of decision trees (Lemma 3.1).

3.3 Experimental Evidence for a Counterexample Filter

The essence of the previous section is: Counterexamples are expected to improve the predictive quality and the ability to follow the correct decision-making mechanism of random forests if the original instance, from which counterexamples are generated, is, on average, similar to the remaining instances in the data set. This section utilizes the theoretical findings and proposes a counterexample filter that controls the number of counterexamples.

Definition 3.15 (Counterexample Filter)**.** Suppose that median$(...)$ returns the median value of a set, m is the index of the most-informative instance (Definition 3.9), and δ is a distance threshold. Let the *counterexample filter* be defined as follows:

$$c = 0 \text{ if median} \left(\left\{ \left\| x^{(m)} - x \right\|_2 | x \in \mathcal{X} \setminus x^{(m)} \right\} \right) > \delta \text{ else } c = 5.$$

The counterexample filter (Definition 3.15) is applied to each **RWR** iteration. It essentially eliminates counterexamples for expected out-of-distribution instances and generates the maximum investigated amount of counterexamples for instances,

where counterexamples are expected to contribute positively to the predictive performance. The range of zero to five counterexamples corresponds to the inspected amount of counterexamples in existing model-agnostic XIML publications (Teso and Kersting, 2019; Slany et al., 2022; Slany, Scheele, and Schmid, 2024a; Slany, Scheele, and Schmid, 2024c).

Setup This section provides experimental evidence on the German Credit Risk (Credit), Adult, Diabetes and Heart data sets[9]. The pre-processing pipeline is identical for all data sets: Rows with missing values are removed. Categorical features are replaced by natural numbers such that the order information is preserved in the case of ordinal data. Traditionally, XIML evaluations involve human annotators (Slany et al., 2022). Other XIML evaluation approaches include the unlearning of induced spurious correlations (Teso and Kersting, 2019; Slany, Scheele, and Schmid, 2024c; Slany, Scheele, and Schmid, 2024b) or the exploitation of remarkable characteristics of the data set. The latter can be divided into the unlearning of biases (Heidrich et al., 2023) or feature-specific knowledge retrieved from deterministic procedures, e.g., that horizontal lines are characteristically for sevens in the binary classification task ones versus sevens (Slany, Scheele, and Schmid, 2024a). This section proposes an additional XIML evaluation approach and utilizes labeling functions to control the data-generating process and substitute the human annotator (Table 3.1). The labeling functions are chosen such that the original proportion of classes is preserved. Some experiments in active learning have a similar setup (e.g., Du and Ling, 2010; Chakraborty, 2020).

The predictive performance is evaluated by the false positive and false negative rates, fp-rate $= fp/(fp + tn)$ and fn-rate $= fn/(fn + tp)$, respectively, where tp is the number of true positives, tn the number of true negatives, fp the number of false positives, and fn is the number of false negatives. The fp- and fn-rates are preferred over precision, recall, and accuracy as they are conditioned on negative and positive labels. They are, therefore, more robust wrt. biases in the data, e.g., the gender bias in the Credit data set. The correct decision making is evaluated by Definition 3.12 and also conditioned on positive and negative correct predictions. For the sake of readability, the presentation of the experimental results will use the terms *correct explanations*, *explanatory performance*, and *correct decision making* as synonyms. This section uses a 70/30 train test split. The test set size for the evaluation of the correct decision-making mechanism is ceiled to 100 per experiment and data set. All experiments are repeated for five different but fixed random seeds. The random forest consists of 100 decision trees with balanced class weights. The minimum split size is set to two. Table 3.2 contains baseline results for the predictive and explanatory performance of random forests trained on all available training data.

For the XIML setting, the random forest is pre-trained on 10 random instances sampled from a n-dimensional uniform distribution, where the marginal distributions cover the feature space of all available training data. The goal is to start with an uninformative model to project all improvements to the XIML optimization. A fixed amount of $\{0, 1, 3, 5\}$ counterexamples per **RWR** iteration is added over the course of 100 XIML optimization iterations. Remind that a XIML optimization with zero counterexamples per **RWR** iteration is equal to coactive learning (Shivaswamy

[9]Credit: `https://www.kaggle.com/datasets/kabure/german-credit-data-with-risk`,
Adult: `https://www.kaggle.com/datasets/wenruliu/adult-income-dataset`,
Diabetes: `https://www.kaggle.com/datasets/akshaydattatraykhare/diabetes-dataset`,
Heart: `https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset`, all 03 June 2024.

Table 3.1: Data-generating process. Labeling functions control the data-generating process. All of which are inputs of a sigmoid function $y^{(n)} = (1 + \exp(-w^{(n)}))^{-1}$. The error terms ϵ are sampled from a normal distribution – specifically, $\epsilon_{\text{Credit}} \sim N(0,1)$, $\epsilon_{\text{Adult}} \sim N(0,4)$, $\epsilon_{\text{Diabetes}} \sim N(0,3)$, and $\epsilon_{\text{Heart}} \sim N(0,1.5)$. Identifiers are replaced by variable names.

Data	**Labeling function** l
Credit	$w^{(n)} = -1 + 0.5 \cdot \mathcal{I}_{\left[\texttt{job}^{(n)} \geq \texttt{skilled}\right]} + \texttt{savingaccounts}^{(n)} + \texttt{checkingaccount}^{(n)} + \texttt{age}^{(n)} \mathcal{I}_{\left[\texttt{age}^{(n)} \leq 60\right]} \cdot \frac{1}{60} + \mathcal{I}_{\left[\texttt{housing}^{(n)} = \texttt{own}\right]} + \epsilon_{\text{Credit}}$
Adult	$w^{(n)} = -4.25 + \mathcal{I}_{\left[\texttt{workclass}^{(n)} = \texttt{private}\right]} + 4 \cdot \frac{\texttt{educationalnum}^{(n)}}{\max(\texttt{educationalnum})} + 5 \cdot \frac{\texttt{hoursperweek}^{(n)}}{\max(\texttt{hoursperweek}^{(n)})} + \epsilon_{\text{Adult}}$
Diabetes	$w^{(n)} = -2.25 + \frac{\texttt{glucose}^{(n)}}{\max(\texttt{glucose})} + 3 \cdot \frac{\texttt{bmi}^{(n)}}{\max(\texttt{bmi})} + 3 \cdot \frac{\texttt{bloodpressure}^{(n)}}{\max(\texttt{bloodpressure})} + \epsilon_{\text{Diabetes}}$
Heart	$w^{(n)} = -2.3 + \mathcal{I}_{\left[\texttt{fbs}^{(n)} > 120\right]} + \mathcal{I}_{\left[\texttt{restecg}^{(n)} \neq \texttt{normal}\right]} + \mathcal{I}_{\left[\texttt{thal}^{(n)} \neq \texttt{normal}\right]} + \frac{\texttt{chol}^{(n)}}{\max(\texttt{chol})} - \frac{\texttt{thalach}^{(n)}}{\max(\texttt{thalach})} + \epsilon_{\text{Heart}}$

Table 3.2: Baseline results on synthetic data. A random forest is trained on all available training data. The **Predictive performance** is calculated by the fp**-** and fn**-rate**. The correctness of the decision-making mechanism is measured by the ratio of **Correct explanations** conditioned on positive (**pos. preds.**) and negative predictions (**neg. preds.**) for each **Data** set. The results are mean values of five experimental iterations with standard deviations in brackets.

	Predictive performance		**Correct explanations**	
Data	fp**-rate**	fn**-rate**	**pos. preds.**	**neg. preds.**
Credit	0.1532 (0.0312)	0.0680 (0.0240)	0.5794 (0.1434)	0.2993 (0.1521)
Adult	0.1013 (0.0049)	0.3082 (0.0141)	0.0000 (0.0000)	0.6084 (0.0187)
Diabetes	0.1281 (0.0190)	0.4229 (0.0502)	0.0269 (0.0394)	0.4912 (0.0630)
Heart	0.2352 (0.0366)	0.2040 (0.0666)	0.2503 (0.0600)	0.3378 (0.1002)

and Joachims, 2015). Additional experiments with the counterexample filter (Definition 3.15) are conducted for each data set. Its threshold parameter δ is the optimal value obtained from a grid search testing threshold values ranging from 0.5 to 5.0 with a 0.5 increment. Prior to the XIML optimization loop, k-means divides the feature space into ten clusters. The GitHub repository[10] contains all code and results of this section and ensures the exact reproducibility of the experiments.

Results Figure 3.4 exemplarily depicts the XIML optimization process for the Credit data set. It visualizes hereby results for zero (blue) and five (red) counterexamples without as well as for five counterexamples with filter (green). The counterexample filter for the Credit data set has a distance threshold of 1.0 retrieved from a grid search. The baseline results are included as gray, dashed, horizontal lines. Figure 3.4 contains subfigures for each evaluation metric: the false positive (top left) and false negative rate (top right) as well as the ratio of correct explanations conditioned on correct positive (bottom left) and correct negative predictions (bottom right). The lines represent the average outcomes of five experimental iterations. Standard deviations are included as transparent fillings around the means. In all plots, a model optimized with XIML at least matches or

[10] `https://github.com/emanuelsla/MathXIML/`, 06 June 2024.

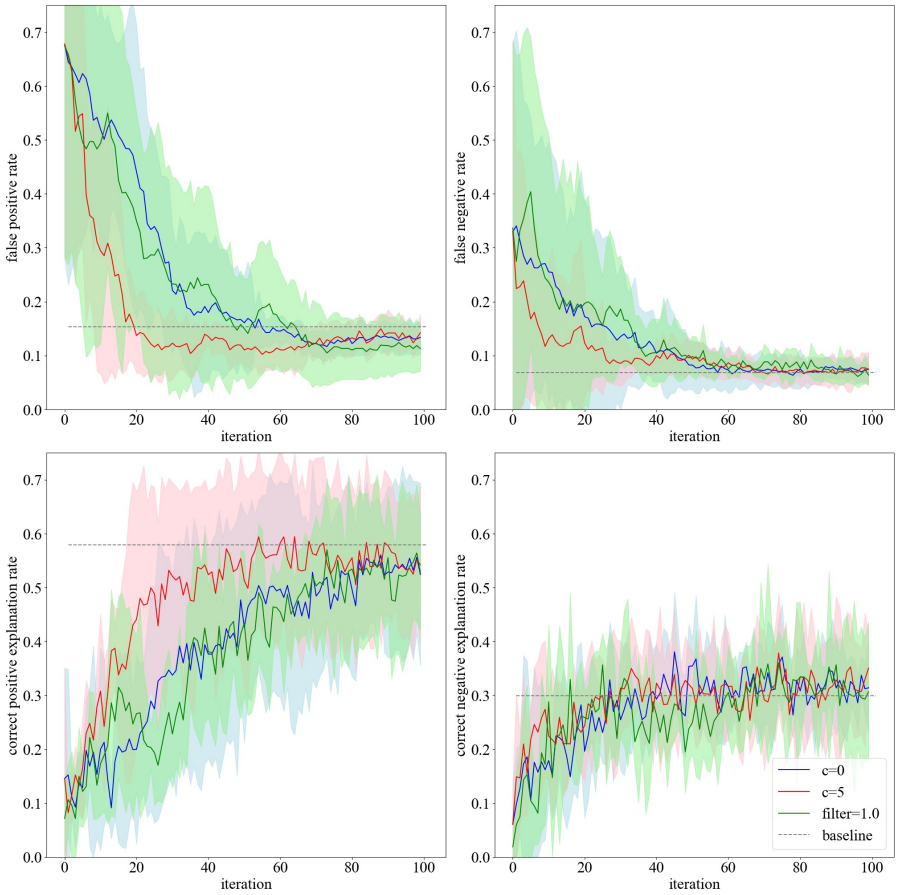

Figure 3.4: Optimization outcome on synthetic Credit data. Comparison of zero (blue) and five (red) counterexamples per **RWR** iteration on the Credit data set with a filtered optimization run with an Euclidean distance threshold of 1.0 (green). Standard deviations of five experimental iterations are highlighted as transparent surroundings wrt. the mean results. Baseline results are depicted as dashed grey lines.

even outperforms the baseline in its best iteration, where the best iteration is not necessarily the final one. Remarkable is that models optimized by the proposed XIML framework contain only 100 distinct training instances and counterexamples if added at the end of the optimization process compared to the baseline model, which has been trained on 70 percent of all available data. An XIML optimization with five counterexamples seems to cause models to converge faster compared to the zero counterexample runs. The performance benefits tend to diminish over the course of the optimization process. The counterexample filter essentially acts as a gatekeeper. Its results are, therefore, compromises between the zero and five counterexample cases. For the Credit data set, the counterexample filter tends to improve the predictive performance given a stable explanatory performance.

Table 3.3 aggregates the optimal results of each of the five experimental iterations given the predictive and explanatory performance metrics, the data sets, and the counterexample variations. The results are mean values with standard deviations in brackets. The best result per metric and data set is written boldly. A particularity of this table are the $\uparrow$ and $\downarrow$ symbols. They indicate whether the induction of the counterexample filter lets the optimization results converge or diverge from the optimum compared to the five counterexamples case without a filter. The symbols are omitted if the five counterexamples run with filter reaches the optimum. The utilized

Table 3.3: Experimental results on synthetic data. The **Predictive performance** with fp**-** and fn**-rate** as well as the ratio of **Correct explanations** for positive (**pos. preds.**) and negative predictions (**neg. preds.**) are compared across **Data** sets and $c = \{0, 1, 3, 5\}$ counterexamples per **RWR** iteration as well as counterexample filters with a threshold δ. The results are mean values of five experimental iterations with standard deviations in brackets. Bold numbers are optimal results per column and data set. The $\uparrow$ and $\downarrow$ symbols mark the positive and negative effects of the counterexample filter compared to $c = 5$ without a filter.

		Predictive performance		**Correct explanations**	
Data	c	fp**-rate**	fn**-rate**	**pos. preds.**	**neg. preds.**
Credit	0	0.0724 (0.0614)	0.0408 (0.0308)	0.6715 (0.1371)	0.4620 (0.0778)
	1	0.0906 (0.0490)	**0.0339** (0.0311)	0.6189 (0.0814)	0.4920 (0.0553)
	3	0.0950 (0.0394)	0.0384 (0.0147)	0.6557 (0.1039)	0.5521 (0.0953)
	5	0.0844 (0.0504)	0.0455 (0.0140)	**0.6801** (0.1183)	0.4842 (0.0666)
	5 (δ=1.0)	**0.0665** (0.0588)	$0.0432^{\uparrow}$ (0.0297)	$0.6362^{\downarrow}$ (0.1251)	**0.5579** (0.0667)
Adult	0	0.0196 (0.0247)	0.3577 (0.0936)	0.0000 (0.0000)	0.7939 (0.1027)
	1	0.0280 (0.0271)	0.3280 (0.0368)	0.0000 (0.0000)	0.7451 (0.0986)
	3	0.0219 (0.0312)	**0.2969** (0.0888)	0.0000 (0.0000)	0.7439 (0.0278)
	5	**0.0127** (0.0158)	0.3210 (0.0569)	0.0000 (0.0000)	0.7600 (0.0822)
	5 (δ=0.5)	$0.0196^{\downarrow}$ (0.0247)	$0.3577^{\downarrow}$ (0.0936)	0.0000 (0.0000)	**0.7973** (0.0979)
Diabetes	0	0.0781 (0.0534)	0.3040 (0.1416)	0.1091 (0.0460)	0.5974 (0.1431)
	1	0.0491 (0.0606)	0.2979 (0.1175)	0.1153 (0.0680)	0.5973 (0.0509)
	3	**0.0254** (0.0203)	0.3128 (0.0710)	0.1315 (0.0624)	**0.6780** (0.0681)
	5	0.0581 (0.0540)	**0.2691** (0.0710)	0.1227 (0.0519)	0.6516 (0.0592)
	5 (δ=3.5)	$0.0439^{\uparrow}$ (0.0362)	$0.3137^{\downarrow}$ (0.0644)	**0.1366** (0.1028)	$0.5898^{\downarrow}$ (0.0736)
Heart	0	0.1716 (0.0769)	0.1136 (0.1213)	**0.3957** (0.1436)	0.4882 (0.0540)
	1	0.1542 (0.0473)	0.1109 (0.1082)	0.3773 (0.1188)	**0.5436** (0.1161)
	3	0.1638 (0.0366)	0.0975 (0.0738)	0.3754 (0.0944)	0.5435 (0.1063)
	5	0.1720 (0.0363)	0.0778 (0.0630)	0.3707 (0.0982)	0.5228 (0.0349)
	5 (δ=4.5)	**0.1487** (0.0663)	**0.0675** (0.0723)	$0.3817^{\uparrow}$ (0.1240)	$0.5253^{\uparrow}$ (0.0443)

distance threshold values for the counterexample filters are extracted from a grid search. Overall, it can be observed that more counterexamples are not always better. Such a trend seems only to exist for the Diabetes data set, where a larger amount of counterexamples tends to positively impact the predictive and explanatory performance. XIML optimization always outperforms the baseline (Table 3.2) with two exceptions: the false negative rate and the ratio of correctly explained positive predictions on the Adult data set, where the latter, however, matches the baseline (even if it is zero). Whenever the five counterexample cases do not already find the best solution, the counterexample filter either yields the best outcome or lies closer to the optimum compared to the plain five counterexamples optimization. Two exceptions are the false negative rate on the Adult data set and the ratio of correctly explained negatives on the Diabetes data set.

3.4 Chapter Summary

Summary This chapter has mathematically derived a XIML method tailored for the binary classification of tabular data. It is based on random forests (Breiman, 2001) and counterfactual explanations (e.g., Wachter, Mittelstadt, and Russell, 2017) embedded into a user interface, which lets human users infer and revise the decision-making mechanism of the classification model. The explanation revision conveys into counterexamples sampled from clusters optimized with k-means (Lloyd, 1982) and overweighs the user's perception of the decision boundary in the subsequent optimization iterations.

The main contribution of this chapter is a mathematical investigation of the effects of inducing counterexamples into model optimization. For random forests, counterexamples serve as a vehicle to inject the decision-making mechanism of a user into the ML model. The model refinement by a user explanation revision and the predictive performance metrics are decoupled in the sense that more counterexamples increase the probability that the decision-making mechanism of any decision tree in a random forest gets adjusted, which has been shown to determine the decision-making mechanism of random forests as entity (Breiman, 2001). However, there exists no guarantee that a local adjustment of the decision boundary causes a global performance improvement. This finding also applies for the ability to achieve the correct decision-making mechanism. Furthermore, it can be shown that excessive human intervention results in catastrophic forgetting.

The theoretical deductions are partially based on the expected strength of any decision tree in the random forest, which, among others, determines the expected predictive behavior and the ability for correct decision making of random forests. It is expected that counterexamples built from in-distribution instances improve the quality of random forests while out-of-distribution instances might interfere with it. The subsequent contribution made by this chapter is a simulation study that compares adding multiple counterexamples and coactive learning to an optimization that employs a gatekeeper choosing between both cases. The experiments show that the counterexample filter improves the optimization results in most cases, either by achieving the best outcome or by finding solutions closer to the optimum.

Answer to research question Overall, this chapter has contributed the theoretical evidence to answer the first research question of this thesis:

R1 How do counterexamples affect the optimization of ML models?
Given random forests, binary classification tasks, and tabular data, counterexamples improve the probability that the decision-making mechanism is adjusted. This does not necessarily cause an improvement in the predictive performance nor for the classifier's ability to follow the correct decision-making mechanism.

Related results This chapter extends the spectrum of current predominantly experimental research in the XIML domain (Teso and Kersting, 2019; Schramowski et al., 2020; Slany et al., 2022; Heidrich et al., 2023; Slany, Scheele, and Schmid, 2024a; Slany, Scheele, and Schmid, 2024c; Slany, Scheele, and Schmid, 2024b). Some XIML procedures that utilize the *learning from counterexamples* concept (Slany et al., 2022; Slany, Scheele, and Schmid, 2024a) struggle to explain the missing improvement caused by the induction of an increasing amount of counterexamples. This chapter uses an identical experimental setup to other model-agnostic XIML evaluations (Teso and Kersting, 2019; Slany et al., 2022; Heidrich et al., 2023; Slany, Scheele, and Schmid, 2024a) with the specificity to control the data-generating process. In comparison to the closest related experimental setups (Heidrich et al., 2023; Slany,

Scheele, and Schmid, 2024b), which also apply a CAIPI variant to tabular data, this simulation study varies the amount of counterexamples. The contributions made by this chapter go a step beyond existing XIML methods: This chapter provides indications why the performance of current CAIPI variants might be limited.

Limitations The derivations, proofs, and evaluations in this chapter have several limitations, which can be sorted into three groups: explicitly exploited mathematical assumptions, implicit mathematical assumptions, e.g., conducted by problem simplifications, and flaws in the simulation study.

- **Explicit mathematical assumptions**:
  Although this chapter aims to provide evidence for model-agnostic XIML procedures, it restricts itself to a specific choice of CAIPI components (Teso and Kersting, 2019) to target the binary classification of tabular data, which is a contradiction. This contradiction can be weakened if the derivations are treated as specific formalization for otherwise model-agnostic CAIPI components. A crucial assumption is the numerical encoding of categorical features. The consequence of the derivational section is that numerical features erroneously gain importance during the counterfactual and counterexample generation (Definitions 3.11 and 3.14). This limitation is even amplified in the proof of Theorem 3.1 (Appendix A.3), where also ordinal features are miss-treated when calculating the l2-norm. The same applies for the definition of the counterexample filter (Definition 3.15).
- **Implicit mathematical assumptions**:
  Even though the key results of this chapter are that counterexamples do not necessarily improve random forests and that expectably improving counterexamples can be separated from others, this chapter does not provide insights on the optimal amount of counterexamples. Another missing abstraction is an answer to the question where exactly the theoretical implications apply. In other words, this chapter lacks axioms that state for which combination of ML task, ML model, data type, counterfactual explainer, and counterexample generator Theorem 3.1 and by that Corollary 3.1 hold. The final not explicitly stated assumption is that indecisive feature target relations by means of the labeling function are expected to be uncorrelated. Famous examples like Clever-Hans predictions (Lapuschkin et al., 2019) prove this assumption wrong. The assumption, however, is a crucial simplification for the proof of Lemma 3.3 (Appendix A.1) and thus also essential for the subsequent theoretical implications. Unanswered questions are: Is this assumption necessary to prove Lemma 3.3? If yes, what consequences arise for the implications of this chapter?
- **Simulation study**:
  The counterexample filter is supposed to be a gatekeeper separating instances where counterexamples are expected to be beneficial from others (Definition 3.15). Nevertheless, the filtered optimization does not always reach the optimum (Table 3.3). One reason is that the threshold value might not be optimal, as it emerges from a grid search with a finite set of possible threshold values. Another reason is that model improvements are often compromises: Samples that improve the false positive rate might not improve the false negative rate. Additionally, the ratio of correct explanations, measuring the correct decision making ability, is conditioned on correct predictions for a given class. More correctly predicted instances might therefore decrease the fraction of instances that have undergone correct decision making. The

simulation study also assumes that counterfactual explanations are always capable of revealing the decision-making mechanism. While this might be true in the restricted controlled setting of this chapter, real-world settings could benefit from additional enhancements of the counterfactual explainer (Definition 3.11). An example of such an enhancement is proposed by Appendix B. The final limitation is the modification of the data-generating process (Table 3.1). It might have benefits for the evaluation of the simulation study and has been conducted such that the class distribution of the original data sets are approximately matched. It, nevertheless, leaves room for improvement: For instance, feature interactions could have been included or the magnitude of the error terms could have been varied.

Chapter 4

CAIPI Component Adaptations

The objective of this thesis is the enhancement of XIML with a generalization of CAIPI (Teso and Kersting, 2019) as a specific focus topic. This thesis has a two-sided view on the terms *enhancement* and *generalization*: One side offers mathematical insights into model optimization with counterexamples as well as an algorithmic specification. The other side takes the specified CAIPI algorithm (Algorithm 2.1) as a starting point for further adaptations to enlarge its application spectrum.

While the first topic has been extensively discussed in the previous chapter, this chapter focuses on the second perspective. Its goal is to propose CAIPI variations that either offer beneficial properties for CAIPI's primary original application area – image classification (Teso and Kersting, 2019) –, or extend CAIPI towards tabular data classification. The XIML framework proposed in Section 3.1 is also a CAIPI variant, which belongs to the second group.

This chapter will be the most extensive chapter of this thesis consisting of two main sections: one for tabular-data-specific (Section 4.1) and one for image-specific adaptations (Section 4.2). Although CAIPI has been initially applied to image data (Teso and Kersting, 2019), this chapter starts with tabular data. Major parts of the notation of the previous chapter that formalizes a XIML method for tabular data can be recycled. This minimizes the amount of breaks in the notation. This chapter will go beyond image and tabular data classification in a third section that will propose explanatory ML frameworks that can be integrated into CAIPI to conquer regression, optimization, and clustering tasks (Section 4.3). The integration of those explanatory ML techniques into CAIPI is left for future work.

This chapter will, in combination with the subsequent chapter, answer **R2**. The majority of sections in this chapter have been published. The explicit reference will be given at the beginning of the respective sections. Additionally, Appendix C will state the author's contribution to the cited articles. As most presented methods have originally been structured as individual scientific contributions, they contain individual research questions. One objective of this chapter is therefore the unification of the sometimes deviating structures. The presentation of each method will start with a motivational paragraph that includes a problem description, a brief summary of the proposed solution, and a statement about the resulting scientific contributions. Furthermore, modified versions of Figure 1.2 (right) will highlight the changes of the proposed CAIPI variant compared to the original algorithm (Figure 1.2, left). As CAIPI is defined to be a model-agnostic framework (Teso and Kersting, 2019), the figures highlighting the modifications spare out the model component. The research questions will be consecutively enumerated, e.g., **R2.1**, **R2.2**, etc., throughout this chapter and answered in short summaries at the end of each section, which also includes specific related results and limitations. The main part will consist of three paragraphs: specific auxiliary methods, a specification of the conducted CAIPI adaptation with a modified CAIPI algorithm if necessary, and experimental evidence.

A chapter summary (Section 4.4) will aggregate all findings. In contrast to the previous chapter summary (Section 3.4), it will not answer one of the primary research questions. The answer to **R2** will be part of the next chapter, which will also consider the results of this chapter.

4.1 Tabular-data-specific Adaptations

This section will propose two tabular-data-specific CAIPI variations: First, the counterexample generation of the CAIPI variant from Section 3.1 will be replaced by Large Language Models (LLMs) (Slany, Scheele, and Schmid, 2024b). LLMs allow users to query an infinite amount of novel training data by a natural language prompt. However, LLMs sometimes produce invalid results (Hammond and Leake, 2023), which is why the counterexample generation procedure will additionally be enhanced by logical post-processing (Section 4.1.1). And second, this section will demonstrate that CAIPI can also be utilized to optimize classification metrics apart from the predictive performance or the ability to follow the correct decision-making mechanism. Precisely, Section 4.1.2 will describe FAIRCAIPI (Heidrich et al., 2023) – a CAIPI variant that reduces biases of ML models and informs users if their annotations induce biases.

4.1.1 Counterexamples by Constrained Large Language Models

This section paraphrases and occasionally extends contents of Slany, Scheele, and Schmid (2024b). It also uses figures, tables, definitions, and an algorithm of Slany, Scheele, and Schmid (2024b), which will be cited explicitly. Appendix C.1 states the contributions the author made in the scope of the referenced publication.

Historically, novel or augmented training data have been generated by methods that leverage statistical concepts – for instance, random sampling or Bayesian methods, such as multiple imputation (Li, Stuart, and Allison, 2015) or Gibbs sampling (Gelfand, 2000). Recently, a XIML method (Slany, Scheele, and Schmid, 2024a) has even exploited generative statistical ML to generate counterexamples (Section 4.2.2). The rise of Generative Pre-trained Transformer (GPT) models (Brown et al., 2020) and the public availability of LLMs that can be operated by end-users using natural language prompts has expanded the opportunities to generate novel training data (Chung, Kamar, and Amershi, 2023; Ribeiro and Lundberg, 2022).

Problem The validation of LLM outputs becomes crucial because LLMs have been shown to potentially produce invalid results from either a semantic or a causal perspective (Hammond and Leake, 2023) – a behavior that is publicly discussed as *LLM hallucination* (e.g., Huang et al., 2023). Ill-generated counterexamples that do not reflect the properties of correct decision making (Definition 3.12) induce incorrect correlations into the training data of ML models. This prevents the ML models from reaching the human-defined optimal decision-making mechanism.

Solution The side-effect is mitigated as soon as the training data set is only appended by counterexamples that reflect the correct decision-making mechanism. Therefore, this section pairs the LLM with a logical program that selects valid counterexamples from the LLM output. The logical program acts as a gatekeeper preventing invalid counterexamples from corrupting the training data.

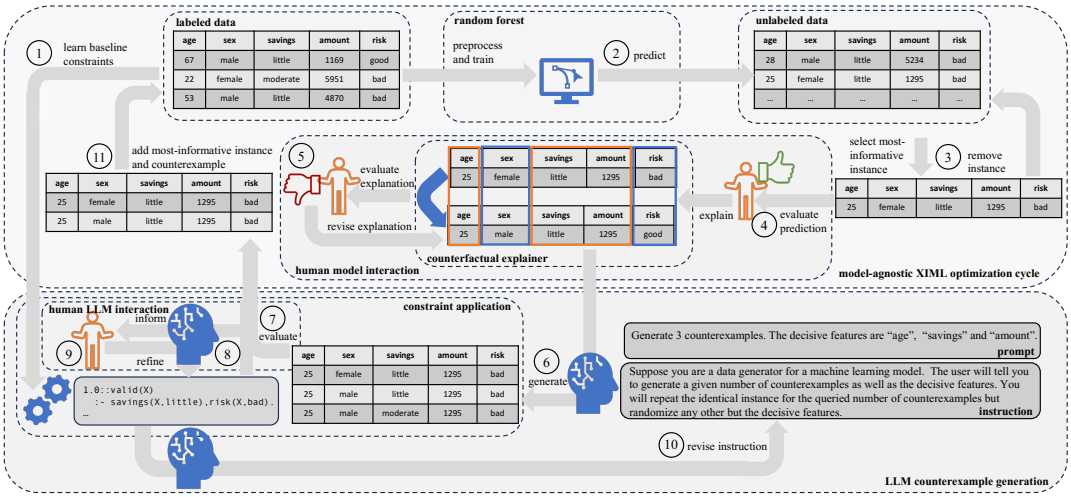

Figure 4.1: LLMXIML overview (Slany, Scheele, and Schmid, 2024b). Counterexample generation with constrained LLMs. A set of validity constraints is induced from the labeled data set (1). A random forest, pre-trained on the labeled data set, predicts the unlabeled data set (2) to select the most-informative instance, which is removed from the unlabeled data set (3). A user evaluates the prediction (4) and the explanation (5). The LLM-generated counterexamples (6) are evaluated by a probabilistic logic program (7). Additionally, the LLM translates violated validity constraints to humans (8) and humans are enabled to generate additional constraints (9). The LLM revises its instruction using violated constraints as prompts (10). The validated counterexamples are added to the training data set (11).

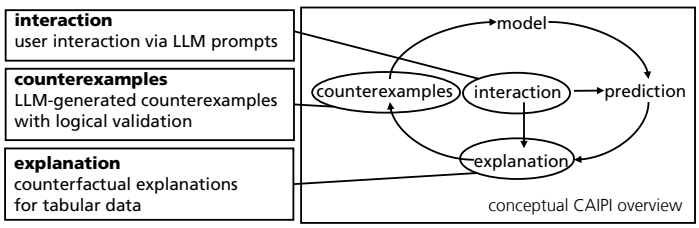

Figure 4.2: LLMXIML component adaptations.

Contribution LLMs have been used for data generation (Chung, Kamar, and Amershi, 2023; Ribeiro and Lundberg, 2022) and they have also already been paired with logical reasoning (Nye et al., 2021) but not in the context of a XIML framework, which is the first contribution of this section. Second, once validity constraints in the gatekeeper program are violated, the LLM prompt to generate counterexamples can automatically be revised. Translators between natural language and logical programs are already available (Yang, Ishay, and Lee, 2023) . In XIML, they can be aligned iteratively to inform users about constraint violations or to refine the constraints – the third contribution.

More specifically, this section introduces the model-agnostic XIML framework LLMXIML (Figure 4.1). In comparison to CAIPI (Teso and Kersting, 2019), it uses counterfactual explanations (Wachter, Mittelstadt, and Russell, 2017) instead of LIME (Ribeiro, Singh, and Guestrin, 2016) as local explanation method (Figure 4.2). The originally absent user interaction is modeled by an LLM, which also generates the counterexamples instead of data augmentation used in traditional CAIPI.

Precisely, LLMXIML (Figure 4.1) utilizes a random forest that selects in each XIML iteration the most-informative instance. A human annotator evaluates its prediction. If the prediction is correct, a counterfactual explainer alters the feature space

to enable human users to assess the decision boundary and thus evaluate the correctness of the decision-making mechanism. In cases of erroneous decision making, counterexamples are queried from the pre-instructed LLM. The counterexamples are evaluated afterwards using a probabilistic logical program containing validity constraints. The user is informed about constraint violations by the LLM that additionally acts as a semantic translator between probabilistic logic and natural language. In this regard, violated constraints can be transformed into LLM instruction revisions and users can also instruct the LLM to refine the validity constraints.

Subordinated research questions Similarly to CAIPI (Teso and Kersting, 2019), LLMXIML is evaluated in its ability to unlearn spurious correlations. In contrast to CAIPI, however, LLMXIML focuses on tabular data instead of image data as the conversion of an unstructured natural language prompt into a tabularly structured natural language output is closer than from natural language into images. This section will answer the following research questions (Slany, Scheele, and Schmid, 2024b):

R2.1 Does reasoning enhance the correctness of counterexamples generated by LLMs?

R2.2 How do LLM-generated counterexamples with and without reasoning affect the predictive and explanatory quality of XIML optimization?

The remaining structure of this section is as follows: First, auxiliary methods will be described. In a direct comparison to Slany, Scheele, and Schmid (2024b), this paragraph will be more extensive. Second, the CAIPI adaptations (Figure 4.2) will be discussed in detail. A third subsection will contain experimental evidence, which will be used to answer the subordinated research questions and to discuss the preliminary findings in a final subsection.

Auxiliary Methods

This section is situated in the domain of Notation 2.1. Figure 4.2 reveals the differences of LLMXIML to CAIPI (Teso and Kersting (2019), Algorithm 2.1) – precisely, the utilization of counterfactual instances as local explanations, a user interaction via LLM prompts, and the counterexample generation and validation by LLMs and probabilistic logic, respectively. While the counterfactual explanation framework of LLMXIML is equal to the one defined in Definition 3.11, the subsequent paragraphs will provide background information on the idea behind LLMs and an introduction in probabilistic logic. The latter part will be split into probabilistic logic and probabilistic rule learning. As the auxiliary methods are no key concepts of this thesis, the introduction will be restricted to necessary prerequisites for LLMXIML and other XIML variants contained by this thesis. References to primary sources and in-depth derivations will be given at the appropriate points.

Large language models Historically, ML architectures for natural language processing (NLP) tasks have been task-specific, meaning that a single model is required for each ML task (Brown et al., 2020). Self-attention network architectures such as present in transformer models provide the ability to capture sequences in a weakly supervised fashion, which means that an embedding sequence is connected with a label (Vaswani et al., 2017). Transformer models, therefore, have enabled ML researchers to use task-agnostic architectures for NLP (Brown et al., 2020). Until recently, a subsequent task-specific fine-tuning step, termed single- or few-shot learning has been required to match the performance of traditional NLP models such as executed in the second generation of GPT (GPT-2) models (Radford et al., 2019).

Thereafter, the NLP models' complexity has increased in both dimensions, the number of training data and the number of inherent parameters, making the third generation of GPT (GPT-3) models *large* language models (Brown et al., 2020). This has enabled ML researchers to apply zero-shot fine-tuning, which is a natural language task description instead of traditional ML training examples (Brown et al., 2020).

The GPT-3 models (Brown et al., 2020) have recently been made publicly available by OpenAI, which makes them operable even by end-users. LLMXIML exploits OpenAI's `gpt-3.5-turbo` model[11] by the integration of an API provided by OpenAI.

Probabilistic logic Despite LLMs being highly versatile and easy to operate, their output might suffer from semantic or causal shortcomings (Hammond and Leake, 2023). LLMXIML will overcome these flaws with probabilistic logical post-processing. Intuitively, probabilistic logic (Definition 4.1) such as proposed by the ProbLog framework (Raedt, Kimmig, and Toivonen, 2007) is a flexible yet simple representation of a probability to which a conjunction of events, so-called body predicates, affects another event, termed target predicate, in rule format. The flexibility of probabilistic logic becomes noticeable once readers understand that each entity in a probabilistic logic program can have its own probability, being it rules, predicates, or facts.

Definition 4.1 (Probabilistic Logic (Raedt, Kimmig, and Toivonen, 2007; Slany, Scheele, and Schmid, 2024c))**.** Let a *probabilistic logic* rule be defined as

$$r = p :: H \texttt{ :- } B \in R,$$

where $p \in [0, 1]$ denotes the rule's truth probability, H is the rule's head, also known as the target predicate that remains constant for all $r \in R$, and B is either a single body predicate, e.g., $B = b$, or a conjunction of body predicates that is $B = b_1, ..., b_n$.

Example 4.1. The following probabilistic logic rule is the only rule in R (Slany, Scheele, and Schmid, 2024c): $0.9 :: \texttt{risk\_high}(X) \texttt{ :- } \texttt{status}(X, \texttt{part-time})$. With a 90 percent probability, a part-time employment status yields a high credit risk.

Remark 4.1. Note that the target predicate is modeled as a unary predicate because the validity constraints will also be of arity one. The target predicate in this context could equally be written as: $\texttt{risk}(X, \texttt{high})$.

Remark 4.2. Using probabilistic logic (Raedt, Kimmig, and Toivonen, 2007), each predicate (and fact) can have its own truth probability, e.g., $p_b :: b$.

Example 4.2. $0.9 :: \texttt{status}(X, \texttt{part-time})$ expresses that 90 percent of part-time employed instances are truly part-time employed, e.g., caused by measurement errors.

Probabilistic programs are evaluated for specific queries given a set of rules by the success probability (Definition 4.2). The success probability is the sum of the products of each rule's probability and the product of the probabilities of all its predicates. This step is comparable to obtaining predictions in a statistical ML setting.

Definition 4.2 (Success Probability (Kimmig et al., 2011; Slany, Scheele, and Schmid, 2024c))**.** Let the *success probability* $Pr_S(q|R)$ of a query q wrt. a set of rules R be

$$Pr_S(q|R) = \sum_{L \subseteq L_R} Pr(q|R) Pr(L|R),$$

where L is a set of ground facts of the set of all ground facts in R denoted as L_R.

[11] `https://platform.openai.com/docs/models/gpt-3-5-turbo`, 01 July 2024.

Example 4.3. Reconsider the probabilistic rule of Example 4.1 and suppose that Marie is part-time employed with a probability of 90 percent. The success probability of Marie having a high credit risk, e.g., $q = \texttt{risk\_high}(\texttt{Marie})$, then is

$$Pr_S(\texttt{risk\_high}(\texttt{Marie})|R) = 0.9 \cdot 0.9 = 0.81.$$

The product of the rule probability and the probability of the predicate present in both the instance Marie and the rule denotes that Marie has a high credit risk with a probability of 81 percent (Slany, Scheele, and Schmid, 2024c).

Probabilistic rule learning The presence of a rule set is a prerequisite for the calculation of the success probability. Figure 4.1 reveals that users are able to refine the set of rules by appending validity constraints. The preliminary rule set, however, is induced by probabilistic rule learning, for which this thesis applies PROBFOIL$^+$ (Raedt et al., 2015), a probabilistic version of the first-order inductive rule learning (FOIL) (Quinlan, 1990) algorithm.

Similar to statistical ML algorithms, probabilistic logic rule learning algorithms require training data, known as probabilistic examples (Definition 4.3). Probabilistic examples model probabilistic logical relations between an instance and an attribute of the instance – in this case, either by unary or binary predicates. The former stands for the certainty to which an instance is associated with a binary variable; the latter expresses the probability that a variable of an instance possesses a certain value.

Definition 4.3 (Probabilistic Example (Raedt et al., 2015; Slany, Scheele, and Schmid, 2024c))**.** Let $t(x^{(n)}) = \texttt{i\_}n$ translate the index of $x \in \mathcal{X}$ into a ProbLog literal and let $name(x_i)$ return the name of feature x_i. Further, let ϕ denote a target predicate that is the target name and the name of the positive class separated by an underscore. A *probabilistic example* $e \in E$ for the n-th instance is defined as follows, where the second column is repeated for each $i \in \mathcal{F}$ and $p = 1$ if $y^{(n)} = 1$ and 0 otherwise:

$$\texttt{instance}\big(t(x^{(n)})\big). \qquad name(x_i)\big(t(x^{(n)}), x_i^{(n)}\big). \qquad p{::}\,\phi\big(t(x^{(n)})\big).$$

The procedure TOPROBEX with inputs $x^{(n)}$, $y^{(n)}$, and ϕ returns e.

Example 4.4. Suppose the probabilistic example of Marie, assuming that she is the first instance in $\mathcal{X}$ ($n = 1$) (Slany, Scheele, and Schmid, 2024c):

$$\texttt{instance}(\texttt{i\_1}). \qquad \texttt{status}(\texttt{i\_1},\texttt{part-time}). \qquad 1.0{::}\,\texttt{risk\_high}(\texttt{i\_1}).$$

Remark 4.3. Definition 4.3 can easily be transformed into modeling validity constraints by adding $1.0{::}\,\texttt{valid}(t(x^{(n)}))$ to each $(x, y) \in \mathcal{L}$, where $\texttt{valid} = \phi$.

Example 4.5. Transferring Example 4.4 into a validity constraint then yields:

$$\texttt{instance}(\texttt{i\_1}). \quad \texttt{status}(\texttt{i\_1},\texttt{part-time}). \quad \texttt{risk}(\texttt{i\_1},\texttt{high}). \quad 1.0{::}\,\texttt{valid}(\texttt{i\_1}).$$

PROBFOIL$^+$ (Raedt et al., 2015) automatically extends the positive probabilistic examples such as in Example 4.5 with negative ones. PROBFOIL$^+$ starts with an empty rule set $R = \emptyset$ and the most general proposal rule $r = p{::}\,H\,\texttt{:-True}$. It iteratively adds predicates if the following criteria are met: (i) Predicates must exceed a prior specified significance threshold, (ii) adding the predicate must improve the m-estimate, and (iii) adding the rule with its predicate to the set of rules must improve the accuracy. It adds novel rules as long as they are beneficial for the accuracy (Raedt et al., 2015; Slany, Scheele, and Schmid, 2024c). Raedt et al. (2015) propose and extensively explain the PROBFOIL$^+$ algorithm. For the remainder, whenever a set of logical rules is mined from probabilistic examples, this thesis assumes a procedure

INDUCE that takes a target predicate ϕ and a set of examples E as input and returns a set of rules R (Slany, Scheele, and Schmid, 2024c).

Adaptations

In a direct comparison to CAIPI (Teso and Kersting, 2019), LLMXIML contains three modifications (Figure 4.2): Already introduced has been the use of a counterfactual explainer for tabular data (Definition 3.11). Novel modifications are the user interaction by means of semantic LLM translations and the counterexample generation and post-processing by LLMs and probabilistic logic, respectively. Counterexamples are generated by a pre-instructed LLM given a prompt that contains information on the current instance and the amount of queried counterexamples. Counterexamples are supposed to mitigate indecisive and enforce decisive feature target correlations (Definition 2.5). Note that Definition 4.4 locally overloads the counterexample generation procedure exclusively for this section.

Definition 4.4 (Counterexamples by LLMs (Slany, Scheele, and Schmid, 2024b))**.** Suppose that GEN (Definition 2.5) in this section takes x, y, and c as input and applies `gpt-3.5-turbo`[12] with the generation instruction and prompt of Figure 4.3 to generate a counterexample feature and target set $\mathcal{X}'$, $\mathcal{Y}'$.

LLMs, being complex and probabilistic models (Brown et al., 2020), might occasionally produce invalid results from a semantic or causal perspective (Hammond and Leake, 2023). Such LLM hallucinations (Huang et al., 2023) can be alleviated by logical validation, which will be called reasoning in the following (Definition 4.5). Intuitively, LLMXIML induces a preliminary set of validity constraints, which can be refined by human annotators. A candidate counterexample is supposed to satisfy each constraint to be logically valid. Otherwise, it is rejected.

Definition 4.5 (Reasoning (Slany, Scheele, and Schmid, 2024b))**.** Suppose that R contains validity constraints induced by PROBFOIL$^+$ (Raedt et al., 2015) from a set of probabilistic examples E (Example 4.5). Let a procedure REASON with inputs $\mathcal{X}$, $\mathcal{Y}$, and R return $\mathcal{X}^*$ and $\mathcal{Y}^*$ – the instances for which the success probability (Definition 4.2) of each $r \in R$ evaluates to $Pr_S > 0$ – such that $\mathcal{X}^* \times \mathcal{Y}^* \subseteq \mathcal{X} \times \mathcal{Y}$.

Example 4.6. Suppose the following validity constraint:

$$1.0 :: \texttt{valid}(X) \texttt{:-} \texttt{risk}(X, \texttt{high}), \texttt{status}(X, \texttt{part-time}).$$

Further, assume that two counterexamples have been generated:

instance	age	gender	risk	status	valid
c1	32	male	high	part-time	1.0
c2	32	female	high	full-time	0.0

The second instance evaluates to zero because it does not satisfy the status predicate. Hence, after the reasoning step, only the first instance persists.

LLMXIML is formalized by Algorithm 4.1. It transfers the labeled data set into logical predicates (line 1) to induce a preliminary set of validity constraints (line 2). Note that the probabilistic examples are constructed such that the original target is appended to the features and the validity target predicate is added. Each optimization iteration starts with fitting the model on the labeled data set (line 4) to select the

[12] `https://platform.openai.com/docs/models/gpt-3-5-turbo`, 01 July 2024.

Algorithm 4.1: LLMXIML$(\mathcal{L}, \mathcal{U}, c, n)$ (Slany, Scheele, and Schmid, 2024b)

Input: Data sets $\mathcal{L}$ and $\mathcal{U}$, number of counterexamples c, iteration budget n
Output: Model f

1: $E \leftarrow \{\text{TOPROBEX}(x \cup y, \texttt{valid}, \texttt{valid}) | (x, y) \in \mathcal{L}\}$ ▷ *Def. 4.3, Re. 4.3*
2: $R \leftarrow \text{INDUCE}(\texttt{valid}, E)$ ▷ *Raedt et al. (2015)*
3: **for** $1 : n$ **do**
4: $\quad f \leftarrow \text{FIT}(\mathcal{L})$ ▷ *Notation 2.1*
5: $\quad m \leftarrow \text{MII}(f, \mathcal{U})$ ▷ *Definition 3.9*
6: $\quad \hat{y}^{(m)} \leftarrow f(x_{\mathcal{U}}^{(m)})$
7: $\quad$ **if** $\hat{y}^{(m)} \neq l(x_{\mathcal{U}}^{(m)})$ **then**
8: $\quad\quad \mathcal{L} \leftarrow \mathcal{L} \cup \{(x_{\mathcal{U}}^{(m)}, l(x_{\mathcal{U}}^{(m)}))\}$ ▷ *Case:* ***W***
9: $\quad$ **else**
10: $\quad\quad$ **if** $\text{EXP}(f, x_{\mathcal{U}}^{(m)})$ **then** ▷ *Definitions 3.11, 3.12*
11: $\quad\quad\quad \mathcal{L} \leftarrow \mathcal{L} \cup \{(x_{\mathcal{U}}^{(m)}, \hat{y}^{(m)})\}$ ▷ *Case:* ***RRR***
12: $\quad\quad$ **else**
13: $\quad\quad\quad \mathcal{X}^*, \mathcal{Y}^* \leftarrow \text{REASON}(\text{GEN}(x_{\mathcal{U}}^{(m)}, \hat{y}^{(m)}, c), R)$ ▷ *Definitions 4.4 and 4.5*
14: $\quad\quad\quad \mathcal{L} \leftarrow \mathcal{L} \cup \{(x_{\mathcal{U}}^{(m)}, \hat{y}^{(m)})\} \cup \mathcal{X}^* \times \mathcal{Y}^*$ ▷ *Case:* ***RWR***
15: $\quad \mathcal{U} \leftarrow \mathcal{U} \setminus x_{\mathcal{U}}^{(m)}$
16: **return** f

most-informative instance from the unlabeled data set (line 5). Its prediction and, if correct, its decision-making mechanism are evaluated. In case of an erroneous prediction, the instance is added with corrected prediction to the labeled data set (line 8). Correct predictions conducted by correct decision making are added to the labeled data set without additional action (line 11). A LLM generates counterexamples in iterations where erroneous decision making leads to the correct outcome (line 13). The generated counterexamples are validated by a logical reasoning step. Only valid counterexamples are added (line 14). Finally, the current most-informative instance is removed from the unlabeled data set (line 15).

LLMXIML (Algorithm 4.1) has three differences compared to the original CAIPI formalization of this thesis (Algorithm 2.1): First, before entering the optimization loop, it induces a set of validity constraints. This modification is related to the second difference, which is a specification of the counterexample generation procedure. And third, LLMXIML does not contain specific human interaction points. The reasons for this are two-fold: First, the experiments in the next subsection will be purely numerical without specific human annotation. And second, Figure 4.1 indicates that the human algorithm interaction in the presence of a LLM and a logical program is more sophisticated than in the original formalization.

Figure 4.3 specifies how semantic translations between probabilistic logic and natural language and the counterexample generation by a LLM extends the human algorithm interaction compared to Algorithm 2.1. It contains LLM instructions and prompts to **generate** counterexamples, to **inform** users about violated validity constraints, and to enable users to **refine** the probabilistic logical programs by natural language. Additionally, Figure 4.3 shows that LLMs can **revise** their own instruction by extending it with translated violated probabilistic logic constraints.

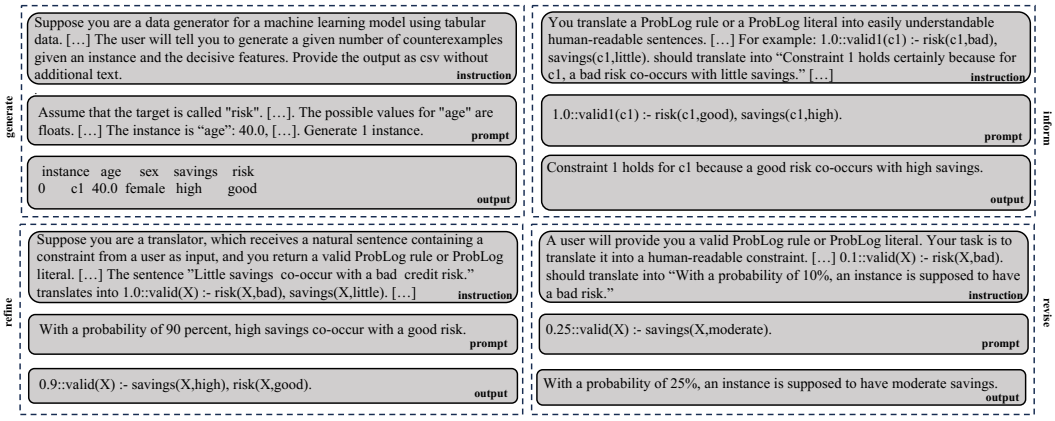

Figure 4.3: LLM interaction (Slany, Scheele, and Schmid, 2024b). Instructions, prompts, and (expected) LLM outputs. LLMXIML uses LLMs to **generate** counterexamples and as a semantic translator between a set of ProbLog rules to **inform** users, to enable users to **refine** the probabilistic logic program, and to **revise** the generation instructions.

Experiments

Setup An ablation study estimates LLMXIML's ability to unlearn a spurious correlation (Definition 4.6) on the the Adult, German Credit Risk (Credit), Diabetes, Diagnostic, and Heart data sets[13]. It compares coactive learning (Shivaswamy and Joachims, 2015), with LLMXIML with and without the reasoning component. LLMXIML without reasoning is CAIPI (Teso and Kersting, 2019) for tabular data (Algorithm 2.1) with LLM-generated counterexamples. The pre-processing is identical for each data set: Rows with missing values are removed, numerical features are standardized, and categorical features are replaced by natural numbers, where the order information for ordinal features is preserved.

Definition 4.6 (Spurious Correlation (Slany, Scheele, and Schmid, 2024c))**.** Let a *spurious correlation* be induced into the labeled instances $(x, y) \in \mathcal{L}$ as follows:

$$x_{\text{spurious}} = l(x) \qquad \text{if } w \leq 0.9, \text{ where } w \sim U(0,1) \text{ and}$$
$$x_{\text{spurious}} \sim \mathcal{Y} \qquad \text{otherwise.}$$

LLM-generated counterexamples are said to be valid if the values for the spurious correlation are valid and the spurious correlation is mitigated (Definition 4.7). In the beginning, the LLM is instructed to also choose invalid values. The experiment is constructed such that the instruction set incrementally narrows down to valid choices by applying the reasoning step (Example 4.7).

Definition 4.7 (Valid LLM-generated Counterexample (Slany, Scheele, and Schmid, 2024b))**.** Let $x_{\text{spurious, c}}$ indicate the spurious correlation feature for counterexamples. For binary classifications, a *LLM-generated counterexample* is *valid* if (i) its value is valid that is $x_{\text{spurious, c}} \in \{0, 1\}$ and (ii) it does not enforce the spurious correlation meaning that $x_{\text{spurious, c}} \neq y$.

[13] Adult: `https://www.kaggle.com/datasets/wenruliu/adult-income-dataset`,
Credit: `https://www.kaggle.com/datasets/kabure/german-credit-data-with-risk`,
Diabetes: `https://www.kaggle.com/datasets/akshaydattatraykhare/diabetes-dataset`,
Diagnostic: `https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data`,
Heart: `https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset`, all 03 July 2024.

Example 4.7. Suppose that the LLM is instructed to choose $x_{\text{spurious, c}}$ from $\{0, 1, 2, ..., 10\}$ for counterexamples. If $x_{\text{spurious, c}} \in \{2, ..., 10\}$ – validity condition (i) failed –, then $\{0, 1, 2, ..., 10\} \setminus x_{\text{spurious, c}}$, which incrementally narrows down the instruction set to valid values (Slany, Scheele, and Schmid, 2024b).

All experiments are repeated for five different but fixed random seeds using a random forest with 100 decision trees and balanced class weights and 250 LLMXIML optimization iterations. Table 4.1 establishes baseline results. It trains classifiers on 70 percent of all available data and assesses the weighted average of precision and recall on the remaining 30 percent. It contains results for the random forest and compares it to a support vector classifier with balanced class weights and a multi-layer perceptron with 100 neurons in a single hidden layer and an adaptable initial learning rate of 0.1[14]. The baseline assessment is not part of Slany, Scheele, and Schmid (2024b). A random forest may only be the optimal choice for the Heart data set. The performance benefits of other models on other data sets, however, are comparatively small, making the random forest still an acceptable choice.

Results Table 4.2 assesses the correctness of counterexamples wrt. Definition 4.7 and compares hereby the presence and absence of the reasoning step for each data set. Without the logical post-processing, the LLM continues to be instructed to choose also invalid values for the spurious correlation feature during the counterexample generation. Therefore, correct counterexamples can only be expected to a minor extent. It can be observed that logical validation – the application of the reasoning component – enhances the correctness of the generated counterexamples. Nevertheless, there exists a high discrepancy regarding the magnitude of the improvement.

Table 4.3 compares the weighted average of precision, recall, and correct explanations conditioned on correct predictions of coactive learning (Shivaswamy and Joachims, 2015) and LLMXIML with and without reasoning across data sets. It shows that LLM-generated counterexamples generally offer predictive and explanatory performance benefits, except on the Adult data set. The variant with the reasoning component is only strictly superior on the Credit and Heart data sets. The overall magnitude of the performance improvements is comparatively low. The graphical assessment confirms the findings (Figure 4.4). Interestingly, counterexamples seem to have measurable benefits especially on the Credit and Diabetes data sets, both having modest predictive performance results. A LLMXIML iteration budget of 250 is insufficient for the Adult data set, as the performance metrics do not converge. Its baseline performance (Table 4.1) is higher. The predictive performance of LLMXIML models is slightly worse compared to the baseline models.

Classifiers that even initially have a profound ability to follow the correct decision-making mechanism result in fewer **RWR** cases. The consequence is that the reasoning component is executed less frequently, giving fewer opportunities to revise the LLM instruction. Therefore, paradoxically, high-performing classifiers are accompanied by lower-quality counterexamples. Lower-quality counterexamples offer less improvement capacities for the classifiers.

[14]Random Forest: `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`, Support Vector Classifier: `https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`, Multi-Layer Perceptron: `https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html`, all 03 July 2024

Table 4.1: Baseline on tabular data. A random forest (**RF**), a support vector machine (**SVM**), and a multi-layer perceptron (**MLP**) are trained on 70 percent of the instances of various tabular **Data** sets to assess the weighted average (**Ø**) of their **Precision** and **Recall** on the remaining 30 percent of instances. Superior results per data set and metric are indicated in bold. Standard deviations of five random seeds are given in brackets.

	Ø Precision			Ø Recall		
Data	**RF**	**SVM**	**MLP**	**RF**	**SVM**	**MLP**
Adult	0.806 (0.0055)	**0.838** (0.0045)	0.812 (0.0045)	0.806 (0.0055)	0.762 (0.0045)	**0.816** (0.0055)
Credit	0.612 (0.0268)	**0.624** (0.026)	0.592 (0.0217)	0.604 (0.0219)	**0.616** (0.027)	0.594 (0.0195)
Diabetes	0.752 (0.0148)	**0.754** (0.0207)	0.732 (0.0311)	**0.756** (0.0114)	0.734 (0.0152)	0.732 (0.0311)
Diagnostic	0.938 (0.0192)	0.944 (0.0207)	**0.946** (0.0089)	0.938 (0.0192)	0.944 (0.0207)	**0.944** (0.0089)
Heart	**0.986** (0.0055)	0.864 (0.0251)	0.972 (0.0179)	**0.986** (0.0055)	0.862 (0.0204)	0.972 (0.0179)

Table 4.2: Ratio of valid counterexamples (Slany, Scheele, and Schmid, 2024b). Comparison of the ratio of valid counterexamples across **Data** sets and the **Mode**s with and and without (w/o) logical validation (reasoning). Standard deviations are given in brackets. Superior results are written as bold numbers. Welch's test results are given as p<0.01: x**, p<0.05: x*.

Data	**Mode:** $c = 1$ w/o reasoning	$c = 1$ with reasoning
Adult	0.0967 (0.0819)	**0.8858**** (0.1413)
Credit	0.1300 (0.0751)	**0.9063**** (0.0557)
Diabetes	0.1304 (0.1068)	**0.4197** (0.0702)
Diagnostic	0.0000 (0.0000)	**0.0291** (0.0482)
Heart	0.0659 (0.0990)	**0.6057*** (0.1564)

Table 4.3: LLMXIML results on tabular data (Slany, Scheele, and Schmid, 2024b). Comparison of the weighted average (Ø) of **Precision**, **Recall**, and the ratio of correct explanations (**Corr. Expl.**) across **Data** sets and **Mode**s: coactive learning (c=0) and XIML (c=1) with and without (w/o) reasoning (r.). Standard deviations are given in brackets. Superior results per metric and data set are written as bold numbers.

Data	**Mode**	**Ø Precision**	**Ø Recall**	**Ø Corr. Expl.**
Adult	c=0	0.7431 (0.0179)	**0.7660** (0.0156)	**0.4166** (0.0122)
	c=1 w/o r.	0.7434 (0.0157)	0.7564 (0.0364)	0.4064 (0.4064)
	c=1 r.	**0.7455** (0.0222)	0.7660 (0.0273)	0.4125 (0.0444)
Credit	c=0	0.6106 (0.0200)	0.6090 (0.0208)	0.3451 (0.0429)
	c=1 w/o r.	0.6005 (0.0197)	0.6013 (0.0215)	0.3687 (0.0387)
	c=1 r.	**0.6217** (0.0328)	**0.6192** (0.0303)	**0.4560** (0.0278)
Diabetes	c=0	0.7290 (0.0251)	0.7191 (0.0317)	0.3929 (0.0521)
	c=1 w/o r.	**0.7398** (0.0222)	**0.7470** (0.0216)	0.4628 (0.0186)
	c=1 r.	0.7393 (0.0207)	0.7452 (0.0205)	**0.4670** (0.0172)
Diagnostic	c=0	0.9074 (0.0085)	0.9059 (0.0083)	0.4987 (0.0018)
	c=1 w/o r.	**0.9083** (0.0078)	**0.9071** (0.0077)	**0.5000** (0.0000)
	c=1 r.	0.9074 (0.0085)	0.9059 (0.0083)	0.4987 (0.0018)
Heart	c=0	0.8949 (0.0217)	0.8932 (0.0232)	0.4974 (0.0010)
	c=1 w/o r.	0.8997 (0.0197)	0.8977 (0.0209)	0.4981 (0.0014)
	c=1 r.	**0.9090** (0.0214)	**0.9068** (0.0238)	**0.4989** (0.0016)

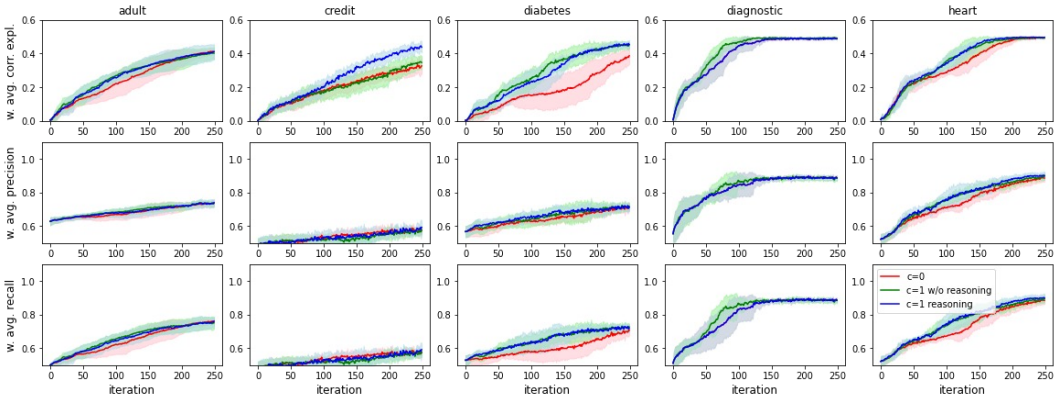

Figure 4.4: LLMXIML optimization on tabular data (Slany, Scheele, and Schmid, 2024b). The weighted average (w. avg.) of precision, recall, and ratio of correct explanations (corr. expl.) of a random forest is evaluated across the data sets over the course of 250 XIML iterations. Each plot compares the absence of counterexamples (c=0, red) to a single LLM-generated counterexample with (c=1 reasoning, blue) and without (c=1 w/o reasoning, green) the logical validation component. The lines depict the mean results over five experimental iterations with standard deviations shown as transparent surroundings.

Section Summary

Summary This section has presented LLMXIML (Figure 4.1) as a model-agnostic XIML variant to train binary classification models on tabular data. LLMXIML utilizes LLMs to generate counterexamples. This section has shown that LLMs frequently generate invalid counterexamples (Table 4.2). Therefore, LLMXIML post-processes counterexamples by a logical reasoning step to filter out invalid counterexamples. The combination of semantic translations of probabilistic logical predicates, LLMs for counterexample generation, and an iterative and interactive optimization setting such as model-agnostic XIML (Teso and Kersting, 2019), makes LLMs revisable. LLM translations of violated validity constraints into natural language extend the LLM instruction to generate counterexamples. The results indicate that such LLM revisions are effective in preventing invalid counterexamples in subsequent optimization iterations (Table 4.2).

Semantic translations between validity constraints and natural language also expand the interaction points (Figure 4.3): Users are informed if counterexamples are invalid. They can also refine the logical program by natural language prompts.

The promising interaction setting and the improved counterexample quality, however, does not yet translate into predictive and explanatory quality improvements (Table 4.3). Although the induction of a LLM-generated counterexample is superior to the absence of counterexamples in most cases, logical post-processing enhances the quality metrics only on a minority of the investigated data sets.

Answers to subordinated research questions The experimental evidence leads to the following answers for this section's research questions (Slany, Scheele, and Schmid, 2024b):

R2.1 Does reasoning enhance the correctness of counterexamples generated by LLMs?
Reasoning by probabilistic logic validation improves the correctness of LLM-generated counterexamples given a random forest and the investigated data sets.

R2.2 How do LLM-generated counterexamples with and without reasoning affect the predictive and explanatory quality of XIML optimization?
In the experimental setting, LLM-generated counterexamples have generally a positive impact on the predictive and explanatory performance of the optimized ML model. The logical post-processing is only beneficial to a minor extent.

Related results Conceptually, LLMXIML borrows ideas from various research directions involving LLMs, e.g., the data generation by LLMs (Chung, Kamar, and Amershi, 2023; Ribeiro and Lundberg, 2022), the integration of logical reasoning into LLMs (Nye et al., 2021), or the combination of logical programs and natural language by LLMs (Yang, Ishay, and Lee, 2023). Algorithmically, LLMXIML modifies the local explanation and counterexample generation component. Compared to CAIPI (Teso and Kersting, 2019), LLMXIML uses counterfactual explanations (e.g., Wachter, Mittelstadt, and Russell, 2017) and LLMs. Both changes make LLMXIML especially suitable for tabular data despite the algorithmic framework still being model-agnostic. This section shows that logical validation improves the LLM output, which is in line with Nye et al. (2021). Similar to other model-agnostic XIML evaluations (Teso and Kersting, 2019; Slany et al., 2022; Heidrich et al., 2023; Slany, Scheele, and Schmid, 2024a; Slany, Scheele, and Schmid, 2024c), LLMXIML's evaluation finds an improved predictive performance and enhanced ability to follow the correct decision-making mechanism when adding counterexamples. In contrast to Teso and Kersting (2019), Slany et al. (2022), and Slany, Scheele, and Schmid (2024a), the experiments in this section do not alter the number of counterexamples. A novel result is that in a XIML setting, models with logically enhanced LLM-generated counterexamples are not strictly superior to models with LLM-generated counterexamples without logical validation.

Limitations The experiments demonstrate that logical validation, on the one hand, mostly even significantly enhances the counterexamples' quality (Table 4.2). On the other hand, the improved counterexample quality does not convey into the predictive nor the explanatory performance of the classification model (Table 4.3). Several limitations contribute to this paradox. They can be grouped into conceptual limitations of the generated counterexamples and flaws in the experimental design. As a thorough theoretical formalization is a claimed key contribution of this thesis, a final group of limitations will address mathematical shortcomings.

- **Generation of counterexamples**:
  In the scope of the experiments, counterexamples are designed to randomize the induced spurious correlation rather than enforce a causal relation. The degree of the spurious correlation is decreased by counterexamples, even if they contain invalid values. Therefore, invalid counterexamples might have the same effect as valid ones. Thus, logical validation for counterexamples designed to randomize spurious correlations offers only a small improvement

potential. A potentially superior alternative is the extraction of validity constraints from structured causal models (e.g., Pearl, 2009), putting logical reasoning into the position to separate causal from in-causal counterexamples.

- **Experimental setup**:
  LLMXIML with a single counterexample and 250 optimization iterations does not find a converged solution for all data sets (Figure 4.4). Hence, the iteration budget and the amount of counterexamples should be increased. The experiments rely on a random forest exclusively, while the baseline test clearly shows that other models perform superior on most data sets (Table 4.1). Despite only being evaluated on tabular data, LLMXIML claims to be a model-agnostic XIML method. This section does not contain a generalized evaluation on multiple data types, which is a necessary improvement step in the future.
- **Theoretical formalization**:
  The derivation of probabilistic logic and probabilistic examples is meant to be general to reconsider the definitions in subsequent sections of this thesis. A visible drawback is that validity constraints cannot be constructed from Definition 4.3 without the additional modification formulated in Remark 4.3. Furthermore, PROBFOIL$^+$ (Raedt et al., 2015) is the first algorithm applied in the context of this thesis without a thorough mathematical derivation. The derivation of PROBFOIL$^+$ is outside the scope of this thesis. Readers interested in the mathematical details are therefore referred to Raedt et al. (2015). Logical reasoning (Definition 4.5) is also loosely defined and refers to the probability that an instance is valid given a set of validity constraints. Although this might be a special case of logical reasoning, it is certainly more specific than following a logical program to obtain a conclusion (Smith, 2020).

4.1.2 Optimization for Fairness

This section rephrases contents originally published in Heidrich et al. (2023). This section contains figures, tables, definitions, and algorithms of Heidrich et al. (2023). The occurrences of which will be cited explicitly. An additional reference is given to Appendix C.2, which contains information about the author's contribution to the cited publication.

In the previous section (Section 4.1.1), model-agnostic XIML has been used to improve the correctness of the decision-making mechanism with a primary focus on predictive performance. Even though the decision-making mechanism of a single instance and a classification model with high predictive performance is correct, counterexamples on a global level might still reproduce biases.

Envision a female customer applying for a loan at a credit institute (Heidrich et al., 2023). The credit institute's risk manager might detect that the assisting classifier has a gender bias, systematically depriving women. Now, consider the counterexamples generated by LLMXIML (Figure 4.1, step 11) that essentially randomize indecisive features – in this case, the gender variable. By chance, counterexamples still contain an over-proportional relation between a female sex and a high credit risk, which amplifies the gender bias. This problem is even intensified for counterexample generation procedures, which are based on conditional probability distributions such as subset sampling (Definition 3.14). Subsets might be split along the gender variable, causing counterexamples to even enforce the gender bias.

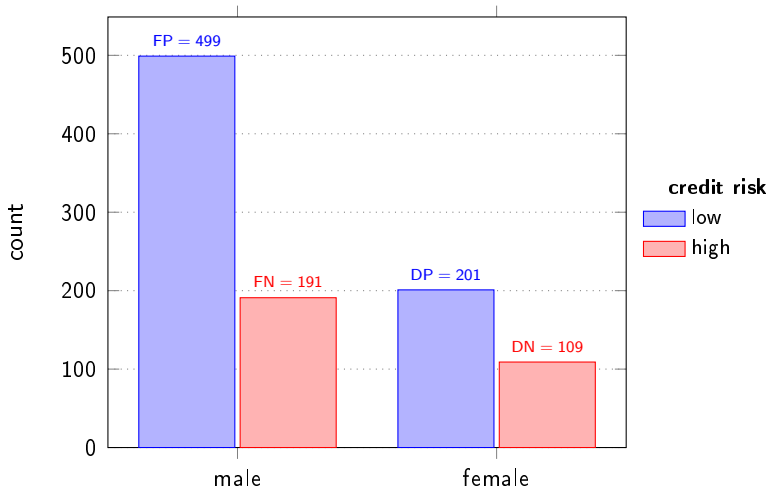

Figure 4.5: Gender distribution on Credit data set (Heidrich et al., 2023). Suppose that men are the favored (F) and women are the deprived (D) group. A low risk is the positive (P) and a high risk is the negative (N) label. Male applicants have a higher proportion of granted loans (FP) compared to female customers (DP).

The German Credit Risk (Credit) data set[15] serves as an example where the probability that partitions separate the gender attribute is elevated. Figure 4.5 visualizes the frequency of applicants associated with a high and low credit risk conditioned on the gender variable. Each bar corresponds to one of the following groups (Kamiran and Calders, 2011; Heidrich et al., 2023):

DP **D**eprived (unprivileged) group with **P**ositive (favorable) label
DN **D**eprived (unprivileged) group with **N**egative (unfavorable) label
FP **F**avored (privileged) group with **P**ositive (favorable) label
FN **F**avored (privileged) group with **N**egative (unfavorable) label

Figure 4.5 suggests that the conditional probability of men having a low credit risk is higher than the one for women. Generally, biases are enforced by instances belonging to either the FP or the DN group.

Problem Disproportional conditional probability distributions of indecisive but sensitive features, e.g., the gender attribute in Figure 4.5, make ML models vulnerable to reproducing data-inherent biases. Even model-agnostic XIML methods that identify learned biases in ML models are not necessarily able to mitigate them, depending on their counterexample generation procedure (e.g., Definition 3.14).

Solution This section proposes FAIRCAIPI (Figure 4.6) as a solution to mitigate learned biases in classification models. In comparison to CAIPI (Teso and Kersting, 2019), its primary optimization objective is the improvement of fairness metrics given a pre-trained classification model (Figure 4.7). FAIRCAIPI focuses on tabular data. Local explanations from SHAP (Lundberg and Lee, 2017) reveal whether the most-informative instance reproduces biases. In the case of biased decision making, counterexamples randomize the protected attribute – in this section, the gender variable.

[15] `https://aif360.readthedocs.io/en/latest/modules/generated/aif360.datasets.GermanDataset.html`, 10 July 2024.

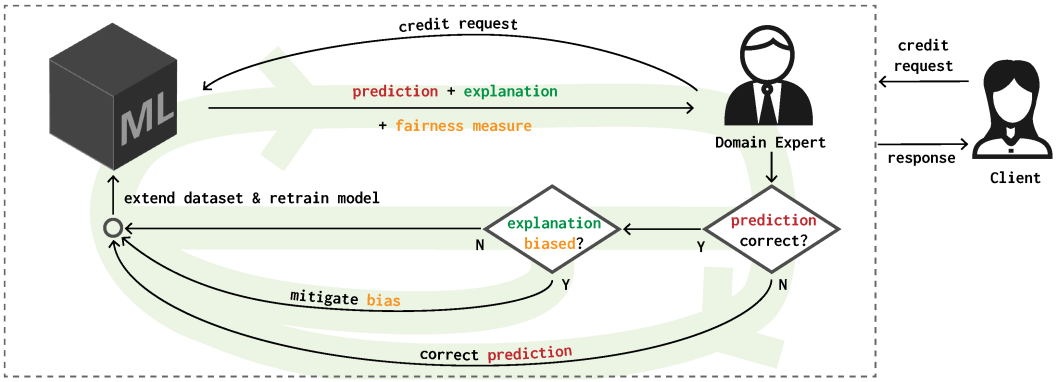

Figure 4.6: FAIRCAIPI overview (Heidrich et al., 2023). XIML in the domain of lending. A client applies for a loan. The credit institute's risk manager is supported by a ML model, assessing the customer's creditworthiness. If the ML model correctly suggests to decline the credit request out of erroneous reasons – for instance, because the applicant is a woman –, the accountant has the opportunity to mitigate the gender bias by re-training the model.

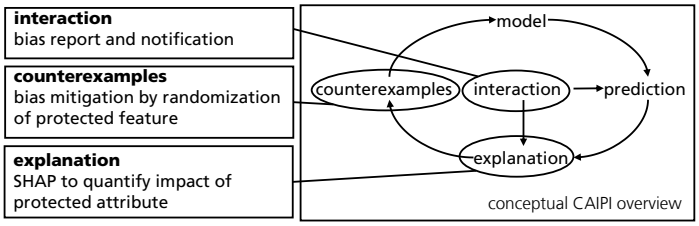

Figure 4.7: FAIRCAIPI component adaptations.

Contribution In the fair ML literature, various methods exist to improve the fairness of ML models. Heidrich et al. (2023) offer an extensive overview. Broadly, bias mitigation methods can be sorted into pre-, in-, and post-processing categories. The latter category is the smallest group, constraining model outcomes or modifying the decision threshold. Pre-processing methods modify the weight of instances before model training. FAIRCAIPI is an in-processing method. The closest related in-processing method that also interacts with explanations is from Nakao et al. (2022). The interaction, however, is not strictly iteratively, making FAIRCAIPI the only model-agnostic XIML method for bias mitigation according to the definition of this thesis (Figure 1.3). Hence, FAIRCAIPI is a model-agnostic XIML framework that (i) uncovers and (ii) reduces learned biases of ML models. Furthermore, FAIRCAIPI (iii) detects human bias during the optimization phase (Heidrich et al., 2023).

Subordinated research questions FAIRCAIPI is evaluated in its ability to unlearn the gender bias of the Credit data set[16] in comparison to Reweighing (Kamiran and Calders, 2011) – a state-of-the-art bias mitigation pre-processing method. Precisely, this section will answer the following research questions (Heidrich et al., 2023):

- **R2.3** Does the correction of explanations for fairness lead to fairer models?
- **R2.4** Does correcting explanations for fairness lead to fairer explanations?
- **R2.5** Does correcting for fair explanations have a negative impact on the predictive performance of the model?
- **R2.6** Which is superior, FAIRCAIPI or the state-of-the-art Reweighing strategy?

[16] `https://aif360.readthedocs.io/en/latest/modules/generated/aif360.datasets.GermanDataset.html`, 10 July 2024.

Auxiliary Methods

Compared to CAIPI (Teso and Kersting, 2019), FAIRCAIPI contains two modifications (Figure 4.7): First, the focus is shifted from optimizing a ML model from scratch to unlearning a bias present in a ML model without compromising its predictive performance. Second, FAIRCAIPI focuses on tabular data and reveals the impact of features causing a bias reproduction by SHAP (Lundberg and Lee, 2017). This section is structured accordingly and starts with auxiliary methods for bias detection. In this regard, it will also present Reweighing (Kamiran and Calders, 2011) as a baseline method for FAIRCAIPI. Afterwards, it will introduce the mathematics behind SHAP. Suppose the domain of Notation 2.1 for this section.

Bias detection and mitigation The protected attribute (Definition 4.8) refers to the feature which causes a biased decision of a ML model (Mehrabi et al., 2022; Chen et al., 2019). In this section, the protected attribute is the gender. Other examples are age, ethnicity, religion, or, generally, socio-demographic features. Protected attributes have a privileged and an unprivileged group. Privileged groups have systematic benefits from biased decision making and unprivileged groups are systematically disadvantaged. Note that this thesis uses the expressions privileged and unprivileged groups, and favored and deprived groups as synonyms.

Definition 4.8 (Protected Attribute (Mehrabi et al., 2022; Chen et al., 2019; Heidrich et al., 2023))**.** Let S be the feature identifier of the *protected attribute* with the values $x_S = s$ indicating the privileged and $x_S = \bar{s}$ the unprivileged group, respectively.

Outcomes of classification models can be favorable (Definition 4.9) or unfavorable (Bellamy et al., 2018). In the context of the running example, the favorable outcome is a low and the unfavorable outcome a high credit risk.

Definition 4.9 (Favorable Label (Bellamy et al., 2018; Heidrich et al., 2023))**.** Let $\hat{y} = d$ and $\hat{y} = \bar{d}$ denote the *favorable* and unfavorable *label* of a prediction $\hat{y} = f(x)$.

The combination of both dimensions results in four groups (Kamiran and Calders, 2011) – deprived/favored group with positive/negative label (Figure 4.5). Bias detection metrics (Table 4.4) compare the conditional distributions of receiving the (un)favorable label given that an instance belongs to the (un)privileged group or compare performance metrics across privileged and unprivileged groups.

Table 4.4: Bias detection metrics (Heidrich et al., 2023). The table contains **Equation**s of bias detection **Metric**s. False and true positive rates are abbreviated by fpr and tpr. The false discovery rate fdr is calculated by $fp(fp + tp)^{-1}$, where fp and tp indicate false and true positives. A procedure COMP with inputs f and S computes all bias detection metrics on the test data.

Metric	Equation
Statistical Parity (Dwork et al., 2012)	$SP = Pr(\hat{y} = \bar{d}\|S = s) - Pr(\hat{y} = \bar{d}\|S = \bar{s})$
Equalized Odds (Hardt, Price, and Srebro, 2016)	$EqOdds = \frac{1}{2}\left[(fpr_{S=\bar{s}} - fpr_{S=s}) + (tpr_{S=\bar{s}} - tpr_{S=s})\right]$
Equalized Opportunity (Hardt, Price, and Srebro, 2016)	$EqOpp = tpr_{S=\bar{s}} - tpr_{S=s}$
False Positive Error Rate Balance (Chouldechova, 2017)	$FPERB = fpr_{S=\bar{s}} - fpr_{S=s}$
Predictive Parity (Chouldechova, 2017)	$PP = fdr_{S=\bar{s}} - fdr_{S=s}$

Equalized Opportunity (Hardt, Price, and Srebro, 2016), the False Positive Error Rate Balance (Chouldechova, 2017), and Predictive Parity (Chouldechova, 2017) all rely on predictive performance metrics. The former focuses on the opportunity for an instance to receive the favorable label. The others target the bias-reproducing association with an unfavorable label. Equalized Odds (Hardt, Price, and Srebro, 2016) is slightly different and averages both objectives. Statistical Parity (Dwork et al., 2012) does unlike the others not require access to the ground truth and calculates the difference of the probabilities of receiving the unfavorable label between the privileged and unprivileged groups. All metrics have their optimum at zero, which expresses parity between privileged and unprivileged groups.

The baseline for FAIRCAIPI is Reweighing (Kamiran and Calders (2011), Definition 4.10). Reweighing is a pre-processing procedure which modifies the proportion of the protected attribute to achieve a Statistical Parity value close to or equal to zero.

Definition 4.10 (Reweighing (Kamiran and Calders, 2011; Heidrich et al., 2023))**.** *Reweighing* a-priori modifies the proportion of $\{x_i \text{ if } i = S | x \in \mathcal{X}\}$ such that it satisfies Statistical Parity (Table 4.4).

FAIRCAIPI calculates all bias detection metrics and presents the results to the user. Users can directly observe the consequences of their annotations on the fairness of the classification model. If bias detection metrics deviate from the optimum, FAIRCAIPI notifies users that their annotations enforce biases. This thesis uses the terms bias detection metrics and fairness metrics as synonyms. The detection of biases is a prerequisite to improve the fairness of classifiers.

SHAP SHapley Additive exPlanations (SHAP) (Lundberg and Lee, 2017) first simplifies the feature space by a transformation function. Afterwards, it approximates the model by a weighted sum of the simplified features. The weights are attribution values, which express the impact of each feature on the decision (Definition 4.11). The attribution values are quantified by eliminating features. Features with attribution values over a pre-defined threshold are said to be relevant.

Definition 4.11 (SHapley Additive exPlanations (SHAP) (Lundberg and Lee, 2017; Heidrich et al., 2023; Amling et al., 2024))**.** Let a model g approximate f by

$$f(x) = g(\tilde{x}) = \psi_0 + \bar{\phi}^T \tilde{x}$$

from a simplified feature space $\tilde{x}$ obtained from a transformation function $\overrightarrow{h}$ such that the reverse transformation yields $x = \overleftarrow{h}_x(\tilde{x})$. Let *SHAP* values be an attribution vector $\bar{\phi} = (\psi_1, ..., \psi_n)^T$ corresponding to $\tilde{x}$ with $\psi_0 = f(\overleftarrow{h}_x(\mathbf{0}))$. It is said that

$$exp = \{x_i \text{ if } \psi_i > \beta | i \in \mathcal{F}\} \subseteq x$$

is the subset of relevant features, where β is an attribution threshold. Suppose that $\tilde{z}$ are subsets of $\tilde{x}$ that has size M. Then, the attribution value of ψ_i is quantified by:

$$\psi_i(f, x) = \sum_{\tilde{z} \subseteq \tilde{x}} \frac{|\tilde{z}|!(M - |\tilde{z}| - 1)!}{M!} \left[f_x(\tilde{z}) - f_x(\tilde{z} \setminus \tilde{z}_i) \right],$$

where f_x is equal to f but has a varying input cardinality. Suppose for this section that EXP (Definition 2.4) takes f and x as input, applies SHAP, and returns exp.

SHAP has three properties (Lundberg and Lee, 2017; Heidrich et al., 2023): local accuracy as long as the transformation function is valid, which ensures faithful explanations; missingness, which states that zero, meaning missing, features have zero attribution; and consistency, which means that a stronger feature importance for the original model is also reflected in a higher attribution value. Proofs are given in Lundberg and Lee (2017), which build upon the work of Young (1985).

Adaptations

FAIRCAIPI has the objectives to (i) detect and (ii) mitigate biases of ML models and to (iii) inform users if their explanation revision enforces biases (Heidrich et al., 2023). Model decisions are said to be biased if either the favorable outcome is associated with the privileged or the unfavorable outcome co-occurs with the unprivileged group (Definition 4.12).

Definition 4.12 (Biased Decision Making (Heidrich et al., 2023))**.** Let $\hat{y} = f(x) \in \{d, \bar{d}\}$ be a favorable or an unfavorable prediction of a feature x with a protected attribute S and the deprived and favored groups s and $\bar{s}$. Let $exp \subseteq x$ be the relevant features for $f(x)$ (Definition 4.11). The *decision making* is said to be *biased* if

$$\hat{y} = \bar{d} \text{ and } \exists_{exp_i} exp_{i=S} = \bar{s} \text{ or } \hat{y} = d \text{ and } \exists_{exp_i} exp_{i=S} = s.$$

The bias mitigation strategy of FAIRCAIPI is embedded in its counterexample generation procedure (Algorithm 4.2). The general objective is to remove a bias-enforcing association between the protected attribute and the target. FAIRCAIPI's counterexample generator differentiates between favorable and unfavorable classification outcomes (line 2). It generates a proposal set for the protected attribute (line 3). Hereby, it removes the value belonging to the deprived group for unfavorable predictions and the value of the favored group for favorable predictions. It replaces the protected attribute with samples from the proposal set (line 5) and assigns the target (line 6) for the queried amount of counterexamples. The counterexample generator (Algorithm 4.2) is exclusively applied in this section and abbreviated by GEN.

FAIRCAIPI re-trains a classification model in each iteration. Before and after the training step, it computes bias detection metrics to inform users about the impact of their annotation (line 2). Afterwards, it selects the most-informative instance (line 3), which is predicted (line 4) and evaluated by a human annotator (line 5). If the prediction is wrong, the corrected most-informative instance is added to the labeled data set (line 7). Otherwise, a SHAP explanation reveals the decision-making mechanism (line 9). A user decides if the prediction is corrupted by biases (line 10). In the absence of biases, the most-informative instance is added to the labeled data set without further corrections (line 12). If biases have been identified, bias-mitigating counterexamples (line 14) are added additionally (line 15). Finally, the most-informative instance is removed from the unlabeled data set (line 16).

Apart from the counterexample generation procedure, FAIRCAIPI has three key differences compared to Algorithm 2.1: First, FAIRCAIPI uses SHAP as local explanation procedure. Second, users interact with the decision-making mechanism with the sole objective of mitigating biases. Third, FAIRCAIPI calculates bias detection metrics before and after fitting the model. This reveals the impact of user annotations on the fairness of the classification model and potentially educates users.

Algorithm 4.2: GEN(x, y, S, c) (FAIRCAIPI) (Heidrich et al., 2023)

Input: Feature x, label y, protected attribute S, number of counterexamples c
Output: Counterexample data sets $\mathcal{X}', \mathcal{Y}'$

1: $\mathcal{X}' \leftarrow \emptyset; \mathcal{Y}' \leftarrow \emptyset$
2: $s^* \leftarrow \bar{s}$ **if** $y = \bar{d}$ **else** $s^* \leftarrow s$
3: $s' \leftarrow \{x_{i=S} | x_i \in \mathcal{X}\} \setminus s^*$
4: **for** $1 : c$ **do**
5: $\quad x' \leftarrow \text{sample}(s')$ **if** $x_{i=S} = s^*$ **else** x_i **for** $x_i \in x$
6: $\quad \mathcal{X}' \leftarrow \mathcal{X}' \cup \{x'\}; \mathcal{Y}' \leftarrow \mathcal{Y}' \cup \{y\}$
7: **return** $\mathcal{X}', \mathcal{Y}'$

Algorithm 4.3: FAIRCAIPI($\mathcal{L}, \mathcal{U}, S, c, n$) (Heidrich et al., 2023)

Input: Data sets $\mathcal{L}$ and $\mathcal{U}$, protected attribute S, number of counterexamples c, iteration budget n
Output: Model f

for $1 : n$ **do**
  $f \leftarrow \text{FIT}(\mathcal{L})$ **and** $\text{COMP}(f, S)$ ▷ *Notation 2.1, Table 4.4*
    ▷ *Calculate fairness metrics before and after* FIT *and present to the user*
  $m \leftarrow \text{MII}(f, \mathcal{U})$ ▷ *Definition 2.1*
  $\hat{y}^{(m)} \leftarrow f(x_{\mathcal{U}}^{(m)})$
  $y^{(m)} \leftarrow \text{INTERACT}(x_{\mathcal{U}}^{(m)})$ ▷ *Definition 2.6*
  if $\hat{y}^{(m)} \neq y^{(m)}$ **then**
    $\mathcal{L} \leftarrow \mathcal{L} \cup \{(x_{\mathcal{U}}^{(m)}, y^{(m)})\}$ ▷ *Case:* ***W***
  else
    $exp \leftarrow \text{EXP}(f, x_{\mathcal{U}}^{(m)})$ ▷ *Definition 4.11*
    $biased \leftarrow \text{INTERACT}(exp)$ ▷ *Definitions 2.6 and 4.12*
    if not *biased* **then**
      $\mathcal{L} \leftarrow \mathcal{L} \cup \{(x_{\mathcal{U}}^{(m)}, \hat{y}^{(m)})\}$ ▷ *Case:* ***RRR***
    else
      $\mathcal{X}', \mathcal{Y}' \leftarrow \text{GEN}(x_{\mathcal{U}}^{(m)}, \hat{y}^{(m)}, S, c)$ ▷ *Algorithm 4.2*
      $\mathcal{L} \leftarrow \mathcal{L} \cup \{(x_{\mathcal{U}}^{(m)}, \hat{y}^{(m)})\} \cup \mathcal{X}' \times \mathcal{Y}'$ ▷ *Case:* ***RWR***
  $\mathcal{U} \leftarrow \mathcal{U} \setminus x_{\mathcal{U}}^{(m)}$
return f

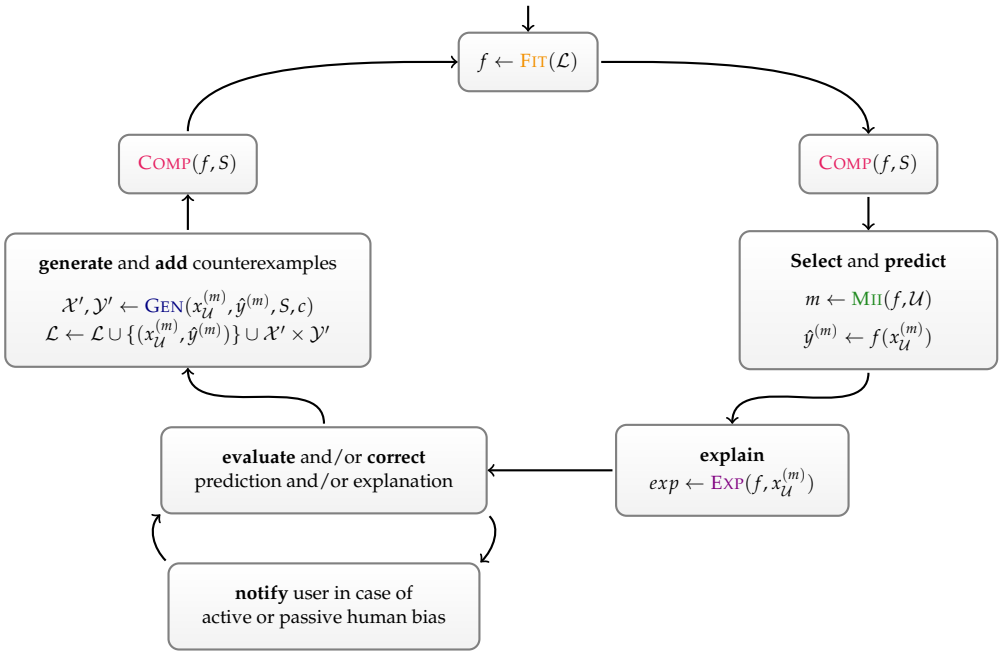

Figure 4.8: Graphical representation of FAIRCAIPI (Heidrich et al., 2023). FAIRCAIPI selects the most-informative instance. If the prediction is correct, a local explanation reveals potential biases. Counterexamples generated from user explanation revisions mitigate detected biases. FAIRCAIPI notifies users if their feedback has a negative impact on the fairness of the classification model.

Technically, as long as the knowledge about biased decision making in the sense of Definition 4.12 can automatically be inducted into the optimization cycle, FAIRCAIPI does not need a manual human annotation. Although the experiments in the next section will exploit such an automatic evaluation, biases might be more complex than in the comparatively simple scenario of gender biases in the credit lending setting. Moreover, FAIRCAIPI, being a model-agnostic XIML method, explicitly aims to involve human users during model optimization. Figure 4.8 visualizes how FAIRCAIPI (Algorithm 4.3) can be complemented such that users effectively mitigate model bias and FAIRCAIPI notifies human users if they induce biases. The distinction between active and passive bias is especially important. Passive bias refers to unintended human actions that induce bias. Active bias, on the contrary, refers to human actions that knowingly enforce the model bias. Therefore, FAIRCAIPI can help reveal bias but is only effective in mitigating the model and potentially even the human bias if human users act out of good intentions (Heidrich et al., 2023).

Supplementary to the publication (Heidrich et al., 2023), there exists a GitHub repository[17]. Both FAIRCAIPI variants are implemented: the autonomous execution setup suitable for reproducing experiments and the interactive mode, where human users can mitigate biased decision making of a pre-trained ML model.

A final remark on the time complexity of FAIRCAIPI (Heidrich et al., 2023): The time complexity of model-agnostic XIML methods in general is linear in terms of the alignment of CAIPI components. In this case, FAIRCAIPI integrates SHAP as an explanation framework. Generally, SHAP is intractable but can be approximated in polynomial time (Lundberg and Lee, 2017; Arenas et al., 2023). Hence, the CAIPI skeleton without specific operationalization of its components is computationally viable. If the integrated methods are computationally inefficient, CAIPI variants will inherit their time complexity and therefore also become computationally inefficient.

Experiments

Setup FAIRCAIPI is evaluated in its ability to mitigate the gender bias of a pre-trained random forest[18] on the Credit data set[19]. In this regard, 550 instances are treated as labeled and 150 as pseudo-unlabeled, meaning that the label can be queried during the FAIRCAIPI optimization. The remaining 300 instances are test data. This setup results in a comparatively high starting predictive performance of 75 percent accuracy. At this point, it is important to have a well-performing model to determine whether the occlusion of the relation between the protected attribute and the target compromises the predictive performance.

The a-priori predictive performance is higher than reported by Table 4.1. This has several reasons: First, the included features are different. Second, the evaluation metrics are different. Whereas this section reports the accuracy and precision and recall per class, the former section has calculated the weighted average. Third and most importantly, the pre-processing pipeline is optimized for the Credit data set. This means, for instance, clustering, ceiling, and min max scaling numerical features.

The experiment is executed once with 100 FAIRCAIPI optimization iterations. The sex feature is the only protected attribute. Whether a decision-making mechanism is biased or not, is evaluated by Definition 4.12 with an attribution threshold of

[17] `https://github.com/emanuelsla/faircaipi`, 09 July 2024.

[18] `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`, 03 July 2024.

[19] `https://aif360.readthedocs.io/en/latest/modules/generated/aif360.datasets.GermanDataset.html`, 10 July 2024.

0.005. Reweighing (Kamiran and Calders (2011), Definition 4.10) serves as a benchmark method. The experiment can be reproduced by the GitHub repository[20].

Results In terms of the predictive performance, FAIRCAIPI even improves the baseline model and outperforms the bias mitigation strategy Reweighing (Table 4.5). Figure 4.9 visualizes that FAIRCAIPI's predictive performance is constant during its optimization process. This finding is noteworthy because FAIRCAIPI unlearns relevant but indecisive correlations. Decreasing the number of correlations within a ML model is expected to cause a reduced predictive performance.

The bias detection metrics in Table 4.6 have their optimum at zero. FAIRCAIPI is superior for each metric compared to Reweighing. The only exception is Statistical Parity, which is the optimization constraint of Reweighing. Remarkable is also that the final FAIRCAIPI model, which is not necessarily the optimum, is superior to Reweighing. Moreover, Reweighing seldomly improves the baseline model. The results suggest that FAIRCAIPI makes models fairer, whereas Reweighing only improves Statistical Parity, which is not a general fairness improvement.

Table 4.5: FAIRCAIPI predictive performance results (Heidrich et al., 2023). Comparison of a **Default** random forest without fairness optimization to **FAIRCAIPI** and **Reweighing** (Kamiran and Calders, 2011) for various predictive performance **Metric**s. Precision, recall, and F1-score are evaluated on each **Subset** of the label. Superior results are written boldly.

Metric	Subset	Default	FAIRCAIPI	Reweighing
Accuracy	-	0.73	**0.76**	0.72
Precision	good risk	0.76	**0.77**	0.75
	bad risk	0.60	**0.68**	0.58
Recall	good risk	0.89	**0.92**	0.89
	bad risk	0.38	**0.39**	0.34
F1-score	good risk	0.82	**0.84**	0.82
	bad risk	0.47	**0.5**	0.43

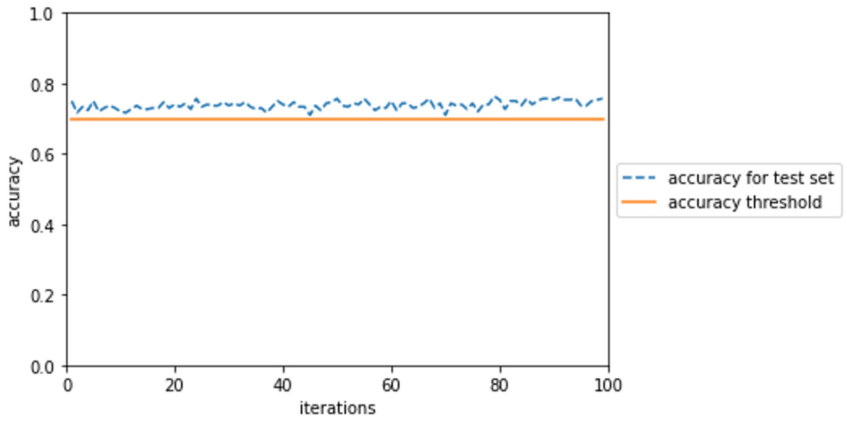

Figure 4.9: Predictive performance during FAIRCAIPI optimization (Heidrich et al., 2023). Accuracy assessment (dashed blue line) over the course of 100 FAIRCAIPI iterations. The mandatory lower bound for the accuracy lies at 70 percent (orange line).

[20] `https://github.com/emanuelsla/faircaipi`, 09 July 2024.

Table 4.6: FAIRCAIPI fairness results (Heidrich et al., 2023). Comparison of bias **Metrics** (Table 4.4). **Default** refers to the random forest without fairness optimization. **FAIRCAIPI** after 100 optimization iterations and its optimal value (**FAIRCAIPI (opt.)**) are compared to **Reweighing** (Kamiran and Calders, 2011). Superior results are written as bold numbers.

Metric	Default	FAIRCAIPI	FAIRCAIPI (opt.)	Reweighing
Statistical Parity	-0.0886	-0.0447	-0.0391	**0.0000**
Equalized Odds	-0.1568	-0.0909	**-0.0038**	-0.1819
Equalized Opportunity	-0.0514	-0.0295	**-0.0007**	-0.1237
FPERB	-0.2622	-0.1524	**0.0038**	-0.2401
Predictive Parity	-0.0322	-0.0026	**0.0008**	-0.0053

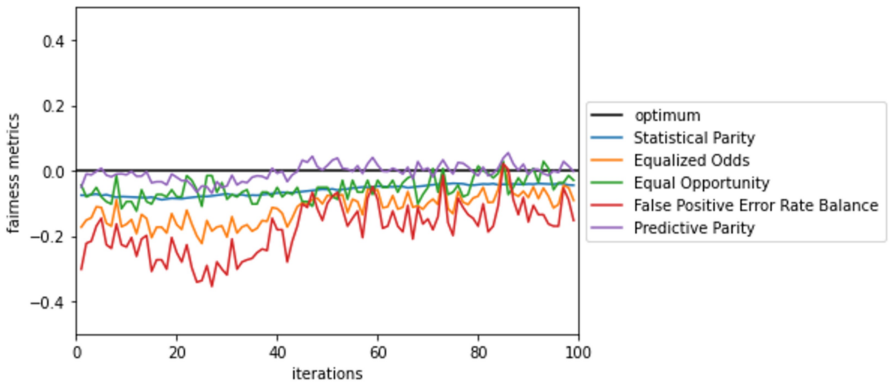

Figure 4.10: FAIRCAIPI optimization for fairness metrics (Heidrich et al., 2023). Comparison of bias detection metrics (Table 4.4) over the course of 100 FAIRCAIPI iterations. The optimal value of each metric is zero (black line).

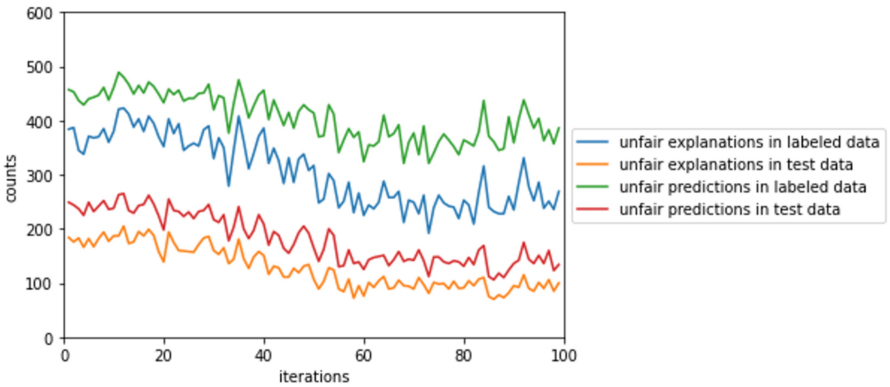

Figure 4.11: Fairness of predictions and explanations in FAIRCAIPI optimization (Heidrich et al., 2023). Number of unfair predictions and explanations in each of the 100 FAIRCAIPI optimization iterations on labeled and test data.

Figures 4.10 and 4.11 depict that the bias detection metrics improve and the amount of unfair predictions and explanations decrease over the course of the FAIRCAIPI optimization. The improvements mainly arise around the optimization iterations 40 to 60 with comparatively stable results before and after. Interestingly, the number of unfair predictions is strictly higher than the number of unfair explanations. This is the case because unfair explanations always cause unfair predictions. Contrary, predictions can be unfair – e.g., if an instance of the deprived group receives an unfavorable label –, even if the decision-making mechanism is unbiased.

Section Summary

Summary FAIRCAIPI (Algorithm 4.3) is a bias mitigation in-processing method based on model-agnostic XIML. It modifies CAIPI (Teso and Kersting, 2019) in the sense that it uses SHAP explanations to evaluate whether the underlying decision-making mechanism of an instance contains biases (Figure 4.7). If a bias in the decision-making mechanism is detected, counterexamples outweigh the instance's correlation between the protected attribute and the predictive outcome (Algorithm 4.2). FAIRCAIPI can either train models from scratch or fine-tune models to mitigate learned biases.

The experiments indicate that FAIRCAIPI improves the fairness of ML models (Figures 4.10 and 4.11). Hereby, FAIRCAIPI is also superior to the state-of-the-art bias mitigation pre-processing strategy Reweighing (Kamiran and Calders (2011), Definition 4.10, Table 4.6). Whilst retaining a stable predictive performance over the optimization process (Figure 4.9) and not being the primary optimization objective, FAIRCAIPI tends to improve the model's predictive performance (Table 4.5).

Despite FAIRCAIPI is evaluated automatically in a simulation study, it can also be operated interactively and thus iteratively involve human users (Figure 4.8). In addition to prior presented user interfaces such as Figure 3.3, FAIRCAIPI presents fairness metrics to users. With FAIRCAIPI, users have an awareness of how their annotations affect the fairness of the classification model. In summary, FAIRCAIPI is a tool that (i) detects and (ii) mitigates machine bias and (iii) reveals human bias. FAIRCAIPI can potentially be used to educate users to mitigate human bias, which is beyond the scope of this thesis and left for future work.

Answers to subordinated research questions Based on the experimental results, the research questions can be answered as follows (Heidrich et al., 2023):

R2.3 Does the correction of explanations for fairness lead to fairer models?
Yes, for a random forest pre-trained on the Credit data set, FAIRCAIPI *optimization improves the investigated fairness metrics with results close to the metrics' optimum.*

R2.4 Does correcting explanations for fairness lead to fairer explanations?
Yes, the experimental results show that FAIRCAIPI *reduces the number of unfair explanations. This means that the decision-making mechanism is less likely to be corrupted by the gender bias of the Credit data set.*

R2.5 Does correcting for fair explanations have a negative impact on the predictive performance of the model?
No, the results indicate a stable predictive performance over the course of the FAIRCAIPI *optimization.* FAIRCAIPI *even tends to improve the predictive performance compared to the baseline model and a model trained on data pre-processed by Reweighing.*

R2.6 Which is superior, FAIRCAIPI or the state-of-the-art Reweighing strategy? FAIRCAIPI *is superior to Reweighing within the scope of the experiments. Overall,* FAIRCAIPI *outperforms Reweighing in improving the model's fairness and in terms of the predictive quality.*

Related results FAIRCAIPI is a bias mitigation in-processing method similar to Nakao et al. (2022). It has been evaluated in comparison to a state-of-the-art bias mitigation pre-processing procedure (Kamiran and Calders, 2011) and achieved superior results regarding fairness and predictive performance. Furthermore, FAIRCAIPI is a CAIPI variant (Slany, Scheele, and Schmid, 2024a; Slany, Scheele, and Schmid, 2024c; Slany, Scheele, and Schmid, 2024b) that alters both the explanation and counterexample generation framework compared to the traditional algorithm (Teso and Kersting, 2019). The closest related experimental setup is the one of Slany, Scheele, and Schmid (2024b), which also induces a single counterexample per **RWR** iteration – in this section, an iteration with biased decision making. The results do not compare well for several reasons: First, the data set and the pre-processing pipeline differ. Second, FAIRCAIPI fine-tunes a pre-trained model, whereas Slany, Scheele, and Schmid (2024b) train a model from scratch. Third, the local explainers are different. FAIRCAIPI uses SHAP (Lundberg and Lee, 2017), whereas Slany, Scheele, and Schmid (2024b) rely on counterfactual explanations (Wachter, Mittelstadt, and Russell, 2017). Fourth, the optimization objective is shifted from predictive and explanatory performance improvements to improve the fairness of the classification model, which also conveys into the construction of counterexamples.

Limitations It has been shown that FAIRCAIPI is a viable approach to optimize classification models on the Credit data set[21] with the gender feature as a single protected attribute for fairness. The restrictions to classification models and a specific data set with a single protected attribute are the major limitation of FAIRCAIPI. This limitation can be further broken down into the categories algorithmic formalization and experimental setup:

- **Algorithmic formalization**:
  FAIRCAIPI (Algorithm 4.3) has one algorithmic and one conceptual drawback: Algorithmically, FAIRCAIPI is formalized such that it can consider a single protected attribute. Although the extension to multiple uncorrelated protected attributes seems straightforward, FAIRCAIPI leaves open whether biases consisting of correlated features can be detected and mitigated. On the conceptual level, user knowledge about biased decision making is mandatory. Bias mitigation with FAIRCAIPI is only possible if users are able to identify biases in the decision-making mechanism. Apart from that, FAIRCAIPI leaves open how it can be transferred to other ML tasks and data types.
- **Experimental setup**:
  The experimental evaluation is sparse in direct comparison to the model-agnostic XIML methods presented in the previous sections. The experiments are only executed once such that no standard deviations can be reported. FAIRCAIPI is compared against a bias mitigation pre-processing method (Kamiran and Calders (2011), Definition 4.10), although it is classified as in-processing method. The single benchmark, moreover, is not even strictly superior to the default model without bias mitigation (Table 4.6). In general, the experiments should be extended: More data sets and benchmark methods

[21] `https://aif360.readthedocs.io/en/latest/modules/generated/aif360.datasets.GermanDataset.html`, 10 July 2024.

should be included as well as the amount of counterexamples should be evaluated. At some points, FAIRCAIPI claims to uncover and potentially even reduce human bias, but does not present necessary psychological background concepts that support the hypothesis.

4.2 Image-specific Adaptations

This section leaves the terrain of tabular data and proposes two CAIPI modifications for image data. Remember that image data has been the primary application domain of CAIPI (Teso and Kersting, 2019). The Introduction (Chapter 1) has found that one limitation of CAIPI is the absence of a user interface, which end-users can operate for the image classification task. The next section closes this gap and will propose a practical user interface for image data (Slany et al., 2022). Moreover, it incorporates a counterexample generation procedure by data augmentation and actively involves humans in the experiments. The second CAIPI modification aims to improve the counterexample generation based on data augmentation (Slany, Scheele, and Schmid, 2024a). Similar to what Section 4.1.1 has done for tabular data, this section will use a statistical generative model – precisely, a Variational Autoencoder (Kingma and Ba, 2015) – to generate counterexamples.

4.2.1 User Interaction and Data Augmentation

The majority of this section has already been published in Slany et al. (2022) with slight modifications. This also includes definitions, figures, and tables, which will be referenced accordingly. Some mathematical notation is part of Slany, Scheele, and Schmid (2024a). Concepts taken from the latter publication will be referenced as such. By adopting the notation of Slany, Scheele, and Schmid (2024a), this section preempts the following one. A unified notation for model-agnostic XIML on image data facilitates the mathematical formalization of the proposed modifications. The Appendices C.3 and C.4 state the author's contribution to Slany et al. (2022) and to Slany, Scheele, and Schmid (2024a), respectively.

Originally, CAIPI (Teso and Kersting, 2019) has been evaluated on the FashionMNIST data set[22], where decoy pixels have been inducted into the images such that the color corresponds to the class label. CAIPI has been evaluated in its ability to unlearn the artificially added spurious correlation. Despite this experimental setup has been adopted by some of the proposed CAIPI variants of this thesis, e.g., Section 4.1.1, it leaves the human out of the equation, which is problematic in the sense that CAIPI claims to be interactively optimizable by human users (Teso and Kersting, 2019).

Problem Despite CAIPI (Teso and Kersting, 2019) puts forward to be an interactive method, it does not contain a user interface for the investigated image classification task. Moreover, the randomization of induced decoy pixels to generate counterexamples does not transfer to real-world classification scenarios.

Solution This section proposes a user interface for CAIPI (Teso and Kersting, 2019). It also specifies a data augmentation procedure that can be applied to binary image classification tasks, where humans are able to identify the decisive features.

[22] `https://github.com/zalandoresearch/fashion-mnist`, 22 May 2024.

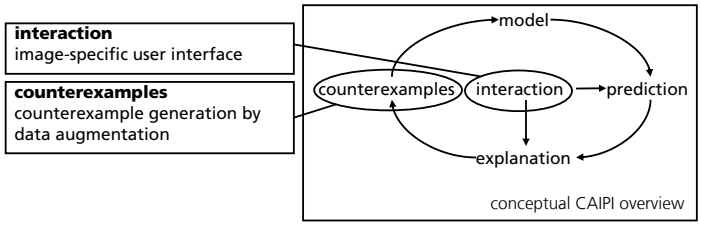

Figure 4.12: Practical CAIPI component adaptations.

Contribution This section contributes (i) a image-specific CAIPI user interface along with (ii) a data augmentation procedure to optimize models with CAIPI within the scope of binary image classification tasks. Using both contributions as prerequisites, this section (iii) contains the first evaluation of CAIPI's algorithmic behavior with label and explanation corrections retrieved from human annotators. It furthermore (iv) clarifies the question of whether additional explanation corrections for wrong predictions are superior to solely correcting their label.

This section re-implements the traditional CAIPI framework within its original, primary application avenue – image classification (Teso and Kersting, 2019). Apart from a mathematical specification of how concepts such as LIME (Ribeiro, Singh, and Guestrin, 2016) are leveraged for CAIPI, it puts CAIPI into practice, meaning that the interaction and counterexample generation components are transformed such that CAIPI can be operated by end-users for any binary image classification task. Thus, a user interface for CAIPI and a fairly simple data augmentation procedure to generate counterexamples from user feedback are the only adaptations compared to the original publication (Figure 4.12, Teso and Kersting (2019)).

Subordinated research questions The reimplementation goal also conveys into the research questions. Compared to Teso and Kersting (2019), the research questions targeting the algorithmic convergence are extended by the question of explanation corrections for wrong predictions and a baseline study (Slany et al., 2022).

R2.7 Do explanation revisions improve the predictive quality?
R2.8 Do explanation revisions lead to an improved explanatory quality?
R2.9 Does the predictive and explanatory quality benefit from explanation revisions for wrong predictions?
R2.10 Which is superior, CAIPI or default deep learning?

Auxiliary Methods

This is the first section that does not contain a mathematical investigation nor an algorithmic modification of CAIPI (Teso and Kersting, 2019). Instead, this section proposes a CAIPI user interface. Nevertheless, it is necessary to introduce some foundational concepts. CAIPI now operates on image data requiring the following notational adjustment (Notation 4.1). Users are enabled to generate counterexamples from LIME (Ribeiro, Singh, and Guestrin, 2016) explanation revisions, which are augmented versions of the decisive features (Definition 4.13).

Notation 4.1 (Image Classification (Slany, Scheele, and Schmid, 2024a))**.** Let $X \in \mathcal{X} \subseteq \mathbb{R}^{W \times H}$ be an image matrix with width W and height H. Let X_i denote a pixel with identifiers $\mathcal{F} = \{1, ..., i, ..., WH\}$. Let $f : \mathcal{X} \rightarrow \mathcal{Y}$ be a binary differentiable classifier with target set $\mathcal{Y} = \{0, 1\}$. An inference is defined as $y = f(X)$. Let $(X_{\mathcal{L}}^{(n)}, y_{\mathcal{L}}^{(n)}) \in \mathcal{L} \subseteq \mathcal{X} \times \mathcal{Y}$ denote the n-th instance of a labeled and $X_{\mathcal{U}}^{(n)} \in \mathcal{U} \subseteq \mathcal{X}$ the n-th instance of an unlabeled subset. Let $l : \mathcal{X} \rightarrow \mathcal{Y}$ be a labeling function.

Definition 4.13 (Decisive Image Features (Slany, Scheele, and Schmid, 2024a))**.** Let $D \in \{0, 1\}^{W \times H}$ ($\{\text{indecisive}, \text{decisive}\}$) be a binary mask of *decisive features* of an *image* X that cause a decision $y = l(X)$ either individually or in combination such that $X_D = D \cdot X$ blacks out indecisive features.

LIME and the data augmentation pipeline will be derived in distinct paragraphs. Both concepts will be exploited for the user interface proposed in the next section.

LIME Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro, Singh, and Guestrin, 2016) simplifies the feature space by projecting the original feature instance into binary super-pixels, which represent the presence or absence of similar pixels in an area. Typically, this step is conducted by segmentation algorithms (e.g., Schallner et al., 2019). Afterwards, multiple samples of the super-pixel instance formulate an auxiliary data set, which is used as input for the explanatory model. The instances are preferably sampled locally wrt. the original instance, which is ensured by the euclidean distance in the exponential kernel. The optimal explanatory model, according to LIME, is the one that minimizes the distance to the original model and has a low complexity – for instance, a small amount of weights in the case of a linear explanation model (Definition 4.14).

Definition 4.14 (Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro, Singh, and Guestrin, 2016; Slany et al., 2022))**.** Let *LIME* search for the optimal explanation model $g \in G$ for f wrt. X based on a combined loss function consisting of an approximation and a complexity term:

$$\xi(X) = \arg\min_{g \in G} L(f, g, \pi_X) + \Omega(g).$$

Approximation:
Let $\tilde{X} \in \{0, 1\}^{W \times H}$ be a binary representation of X, indicating the absence or presence of super-pixels. Let $(\tilde{Z} \sim \tilde{X}) \in \tilde{\mathcal{Z}}$ be a sample of super-pixels aggregated to a set. Let the approximation loss between g and f be defined as

$$L(f, g, \pi_X) = \sum_{\tilde{Z} \in \tilde{\mathcal{Z}}} \pi_X(Z)(f(Z) - g(\tilde{Z}))^2 \text{ with}$$

$$\pi_X(Z) = \exp(-\|X, Z\|_2^2 \sigma^{-2}),$$

where $Z = \overleftarrow{h}_X(\tilde{Z})$ is a reverse transformation function such that $Z \in \mathbb{R}^{W \times H}$ in the sense of reversing the transformation $\tilde{X} = \overrightarrow{h}(X)$ and σ is the width of $\|X, Z\|_2$.

Complexity:
Let $\Omega(g)$ be a measure for the complexity of g. Assuming that g is a linear model, $\Omega(g)$ is the number of adaptable weights.

Remark 4.4. Note that similar to Definition 4.11, g and $\overrightarrow{h}$ in Definition 4.14 denote the approximation and transformation functions. Each definition parameterizes the functions differently, as they belong to distinct notational domains.

LIME claims to produce faithful explanations given a feature instance, as it reverses the simplification of the sampled super-pixel set to query the label of super-pixel instances directly from the classification model (Ribeiro, Singh, and Guestrin, 2016). The variance of relevance areas attributed by LIME has been shown to be comparatively high when using different image segmentation algorithms (Schallner et al., 2019). This jeopardizes the faithfulness assumption. Yet, this section still uses LIME as a local explainer, as it is also present in the traditional CAIPI framework (Teso and Kersting, 2019).

Data augmentation Counterexamples in CAIPI are supposed to overweigh the correlation of the decisive features and the target. Basically, this can be seen as increasing the weights of a subset of features for a specific instance. A way to accomplish such a local weight adaptation is by repeating the subset of features multiple times. Copying the instance as is potentially results in problems such as overfitting. Data augmentation is an appropriate alternative, as it replicates yet randomly alters the decisive features. The random modification potentially increases the robustness of the classifier against the position or angle of the decisive features. This might be especially beneficial in early CAIPI iterations or for a weakly pre-trained model.

CAIPI's data augmentation procedure is visualized in Figure 4.13. Essentially, a user selects the decisive features for the prediction in **RWR** cases, which are then re-scaled, rotated, and translated. The data augmentation procedure is executed multiple times corresponding to the queried amount of counterexamples with an identical set of decisive features. Each output of the data augmentation pipeline is a counterexample feature instance. The counterexample target set is the repeated prediction of the original instance (Definition 4.15).

Definition 4.15 (Counterexamples by Data Augmentation)**.** Let the *counterexample* feature and target sets *by* means of *data augmentation* be defined as:

$$(\mathcal{X}' = \{\tau(D \cdot X)|1 : c\}) \times (\mathcal{Y}' = \{f(X)|1 : c\}),$$

where τ augments an image with operations corresponding to the data augmentation pipeline (Figure 4.13), D is the matrix of decisive features (Definition 4.13), and c is the number of counterexamples.

Remark 4.5. Practically, the augmentation operations in τ depicted by Figure 4.13 are taken from Buslaev et al. (2020).

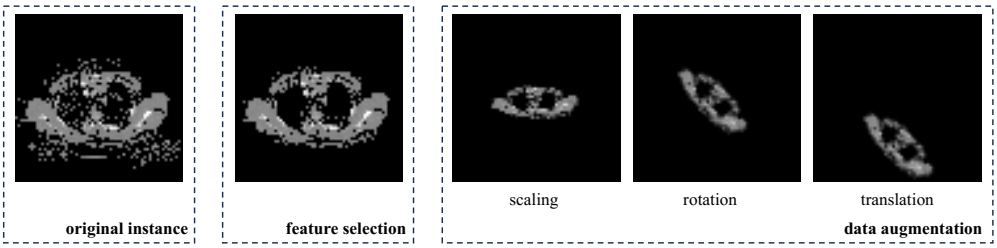

Figure 4.13: Data augmentation (Slany et al., 2022). A **data augmentation** procedure scales, rotates, and translates a subset of pixels of an **original instance** after the **feature selection**.

Adaptations

Suppose that a binary classifier distinguishes computer tomography (CT) images of the chest from CT images of the abdomen[23]. In each iteration, the CAIPI user interface (Figure 4.14, top left) presents the most-informative feature instance and the classifier's prediction to the user along with the possible XIML outcome cases (Table 2.1). First, the user assesses the correctness of the classifier's prediction. In this case, the depicted instance is correctly predicted to belong to the class of chest CT images. The `Explanation` button reveals the underlying reasons for the prediction by means of a LIME explanation (top right). Here, the user argues that the explanation is correct but is (partly) based on erroneous reasons as LIME (Definition 4.14, Ribeiro, Singh, and Guestrin (2016)) partly attributes the background to be decisive for the prediction – an **RWR** case. By pressing the `True(WR)` button, the user has the opportunity to revise the explanation (bottom left) by annotating the decisive pixels (Definition 4.13). The annotated image can be visualized in a preview mode (bottom right). It serves as an input for the data augmentation procedure (Figure 4.13) to generate counterexamples (Definition 4.15). The presented instance is added with its counterexamples to the labeled data set.

Accordingly, instances are added to the labeled data set without modifications in the **RRR** case (button `True(RR)`) and with corrected label in the in the **W** case (button `False(W)`). In the background, the currently presented most-informative instance is removed from the unlabeled data set.

By the proposed CAIPI frontend (Figure 4.14), users without ML expertise can optimize binary image classification models with CAIPI (Teso and Kersting, 2019). The subsequent experimental section contains evidence for models optimized with user interface annotations.

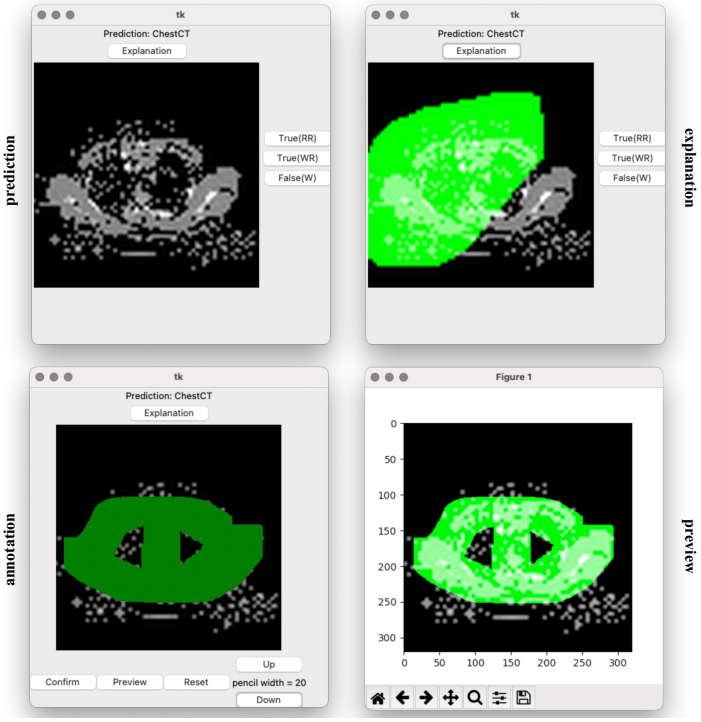

Figure 4.14: Image-specific user interface (Slany et al., 2022). The CAIPI user interface displays the model's **prediction** and **explanation** to a user. The user **annotation** revising the explanation is visualized in a **preview** mode.

[23] `https://www.kaggle.com/datasets/andrewmvd/medical-mnist`, 15 July 2024.

Experiments

Setup This section provides experimental evidence for optimizing binary classification models on the FashionMNIST (T-shirt/top versus pullover) and the MedicalMNIST (chest versus abdomen) data sets[24]. Whereas the FashionMNIST has also been used by Teso and Kersting (2019), the MedicalMNIST extends CAIPI's application spectrum. All data sets (labeled, unlabeled, and test sets) are balanced. Each experiment is only executed once.

Each optimization cycle starts with 100 labeled instances and continues for 100 CAIPI optimization iterations. Similarly to Teso and Kersting (2019), the experiments compare the impact of zero, one, three, and five counterexamples per **RWR** iterations, where the zero counterexamples case can be seen as coactive learning (Shivaswamy and Joachims, 2015). Additionally to the original evaluation, each experiment is repeated with the modification that also the explanations of wrong predictions are revised such that counterexamples are generated in **RWR** *and* **W** cases.

All experiments optimize a convolutional neural network. It consists of a single convolutional layer with two filters, 9×9 kernel size, and stride parameter one; a pooling layer with kernel size 8×8 and stride parameter eight; a linear layer with 98 neurons; a dropout layer with a dropout rate of 0.5; and two dense layers with 16 and a single neuron. The model is re-trained in each CAIPI iteration with a batch size of 64 for five epochs. The neural network is optimized by Adam with the binary cross-entropy loss and a 0.0001 learning rate. It has been implemented in PyTorch[25]. The baseline exploits the identical settings but conducts a 70/30 train test split.

The optimization is conducted by human annotation by the aid of the CAIPI user interface (Figure 4.14). The label corrections are equal to the ground truth labels, which is possible as both unlabeled data sets are only pseudo-unlabeled. The explanation revisions are subjective annotations of two human annotators, who have been instructed to annotate the entire object without fragments as being decisive.

The predictive performance is quantified by the accuracy metric on 30 percent of the instances that serve as test data. The ability to follow the correct decision-making mechanism, or simply, the explanatory performance, is computed by the intersection over union metric (Definition 4.16) on 200 randomly selected pre-annotated test instances. It divides the intersection of the revised and the LIME explanation by their union (Definition 4.14, Ribeiro, Singh, and Guestrin (2016)).

Definition 4.16 (Intersection over Union (IoU) (Rezatofighi et al., 2019))**.** Suppose two bounding boxes A and B with arbitrary dimensions within the dimensionality of X. Their IoU is defined as:

$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

Results Overall, the results do not indicate that the induction of more counterexamples is superior, not even that counterexamples in general are beneficial compared to coactive learning (Shivaswamy and Joachims, 2015), in terms of the predictive (Table 4.7) or the explanatory performance (Table 4.8). Also, no clear trend exists on whether additional explanation revisions for wrong predictions – overall more counterexamples – are superior or perhaps inferior compared to the traditional CAIPI setup with counterexamples in **RWR** iterations (Teso and Kersting, 2019).

[24] FashionMNIST: `https://github.com/zalandoresearch/fashion-mnist`, MedicalMNIST: `https://www.kaggle.com/datasets/andrewmvd/medical-mnist`, 15 July 2024.

[25] `https://pytorch.org/docs/stable/index.html`, 15 July 2024

Table 4.7: CAIPI predictive performance (Slany et al., 2022). Accuracy obtained by CAIPI optimization conditioned on **Data** sets, **Mode**s and c counterexamples.

Data	**Mode**	c:	0	1	3	5
MedicalMNIST	RWR		96.02	95.42	94.51	96.75
	RWR + W		96.83	97.48	96.92	**97.52**
FashionMNIST	RWR		**95.64**	94.79	95.40	94.95
	RWR + W		94.33	95.02	94.24	94.10

Table 4.8: CAIPI explanatory performance (Slany et al., 2022). Average *IoU* (Definition 4.16) between LIME and user explanation wrt. **Data** sets, **Mode**s and c counterexamples.

Data	**Mode**	c:	0	1	3	5
MedicalMNIST	RWR		41.59	**44.74**	40.36	42.87
	RWR + W		39.64	42.37	41.12	40.47
FashionMNIST	RWR		65.24	64.41	64.40	65.48
	RWR + W		64.43	65.76	63.10	**66.38**

The predictive performance results are constantly high and in line with the baseline: 94.67 percent accuracy on the MedicalMNIST and 95.26 percent accuracy on the FashionMNIST. The explanatory performance tends to be stable on a fairly high level. The baseline explanatory performance has not been assessed by Slany et al. (2022). Still interesting is the finding that both coactive learning (Shivaswamy and Joachims, 2015) and CAIPI (Teso and Kersting, 2019) reduce the labeling effort dramatically. It requires only 200 annotated instances compared to thousands of instances on the FashionMNIST and the MedicalMNIST. Yet, explanation revisions are additional and possibly even more costly annotations.

Section Summary

Summary This section has proposed a user interface for CAIPI (Figure 4.14). It aims to involve human annotators in the optimization cycle actively and hides the technical details of ML optimization. It re-implements CAIPI (Teso and Kersting, 2019) in the sense that it optimizes a comparatively shallow convolutional neural network on the FashionMNIST data set. By including additional results on the MedicalMNIST, it extends CAIPI's application areas. To transfer the optimization framework across domains, this section proposes a counterexample generation procedure by an augmentation of decisive features (Figure 4.13). Teso and Kersting (2019) put forward that more counterexamples are beneficial. Therefore, this section asks whether additional counterexamples obtained by explanation revisions for wrong predictions are superior to generating counterexamples solely in **RWR** cases.

The experimental results, however, are discouraging. Although CAIPI optimization achieves a similar predictive performance while reducing the amount of training data remarkably, CAIPI is not strictly superior to coactive learning (Shivaswamy and Joachims, 2015) in the investigated settings in terms of the predictive (Table 4.7) and the explanatory quality (Table 4.8). This implies that more counterexamples have no beneficial impact – not when increasing the amount of counterexamples in **RWR** iterations and not when generating counterexamples additionally for **W** cases.

Answers to subordinated research questions The following answers to the research questions (Slany et al., 2022) can be deduced from the experimental results:

R2.7 Do explanation revisions improve the predictive quality?
No, counterexamples from explanation revisions do not improve the predictive quality within the context of the experimental setting.

R2.8 Do explanation revisions lead to an improved explanatory quality?
No, counterexamples generated from revised explanations do not enhance the model's ability to follow the correct decision-making mechanism.

R2.9 Does the predictive and explanatory quality benefit from explanation revisions for wrong predictions?
There exists no clear trend whether additional counterexamples from explanation revisions for wrong predictions are superior or inferior.

R2.10 Which is superior, CAIPI or default deep learning?
The predictive performance of both optimization strategies is comparatively similar. CAIPI *optimization reduces the amount of training data. Hence,* CAIPI *can be seen as superior compared to default deep learning.*

Related results This section aims to re-implement the components motivated by the original CAIPI framework (Teso and Kersting, 2019). Different from Teso and Kersting (2019) is the experimental setup, where humans actively revise explanations instead of an automatic mitigation of induced spurious correlations. This experimental setup, in fact, is unique in the model-agnostic XIML literature (Table 1.1). In general, CAIPI (Teso and Kersting, 2019) and all of its variants (Heidrich et al., 2023; Slany, Scheele, and Schmid, 2024a; Slany, Scheele, and Schmid, 2024c; Slany, Scheele, and Schmid, 2024b) benefit from including counterexamples – a finding that this section could not reproduce. A reproducible finding is that models optimized with CAIPI at least match the baseline performance (Teso and Kersting, 2019; Heidrich et al., 2023; Slany, Scheele, and Schmid, 2024a; Slany, Scheele, and Schmid, 2024b).

Limitations Frequently, experiments that do not result in the expected outcome are connected to various limitations. In this case, they can be clustered into problems with human annotations and counterexamples for images as well as conceptual mistakes of the experimental setup.

- **Human annotations**:
  There have been 16 optimization runs (four counterexample variations times two data sets times two modes) with 100 optimization iterations each, which results in $1,600$ human annotations. The decisive features of 200 instances have been annotated on each of the two test data sets. In total, $2,000$ human annotations have been conducted. They have been evenly split across two labelers. A small sample of instances has been annotated by both labelers to ensure a inter-labeler consistency. This has not been quantified and might therefore be lower than expected in the qualitative assessment.
- **Image counterexamples**:
  Counterexamples generated by data augmentation for image data are a modified subset of pixels. Without a conceptual understanding associated to the pixels, counterexamples might not be beneficial for learning the correct decision-making mechanism. In other words, if a pattern such as a T-shirt does not occur similarly in novel images, the counterexamples' effect might

diminish, as image counterexamples do not contain the knowledge that a T-shirt has a torso piece and sleeves. Even if one can expect that a T-shirt will re-occur in the FashionMNIST example, in more complex cases such as the identification of cancer tissue, the identical pattern is more unlikely to be present twice or more frequently. Moreover, the user interface (Figure 4.14) enables humans to annotate positive counterexamples – what features qualify a T-shirt to belong to the T-shirt class. Human explanations for a T-shirt might be negative: An image belongs to the T-shirt class because it has no long sleeves. Such explanations cannot be annotated with the user interface.

- **Experimental setup**:
  The intersection over union metric (Definition 4.16) might be unsuitable to assess the correctness of the decision-making mechanism. In some cases, all features besides the object have been identified as relevant. In the sense as the annotators have been instructed to mark decisive features, this is an perfect inverted explanation. The intersection over union metric, however, reports an explanation quality of zero. A more sophisticated alternative is the generalized intersection over union metric (Rezatofighi et al., 2019), which still would not be able to account for the preliminary problem. LIME (Ribeiro, Singh, and Guestrin, 2016) has been proven to have faithfulness problems when using different image segmentation algorithms (Schallner et al., 2019). It is also worth improving. LIME has been used as it is also the local explanation algorithm implemented in traditional CAIPI (Teso and Kersting, 2019). In general, the experimental setup appears to be too simple as even coactive learning (Shivaswamy and Joachims, 2015) matches the baseline performance, which barely leaves room for improvement when inducing counterexamples. The experiments have been reduced from a categorical to a binary classification in comparison to Teso and Kersting (2019). The experimental complexity has been sacrificed in favor of an intuitive and easy-to-implement user interface. Also, the experiments should have been repeated multiple times with varying random seeds to compute standard deviations, which has not been done due to the high labeling effort.

4.2.2 Counterexamples by Variational Autoencoders

A large proportion of the contents of this section is taken from Slany, Scheele, and Schmid (2024a). This includes definitions, algorithms, figures, and tables, which have been cited accordingly. The text has been rephrased such that the referenced publication is embedded into this thesis. Appendix C.4 provides information to which extent the author has contributed to Slany, Scheele, and Schmid (2024a).

CAIPI as proposed originally by Teso and Kersting (2019) has been evaluated by unlearning the spurious correlation, which had been induced by decoy pixels that correspond to the class label. Counterexamples have a random association between color and label and incrementally weaken the spurious correlation. Obviously, this approach does not transfer well into practical applications of CAIPI beyond academic experiments. Therefore, the previous section has proposed a data augmentation procedure (Figure 4.13), which, in theory, can be applied to every binary image classification task, where users can be expected to identify the decisive features.

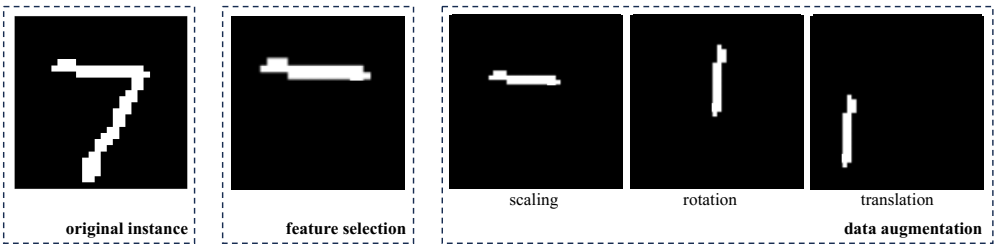

Figure 4.15: Problematic data augmentation (Slany, Scheele, and Schmid, 2024a). Depending on the domain, data augmentation, e.g., Figure 4.13, might produce unintended results. In this case, the augmented decisive features of a seven mimic a one.

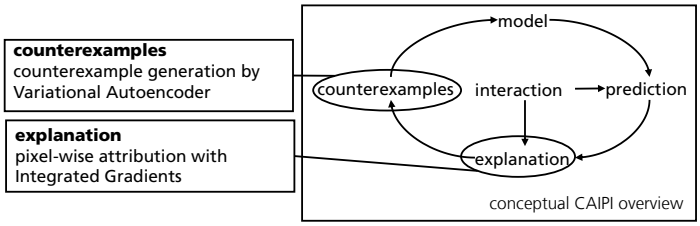

Figure 4.16: Bayesian CAIPI component adaptations (Slany, Scheele, and Schmid, 2024a).

What has not been discussed is that data augmentation pipelines are handcrafted in the sense that they are implemented by ML engineers, frequently without domain knowledge. Figure 4.15 executes the identical data augmentation pipeline, consisting of scaling, rotating, and translating the decisive features, for the classification task ones versus sevens on the MNIST data set[26].

Suppose that horizontal bars are decisive for class seven and vertical bars for class one. Figure 4.15 shows that data augmentation might produce implausible results. A counterexample generated from the augmented decisive features of a seven mimics a one, despite they are associated with the label seven. Such counterexamples do not project the human domain knowledge consistently into the model and, therefore, cannot be expected to have a beneficial impact on the correctness of the decision-making mechanism of the classifier.

Problem Counterexamples generated by data augmentation might be implausible, which impedes the model optimization with CAIPI.

Solution Statistical generative models promise realistic counterexamples. A probabilistic CAIPI variant, termed Bayesian CAIPI, is based on a Variational Autoencoder (Kingma and Welling, 2014) that encodes all available data into the latent space. Counterexamples are sampled from the Variational Autoencoder's likelihood distribution. The classifier predicts images from the latent space and is thus assumed to be differentiable. This enables Bayesian CAIPI to utilize Integrated Gradients (Sundararajan, Taly, and Yan, 2017), a pixel-wise explanation method for differentiable image classification models. The adaptations compared to traditional CAIPI (Teso and Kersting, 2019) are summarized in Figure 4.16.

Contribution Bayesian CAIPI draws counterexamples from the likelihood distribution of a Variational Autoencoder. Therefore, this section (i) contributes a novel

[26] `https://yann.lecun.com/exdb/mnist/`, 18 July 2024.

counterexample generation procedure for image data. Detached from CAIPI adaptations, this section (ii) integrates a classification task into the Variational Autoencoder and trains a classifier directly from the latent space.

Subordinated research questions The predictive performance of Bayesian CAIPI and its ability to find the correct decision-making mechanism will be evaluated and compared to traditional CAIPI (Teso and Kersting, 2019) and default deep learning by the following research questions (Slany, Scheele, and Schmid, 2024a):

R2.11 Do counterexamples improve the model's predictive quality?
R2.12 Do counterexamples improve the model's ability to follow the correct decision-making mechanism?
R2.13 Which is superior, Bayesian CAIPI, CAIPI, or default deep learning?

Auxiliary Methods

This section is built upon Notation 4.1. Despite the process to obtain the most-informative instance does not change compared to Definition 2.1, it still needs to be redefined. This is done in Definition 4.17, which now accounts for image data.

Definition 4.17 (Most-Informative Instance for Images (Slany, Scheele, and Schmid, 2024a))**.** Assume for now that f returns the prediction score. Then, let the *most-informative instance for* a set of *images* be defined as:

$$m = \arg\min_n \left\{ |f(X_{\mathcal{U}}^{(n)}) - \alpha| | X_{\mathcal{U}}^{(n)} \in \mathcal{U} \right\},$$

where $\alpha \in [0, 1]$ is the prediction threshold of f. Suppose that the procedure MII in this section takes f and $\mathcal{U}$ as input and returns m.

Compared to the CAIPI variant for image data from the previous section, this section integrates two existing approaches into CAIPI: Integrated Gradients (Sundararajan, Taly, and Yan, 2017) as pixel-wise attribution map given an image and a Variational Autoencoder (Kingma and Welling, 2014) to generate counterexamples. Both concepts will be formalized in the following two paragraphs.

Integrated Gradients Bayesian CAIPI uses a pixel-wise attribution map to assess the relevance that the classification model associates with each pixel. The attribution map is obtained by Integrated Gradients (Sundararajan, Taly, and Yan, 2017). Its basic idea is to calculate the partial derivatives of the classifier wrt. pixels, which are incrementally influenced by a baseline image. Theoretically, the interpolation factor ranges from zero to one, which is why Integrated Gradients forms an integral. Out of efficiency reasons, Integrated Gradients can be approximated by a subset of the interpolation factors (Definition 4.18).

Definition 4.18 (Integrated Gradients (Sundararajan, Taly, and Yan, 2017; Slany, Scheele, and Schmid, 2024a))**.** Let the Riemman summation approximation of *Integrated Gradients* (IG) be defined as follows:

$$IG_i \approx (X_i - X_{B_i}) \sum_{k=1}^{K} \frac{\partial f(X_B + \frac{k}{K}(X - X_B))}{\partial X_i} \frac{1}{K},$$

where X_B is a baseline image (here an image with pixel values zero) and K is the number of differentiation steps. Let $M \in \{0, 1\}^{W \times H}$ be the explanation mask such that $M_i = 1$ if $IG_i > \beta$ and $M_i = 0$ otherwise, where β is an attribution threshold for a prediction $\hat{y} = f(X)$. Suppose that EXP in this section takes f, X, and X_B as input and returns M.

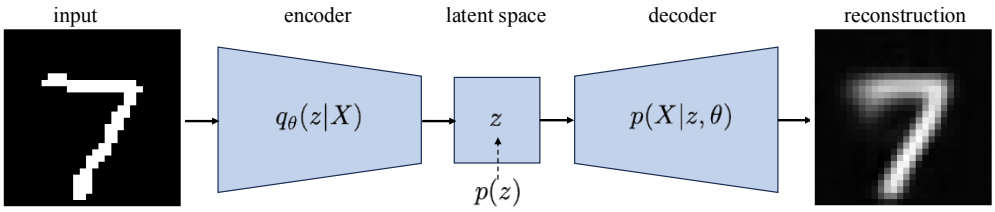

Figure 4.17: Variational Autoencoder (Slany, Scheele, and Schmid, 2024a). The encoder projects an image into the latent space of which the decoder reconstructs it.

The attribution values are mapped to a binary explanation mask. Values above a certain attribution threshold are said to be relevant; others are said to be irrelevant.

Variational Autoencoder Traditional Autoencoders encode data into a smaller representation, termed latent space, and reconstruct the data from the latent representation afterwards (Vincent et al., 2008). Their probabilistic counterparts, Variational Autoencoders (Kingma and Welling, 2014), define a probability distribution over each model component: the prior distribution over the latent space, the posterior distribution over the encoder, and the likelihood distribution over the decoder. Samples from the prior distribution evaluate the likelihood distribution for a given image, resulting in a novel representation of an input instance (Definition 4.19).

Figure 4.17 contains an example from the image domain: An image displaying a seven is compressed into a latent representation by the encoder network. The decoder network reconstructs the image by samples from the latent distribution. As the reconstructed seven is a novel image, Variational Autoencoders are statistical generative models (Kingma and Welling, 2014).

Definition 4.19 (Variational Autoencoder (Kingma and Welling, 2014; Doersch, 2016; Slany, Scheele, and Schmid, 2024a))**.** Let a *Variational Autoencoder vae* : $\mathcal{X} \rightarrow \hat{\mathcal{X}}$ encode an image into a latent space representation by an encoder $enc : \mathcal{X} \rightarrow \mathcal{Z}$ and reconstruct the image from the latent space representation by a decoder $dec : \mathcal{Z} \rightarrow \hat{\mathcal{X}}$. The set of latent space representations is defined as $\mathcal{Z} = \{z^{(1)}, ..., z^{(n)}\}$ and the set of reconstructed images is given as $\hat{\mathcal{X}} = \{\hat{X}^{(1)}, ..., \hat{X}^{(n)}\}$. Both correspond to the set of images $\mathcal{X}$. Suppose a procedure FitVae that takes $\mathcal{X}$ as input and returns *enc* and *dec*. Moreover, let there be the following probability distributions: the marginal distribution $p(X)$, the prior distribution $p(z)$, the likelihood distribution of the decoder $p(X|z,\theta)$, and the posterior distribution of the encoder $q_\theta(z|X)$. The parameter θ is now a placeholder for the parameters in *vae*.

The state-of-the-art optimization procedure for Variational Autoencoders is Variational Inference (Blei, Kucukelbir, and McAuliffe, 2017). The true posterior distribution is intractable caused by an unknown but constant marginal distribution over the data, as the data are only partially observed. Therefore, Variational Inference searches for a tractable distribution that approximates the posterior (Definition 4.20).

By the Bayes rule, the distributional difference, termed Kullback-Leibler (KL) divergence (e.g., Blei, Kucukelbir, and McAuliffe, 2017), can be reformulated into the sum of the marginal and an expectation conditioned on the distribution over the latent space. A second KL divergence is introduced between the proposal posterior and the prior, which is subtracted from the likelihood distribution in the Evidence Lower BOund (ELBO) (e.g., Blei, Kucukelbir, and McAuliffe, 2017). It becomes clear that maximizing the ELBO results in a well-suited approximation of the posterior, as it minimizes the approximation loss given a unknown normalizing constant.

Definition 4.20 (Variational Inference (Kingma and Welling, 2014; Doersch, 2016; Blei, Kucukelbir, and McAuliffe, 2017; Slany, Scheele, and Schmid, 2024a))**.** *Variational Inference* optimizes Variational Autoencoders. The Kullback-Leibler (KL) divergence is the difference between the encoder $q_\theta(z|X)$ and the intractable true posterior distribution $p(z|X)$:

$$\mathrm{KL}\left[q_\theta(z|X) \,||\, p(z|X)\right] = \log p(X) + \mathtt{E}_{q_\theta(z)}\left[\log q_\theta(z|X) - \log p(X|z) - \log p(z)\right].$$

This definition can be reformulated wrt. the marginal distribution $p(X)$:

$$\log p(X) = \mathtt{E}_{q_\theta(z)}\left[\log p(X|z)\right] - \mathrm{KL}\left[q_\theta(z|X) \,||\, p(z)\right] + \mathrm{KL}\left[q_\theta(z|X) \,||\, p(z|X)\right],$$

where a second KL divergence is drawn between the encoder and the prior $p(z)$. Suppose that the Evidence Lower BOund (ELBO) is defined as:

$$\mathrm{ELBO}(q) = \mathtt{E}_{q_\theta(z)}\left[\log p(X|z)\right] - \mathrm{KL}\left[q_\theta(z|X) \,||\, p(z)\right].$$

Inserting the ELBO into the original equation yields:

$$\mathrm{KL}\left[q_\theta(z|X) \,||\, p(z|X)\right] = -\mathrm{ELBO}(q) + \log p(X).$$

Maximizing the ELBO is equal to minimizing the approximation loss of the posterior given the marginal distribution.

Practically, the KL divergence part of the ELBO is still unsuitable for deep learning optimization. The latent representation stems from a distribution and is thus probabilistic, making it costly or even impossible to compute during backpropagation. The solution is the reparameterization trick (Kingma and Welling, 2014). The prior is mostly a standard normal distribution such that the resulting posterior of the encoder will also be normally distributed. The encoder network is constructed such that it outputs two parameters: μ for the posterior mean and σ for the posterior variance. A random draw from a normal distribution with the mean equal to the posterior mean and a fixed variance, mostly chosen to be one, $\epsilon \sim N(\mu, 1)$ serves as a scaling parameter for the posterior variance. The latent representation in the simplest case is: $z = \mu + \epsilon\sigma$. The benefit of reparameterization is that the latent representation is still random, but the randomness originates in a distribution outside the deep learning optimization process. The objective of the likelihood is to maximize the reconstruction quality. Therefore, it is practically measured by conventional deep learning loss functions such as the l2-norm (Doersch, 2016).

Introducing a preliminary latent space encoding requires some notational adjustments for this section (Notation 4.2). First, the association subscripts to labeled and unlabeled data sets need to be expanded on the latent space encodings. Second, nesting is a mathematically elegant way of combining a latent space encoding and a classifier if both functions are differentiable. This results in a single differentiable model. The encoder is fixed when training the classifier and vice versa.

Notation 4.2 (Image Classification from Latent Space (Slany, Scheele, and Schmid, 2024a))**.** Let f' be a differentiable binary classification model from the latent space $f' : \mathcal{Z} \to \mathcal{Y}$ with an inference denoted as $y = f'(z)$. Extending Notation 4.1, let the association of $z \in \mathcal{Z}$ to labeled and unlabeled sets be indicated by subscripts – precisely, $z_\mathcal{L} \in \mathcal{Z}_\mathcal{L}$ for the labeled and $z_\mathcal{U} \in \mathcal{Z}_\mathcal{U}$ for the unlabeled set. Suppose that the procedure FIT now trains or updates f' taking $\mathcal{Z} \times \mathcal{Y}$ as input. The generalized notation for the nested function $f'(enc(X))$ of each $X \in \mathcal{X}$ is $f' \circ enc$.

Adaptations

In comparison to CAIPI (Teso and Kersting, 2019), Bayesian CAIPI uses Integrated Gradients (Sundararajan, Taly, and Yan, 2017) as local explainer and trains a Variational Autoencoder (Kingma and Welling, 2014) on all available training data before the optimization cycle to sample counterexamples from the likelihood distribution. Integrated Gradients and the optimization of Variational Autoencoders has sufficiently been discussed in the previous section. Algorithm 4.4 illustrates Bayesian CAIPI's counterexample generation procedure. Samples from the likelihood distribution are drawn for the desired amount of counterexamples (line 3). Note that the reparameterization trick (Kingma and Welling, 2014) also applies to this step. The generated counterexample is multiplied by the matrix of decisive features to black out indecisive features (line 4). This step is optional and only applicable if the variance in the reparameterization step is sufficiently small such that the positions of the decisive features in the reconstructed image change only to a minor extent. In practice, the matrix of decisive features can also be obtained from the user interface proposed by the previous section (Figure 4.18). In the **RWR** case, where counterexamples are generated, the prediction has already been evaluated as being correct, which is why each counterexample can be connected to its validated label in a final step (line 5).

Bayesian CAIPI (Algorithm 4.5) trains a Variational Autoencoder before the optimization cycle on all available images to encode the labeled and unlabeled data sets (line 1). A classification model is trained with the encoded labeled images (line 4) to select the most-informative instance from the encoded unlabeled data set afterwards (line 5). The prediction of the most-informative instance (line 6) is evaluated (line 7). If it is wrong, the image of the most-informative instance is added to the labeled data set with the corrected label (line 9). In case of a correct prediction, Integrated Gradients (Sundararajan, Taly, and Yan, 2017) generates a mask of relevant pixels wrt. the nested classification model (line 11). A user annotates the decisive pixels (line 12). If the generated explanation is equal to the user annotation, the image of the most-informative instance and its prediction is added to the labeled data set without further actions (line 14). Otherwise, counterexamples – decisive features of novel instances – are generated (line 16). The counterexample data sets accompany the most-informative instance in the labeled data sets (line 17). The image of the most-informative instance is removed from the unlabeled data set (line 18).

Algorithm 4.4: GEN$(p(X|z,\theta), y, D, c)$ (BAYESIANCAIPI) (Slany, Scheele, and Schmid, 2024a)

Input: Decoder's likelihood distribution $p(X|z,\theta)$, label y, decisive features D, number of counterexamples c

Output: Data sets of labeled counterexamples $\mathcal{X}', \mathcal{Y}'$

1: $\mathcal{X}' \leftarrow \emptyset; \mathcal{Y}' \leftarrow \emptyset$
2: **for** $1 : c$ **do**
3: $\quad \hat{X} \leftarrow p(X|z,\theta)$ ▷ *Definition 4.19*
4: $\quad \hat{X} \leftarrow \hat{X} \cdot D$ ▷ *Definition 4.13*
5: $\quad \mathcal{X}' \leftarrow \mathcal{X}' \cup \{\hat{X}\}; \mathcal{Y}' \leftarrow \mathcal{Y}' \cup \{y\}$
6: **return** $\mathcal{X}', \mathcal{Y}'$

Algorithm 4.5: BAYESIANCAIPI($\mathcal{L}, \mathcal{U}, c, n, X_B$) (Slany, Scheele, and Schmid, 2024a)

Input: Data sets $\mathcal{L}$ and $\mathcal{U}$, number of counterexamples c, iteration budget n, baseline image X_B
Output: Models f', enc

$enc, dec \leftarrow \text{FITVAE}(\{X|(X,y) \in \mathcal{L}\} \cup \mathcal{U})$ ▷ *Definition 4.19*
for $1 : n$ **do**
  $\mathcal{Z}_\mathcal{L}, \mathcal{Y}_\mathcal{L} \leftarrow \{(enc(X), y)|(X,y) \in \mathcal{L}\}; \mathcal{Z}_\mathcal{U} \leftarrow \{enc(X)|X \in \mathcal{U}\}$
  $f' \leftarrow \text{FIT}(\mathcal{Z}_\mathcal{L} \times \mathcal{Y}_\mathcal{L})$ ▷ *Notation 4.2*
  $m \leftarrow \text{MII}(f' \circ enc, \mathcal{U})$ ▷ *Definition 4.17*
  $\hat{y}^{(m)} \leftarrow f'(z_\mathcal{U}^{(m)})$
  $y^{(m)} \leftarrow \text{INTERACT}(X_\mathcal{U}^{(m)})$ ▷ *Figure 4.18*
  if $\hat{y}^{(m)} \neq y^{(m)}$ **then**
    $\mathcal{L} \leftarrow \mathcal{L} \cup \{(X_\mathcal{U}^{(m)}, y^{(m)})\}$ ▷ *Case: **W***
  else
    $M \leftarrow \text{EXP}(f' \circ enc, X_\mathcal{U}^{(m)}, X_B)$ ▷ *Definition 4.18*
    $D \leftarrow \text{INTERACT}(M)$ ▷ *Figure 4.18*
    if $M = D$ **then**
      $\mathcal{L} \leftarrow \mathcal{L} \cup \{(X_\mathcal{U}^{(m)}, \hat{y}^{(m)})\}$ ▷ *Case: **RRR***
    else
      $\mathcal{X}', \mathcal{Y}' \leftarrow \text{GEN}(dec(X_\mathcal{U}^{(m)}), \hat{y}^{(m)}, D, c)$ ▷ *Algorithm 4.4*
      $\mathcal{L} \leftarrow \mathcal{L} \cup \{(X_\mathcal{U}^{(m)}, \hat{y}^{(m)})\} \cup \mathcal{X}' \times \mathcal{Y}'$ ▷ *Case: **RWR***
  $\mathcal{U} \leftarrow \mathcal{U} \setminus X_\mathcal{U}^{(m)}$
return f', enc

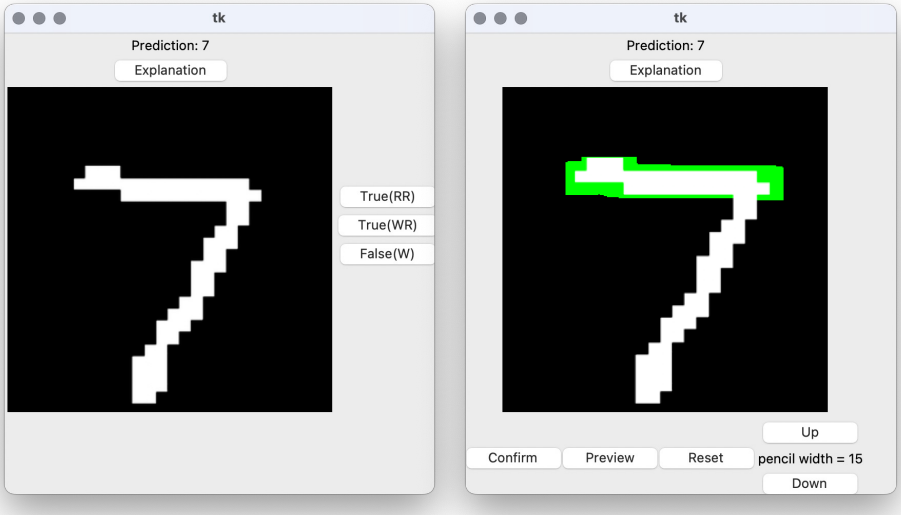

Figure 4.18: User interface for Bayesian CAIPI (Slany, Scheele, and Schmid, 2024a). The CAIPI user interface (Figure 4.14) can also be used for the image classification task ones versus sevens. Here, the seven has been predicted correctly but out of the erroneous reasons as the explanation is empty (left) and the user annotates the decisive features (right).

There are some modifications present in Bayesian CAIPI in comparison to the first CAIPI formalization of this thesis (Algorithm 2.1). Bayesian CAIPI operates on image data; Algorithm 2.1, in contrast, on tabular data. Even when neglecting this adaptation, Bayesian CAIPI contains modifications in three regards: (i) Bayesian CAIPI prepares the counterexample generation procedure in the sense that it approximates a latent representation of all images. The counterexamples are sampled from the likelihood distribution. (ii) The classification model is nested in the sense that either encoded images are used as input or the classifier is attached to the encoder. (iii) This specification is also important to the explanation component of Bayesian CAIPI, which requires differentiable models (Sundararajan, Taly, and Yan, 2017).

By nesting two differentiable models f' and *enc* and using an explainer for differentiable models (Sundararajan, Taly, and Yan, 2017), Bayesian CAIPI is no longer model-agnostic in the strict sense. The first modification is motivated by mathematical convenience, the second because LIME generates varying explanations when substituting the underlying segmentation algorithm (Schallner et al., 2019). By training any image classification model f directly on the latent representation and treating the encoding as a separate step, the model invariance property can be restored. What is left is a replacement for the explainer by a model-agnostic method such as LIME (Ribeiro, Singh, and Guestrin, 2016). Hence, both modifications are not necessary for Bayesian CAIPI and can easily be reversed, which is why Bayesian CAIPI is still classified as being a model-agnostic XIML method.

Experiments

Setup In five experimental iterations, the classification task ones versus sevens on the MNIST data set[27] is evaluated wrt. the predictive performance computed by the accuracy metric and the ability to follow the correct decision-making mechanism. An explanation is said to be correct if at least one high-activated pixel is on the vertical bar for class one and at least one high-activated pixel is on the horizontal bar and none on the vertical bar for class seven. A bar is defined as a consecutive alignment of at least five pixels along one axis. The attribution threshold (Definition 4.18) is set to 0.025. From each of the two classes, $6,000$ randomly selected instances are in the unlabeled data set. The remaining $1,000$ instances of each class are combined to a test data set. The initial labeled data set consists of ten images, of which each pixel is sampled from a uniform distribution with lower bound zero and upper bound one. The initial model is supposed to be completely uninformative. The goal is to project the impact of $\{0, 1, 3, 5\}$ counterexamples per **RWR** iteration during 100 optimization iterations directly to Bayesian CAIPI.

The Variational Autoencoder[28] has an encoder and a decoder network. The encoder consists of two convolutional layers with kernel size three and stride parameter two for each dimension. There are 32 filters in the first and 64 in the second convolutional layer. The convolutional output is flattened for the final dense layer with four neurons. The dense layer contains four neurons as the latent dimensionality is set to two because of two classes and reparameterization implies network outputs for the first and the second moment. This means two parameters times two latent dimensions. The decoder starts with a dense layer with $1,568$ neurons. The latent representation is reshaped into two dimensions and propagated through three transposed convolutional layers to reconstruct the image. The first two transposed convolutional layers invert the encoder, the third has a single filter. Except for the

[27] `https://yann.lecun.com/exdb/mnist/`, 18 July 2024.

[28] `https://www.tensorflow.org/tutorials/generative/cvae`, 19 July 2024.

Table 4.9: Bayesian CAIPI experimental results (Slany, Scheele, and Schmid, 2024a). Average Bayesian CAIPI results for various **Metrics** and c counterexamples. The best results per metric are highlighted in bold. Standard deviations are provided in brackets.

Metric	c: 0	1	3	5
accuracy	**98.60** (0.55)	98.40 (0.55)	98.40 (0.55)	98.40 (0.55)
ratio corr. expl.:				
class one	**77.70** (10.47)	77.62 (19.83)	44.94 (34.96)	58.63 (17.73)
class seven	0.30 (0.17)	0.64 (1.26)	1.19 (1.44)	**8.38** (1.70)

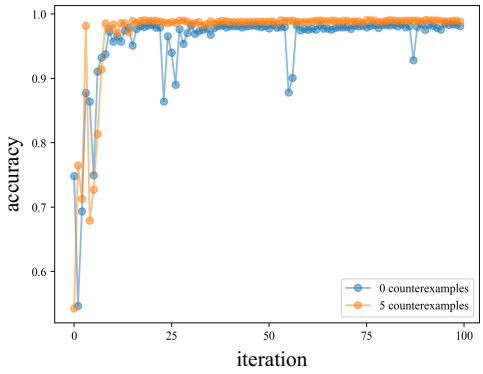

Figure 4.19: Bayesian CAIPI predictive performance (Slany, Scheele, and Schmid, 2024a). Accuracy for a single optimization run of zero (blue) and five (orange) counterexamples.

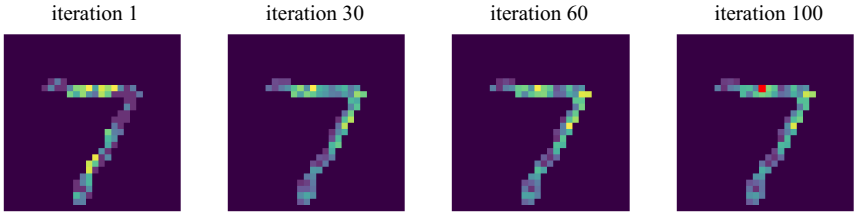

Figure 4.20: Bayesian CAIPI explanatory quality (Slany, Scheele, and Schmid, 2024a). Pixel relevance visualization for an exemplary prediction across Bayesian CAIPI optimization iterations. Only the red pixel lies above the attribution threshold.

final layers of the encoder and the decoder, all layers use rectified linear unit activation. The Variational Autoencoder is trained for 20 epochs with the Adam optimizer and a learning rate of 0.0001.

The classification model is stacked on top of the encoder. It has two dense layers: 512 neurons and rectified linear unit activation in the first and one neuron with sigmoid activation in the second layer. It is trained for 50 epochs with the Adam optimizer, a learning rate of 0.001, and the binary cross-entropy loss function.

The Variational Autoencoder is trained once prior to the optimization cycle. The classification model is re-trained in each Bayesian CAIPI iteration. All models are implemented in TensorFlow[29].

Two baseline tests are conducted: First, Bayesian CAIPI's counterexample generation procedure (Algorithm 4.4) is replaced by the data augmentation pipeline (Figure 4.15) without the rotation step to mimic traditional CAIPI (Teso and Kersting,

[29] `https://www.tensorflow.org`, 19 July 2024.

2019). This experiment is executed with five counterexamples per **RWR** iteration. The second baseline test is the default deep learning optimization, where all available training data are used to train the classifier. Traditional CAIPI optimization yields a test accuracy of 97.97 percent. The model correctly explains 68.67 percent of ones and 0.00 percent of sevens. The default deep learning model has a test accuracy of 98.06 percent and explains 36 percent of ones and 0.69 percent of sevens correctly.

Results The average ELBO of the Variational Autoencoder is -28.22 (0.64). Table 4.9 indicates a stable predictive performance, where the zero counterexamples case, in fact, has slight benefits. The explanatory performance results are interesting. Whereas the correctness of the decision-making mechanism for class one tends to diminish with an increasing amount of counterexamples, the decision-making mechanism for class seven has not been present for the zero counterexamples case but increases up to 8.38 percent in the five counterexamples runs. The results suggest that the model has only learned the decision-making mechanism for a one and has inferred a seven when no one has been detected. Counterexamples pertubate this single-sided decision making – they weaken the decision-making mechanism of class one and strengthen the mechanism for class seven.

A graphical comparison of selected zero and five counterexamples cases visualizes that counterexamples contribute to a faster convergence with fewer numerical instabilities (Figure 4.19). A possible explanation is that counterexamples increase the training data set more quickly. Visualizing the explanation masks for an exemplary seven hint at supporting the numerical results (Figure 4.20). In iteration one, both bars are equally attributed. The attribution tends to move to the horizontal bar, resulting in a relevant pixel after the final optimization iteration.

In general, if balanced decision making is a quality criterion, Bayesian CAIPI with five counterexamples outperforms traditional CAIPI and the default deep learning optimization. Moreover, similar to traditional CAIPI (Section 4.2.1), Bayesian CAIPI also reduces the labeling effort compared to default deep learning optimization.

Section Summary

Summary Historically, CAIPI (Teso and Kersting, 2019) has been evaluated by counterexamples that randomize decoy pixels, which had been induced into image data sets such that each color corresponds to a specific class. A preceding CAIPI variant, applying CAIPI to human-centric classification scenarios (Slany et al., 2022), has used data augmentation to generate counterexamples. Data augmentation pipelines still require some degree of domain knowledge. The absence of the necessary domain knowledge might produce implausible counterexamples (Figure 4.15). Therefore, this section has proposed an alternative approach to generate counterexamples – embedded into Bayesian CAIPI, a probabilistic CAIPI variant. Variational Autoencoders (Kingma and Welling, 2014) encode all available data – labeled and unlabeled – into a latent distribution. Counterexamples are sampled from the decoder and multiplied by the binary mask of decisive features to black out indecisive features. This way, Bayesian CAIPI generates novel yet realistic representations of decisive features. The experiments show that Bayesian CAIPI matches the predictive performance of default deep learning and traditional CAIPI (Teso and Kersting, 2019) but outperforms both regarding the model's ability to follow the correct decision-making mechanism. Despite the predictive performance of Bayesian CAIPI does not benefit from increasing the number of counterexamples, the decision-making mechanism of the classification model transforms from a single-sided representation of the target classes to reflecting both classes (Table 4.9).

Answers to subordinated research questions The experimental evaluation leads to the following answers to the subordinated research questions formulated in this section (Slany, Scheele, and Schmid, 2024a):

R2.11 Do counterexamples improve the model's predictive quality?
No, counterexamples in Bayesian CAIPI *do not improve the predictive quality in terms of the accuracy metric.*

R2.12 Do counterexamples improve the model's ability to follow the correct decision-making mechanism?
If balanced decision making is a desirable criterion, counterexamples in Bayesian CAIPI *improve the ability to follow the correct decision-making mechanism.*

R2.13 Which is superior, Bayesian CAIPI, CAIPI, or default deep learning?
Revisit the preliminary answer. Bayesian CAIPI*'s results are similar to the ones of traditional* CAIPI *and default deep learning, except it is the only evaluated approach that induces the decision-making mechanism of the class seven.*

Related results Bayesian CAIPI proposes an alternative counterexample generation variant compared to CAIPI (Teso and Kersting, 2019) similar to Slany et al. (2022) and Heidrich et al. (2023). The experiments of this section are in line with the original CAIPI evaluation (Teso and Kersting, 2019) and another CAIPI modification that aims to optimize classification models for image data using the CAIPI framework and human feedback (Slany et al., 2022). Both repeat the identical experimental setup with a varying amount of counterexamples per **RWR** iteration ranging from zero to five. In fact, Bayesian CAIPI is supposed to extend the experiments of Slany et al. (2022), which serves as a baseline. In a direct comparison, both Bayesian CAIPI and Slany et al. (2022) match the predictive performance of the default deep learning model but only Bayesian CAIPI improves the model's decision-making mechanism.

Limitations The experiments reveal that Bayesian CAIPI optimization induces the decision-making mechanism of a second class into a binary classification model, which otherwise only decides between presence and absence of the first class. Nevertheless, the magnitude of the adjustment of the decision-making mechanism is relatively small and Bayesian CAIPI provides no measurable benefits for the predictive performance. Possible reasons for this observation can be broken down into the categories limitations of the experimental setup and non-ideal design choices.

- **Experimental setup**:
  The classification task ones versus sevens appears to be too simple – even simpler than in the previous section – for state-of-the-art image classification models. An indication is that the predictive performance does not improve comparing coactive learning (Shivaswamy and Joachims, 2015) and the five counterexamples per **RWR** iteration case. Also, the experiments reveal comparatively high standard deviations, especially for the decision-making mechanism in the three counterexamples case. The variance might originate in the latent dimension of the Variational Autoencoder. Indeed, the hyperparameters of which have not yet been systematically evaluated. The amount of latent dimensions, for instance, has been chosen because of the binary classification model. The risk exists that this representation does not capture the entity of all ones and sevens well, which potentially results in noisy counterexamples for some instances that have been predicted correctly for erroneous reasons.
- **Design choices**:
  The missing systematic evaluation of the Variational Autoencoder transitions

into specific component decisions. Integrated Gradients (Sundararajan, Taly, and Yan, 2017) appears to be a non-ideal choice for Bayesian CAIPI's local explanation component for several reasons: First, as already been discussed, it restricts Bayesian CAIPI to differentiable classification models – unnecessarily making a model-agnostic XIML method model-specific. This section has presented ways to reverse this restriction in context of Algorithm 4.5. Second, Integrated Gradients is a pixel-wise attribution method. Regarding correct decision making, this means that single pixels decide between correct and wrong mechanisms. This might over-amplify the importance of single pixels. Also, explanations for instances of class one are always correct if the attribution is sufficiently high, as no horizontal bars exist. This might be why ones are over-proportionally often predicted out of correct reasons (Table 4.9). In general, the definition of decision-making mechanisms being correct if horizontal or vertical bars are sufficiently important, has its limits if bars are slightly rotated and not aligned along the x- or y-axis. Bars that cannot be identified will always cause incorrect decision making according to the experimental setup of this section. A final point to discuss is the mask of decisive features (Definition 4.13). In the context of Variational Autoencoders, it might be the case that the reconstruction deviates from the original image. An unexpectedly high deviation can lead to counterexamples with artifacts if some pixels are not blacked out or a subset of pixels is blacked out unintentionally.

4.3 Potential CAIPI Extensions

The model invariance property claimed by CAIPI (Teso and Kersting, 2019) would also include models used for regression, optimization, and clustering. Although XIML algorithms have been used for various data types, none of the reviewed publications specifically addresses the mentioned ML tasks (Table 1.1). Unfortunately, this is also not a contribution made by this thesis. However, it notices this gap and proactively proposes suitable explanation algorithms for regression and optimization (Section 4.3.1) as well as for clustering (Section 4.3.2). Each of the two sections will conclude with potential feedback mechanisms such that the proposed method can be integrated into variants of CAIPI. Contrary to the evaluations of previously proposed CAIPI variations, the subsequent sections will not contain research questions, as they would not pay into the overarching research questions of this thesis (Section 1.2).

4.3.1 Regression and Optimization

Regression models have various applications. They can either be used for pure analysis purposes, where a statistician is interested in the quantitative relationship between a feature and a regression target, they can be used to explore relationships between variables, or for forecasting (Welc and Esquerdo, 2018). Concepts of regressions are also recycled in optimization (Wirth, Schmid, and Voget, 2022). Depending on the use case, the number of features, and the complexity of the model, they may cause incomprehensibility for domain experts. An iterative application of regression models exacerbates this circumstance, as high-quality feedback from domain experts influences optimization positively (Wirth, Schmid, and Voget, 2022).

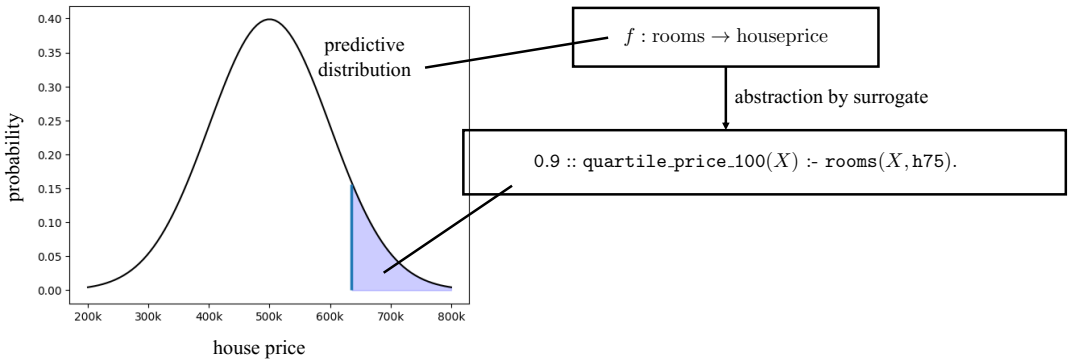

Figure 4.21: PHAL overview. PHAL approximates an opaque regression model by probabilistic logic rules corresponding to a statistical property of the regression target.

Problem Although regression models are vastly built upon domain-specific knowledge, their ex-post adaptation by the induction of novel knowledge is a major challenge due to their complexity and the number of features involved. By now, there exists no XIML framework that focuses on regression models.

Solution This section overcomes a gap that prevents the induction of knowledge into regression models by symbolizing regressions with probabilistic logic (Raedt, Kimmig, and Toivonen, 2007). Regression models are abstracted into multiple binary classifications wrt. statistical characteristics of the regression target. The central argument is: Statistical abstractions, such as means, quartiles, or standard deviations, describe regression models more intuitively than simple scalars do.

Contribution This section contributes Post-Hoc model Approximation with Logic (PHAL). PHAL relies on a statistical feature extraction procedure, where the extracted features are exploited as probabilistic facts and the rule learning algorithm PROBFOIL$^+$ (Raedt et al., 2015) yields probabilistic logic rules, which serve as a surrogate for the regression model. The algorithmic contribution behind PHAL is the connection of statistical feature extraction and probabilistic logic, which reduces the search space for probabilistic rule learning and expands PROBFOIL$^+$ to regression scenarios.

Among other experiments, PHAL is applied to the Boston housing data set[30]. Figure 4.21 visualizes the intuition behind PHAL: Suppose a scenario, where the average house prices of districts in Boston are conditioned on the room number, which can be expressed in a regression model that is assumed to be opaque to domain experts such as the local mayors. The regression model is then abstracted in a logical surrogate model obtained by PHAL.

This section does not contain dedicated research questions, as those would be unrelated to this thesis's major research questions. It, nevertheless, experimentally evaluates PHAL, which will be derived in detail in the next section, in comparison to GridEx (Sabbatini, Ciatto, and Omicini, 2021) – a state-of-the-art knowledge extraction method for regression models, which is based on hypercubes. The section summary will address the impact of the experimental results on CAIPI.

[30] `https://www.kaggle.com/datasets/schirmerchad/bostonhoustingmlnd`, 24 July 2024.

Methods

During this section, a Gaussian Process regression (Rasmussen and Williams, 2006) models the relationship between a tabular feature set with arbitrary values and a continuous target. PHAL will use a subset of feature instances – an explanation set $\mathcal{X}_{Exp}$ –, which is excluded from training to induce the surrogate model (Notation 4.3).

Notation 4.3 (Tabular Data Regression)**.** Again, let $(x_1, ..., x_i, ..., x_n)^T = x \in \mathcal{X}$ be a feature vector of a feature set with identifier set $\mathcal{F} = \{1, ..., i, ..., n\}$. Now, let $y \in \mathcal{Y} \subseteq \mathbb{R}$ be the target of a regression model $f : \mathcal{X} \to \mathcal{Y}$. Specifically, f is a Gaussian Process regression (Rasmussen and Williams, 2006) with radial basis kernel function. Let

$$f(x) = f_* | \mathcal{X}, \mathcal{Y}, x \sim N(\mathbb{E}\left[f_* | \mathcal{X}, \mathcal{Y}, x\right], \text{cov}(f_*)),$$

where f_* is the predictive distribution. Suppose that $y = (y_{mean}, y_{cov}) = f(x)$. Let $x^{(n)}$ ($y^{(n)}$) reference the n-th instance in $\mathcal{X}$ ($\mathcal{Y}$). Let $\mathcal{X}_{Exp} \subseteq \mathcal{X}$ be an explanation set.

Remark 4.6. The covariance of the predictive distribution $\text{cov}(f_*)$ is computed with the kernel function of the Gaussian Process regression model. See Rasmussen and Williams (2006) for details.

Example 4.8. A regression model estimates the average house price of districts in Boston. Assume that the average house price of `Charlestown` is calculated as:

$$y^{(\texttt{Charlestown})} = (y_{mean} = 750,000, y_{cov} = 375,000) = f(x^{(\texttt{Charlestown})}),$$

where the indication of the instance `Charlestown` is simplified by a superscript.

With Gaussian Processes (Rasmussen and Williams, 2006), each prediction emerges from a distribution, which explicitly quantifies the variation by the second moment of a normal distribution. The inverse coefficient of variation (Definition 4.21) projects the coefficient of variation to an interval ranging from zero to one and subtracts the value from one. It is assumed to be a suitable measure for the certainty of a prediction because it puts the estimate in relation to its variation.

Definition 4.21 (Inverse Coefficient of Variation)**.** Let the *inverse coefficient of variation* (*inv_cv*) be a measure for the certainty associated with a prediction:

$$y_{prob} = inv\_cv(y) = 1 - cv \text{ with } cv = y_{cov} \cdot (y_{mean} + 0.01)^{-1} \text{ and } cv \in [0, 1].$$

Example 4.9. Consider the values obtained by Example 4.8. The inverse coefficient of variation is calculated as follows:

$$y_{prob} = 0.5 \approx 1 - \left[375,000 \cdot (750,000 + 0.01)^{-1}\right].$$

PHAL has two steps: (i) An abstraction step, where the level of measurement is reduced by statistical feature extraction procedures (Definition 4.22) and (ii) a rule learning step, where PROBFOIL$^+$ (Raedt et al., 2015) returns a probabilistic logical surrogate for the regression model wrt. constructed examples. Note that statistical feature extraction can go beyond locality measurements such as the quartile values (Example 4.10). For instance, higher-order derivatives can be used to relate the slope of a function to the regression target. Hence, on the one hand, the abstraction step reduces the information degree, but simultaneously, on the other hand, increases the flexibility of the surrogate representation.

Definition 4.22 (Statistical Feature Extraction)**.** Suppose a transformation function $\overrightarrow{h}$ that serves as a *statistical feature extraction* method to reduce the information degree. Then, $\tilde{x} = \overrightarrow{h}(x)$ and $\tilde{y} = \overrightarrow{h}(y_{mean})$ indicate abstractions of feature and target.

Example 4.10. Assume that the room number is the only feature in $\mathcal{X}$ and $x_{\texttt{rooms}}^{(\texttt{Charlestown})} = 4$. Further, assume that $\overrightarrow{h}$ translates the numeric feature into quartile values. Then, suppose that $\overrightarrow{h}(x^{(\texttt{Charlestown})}) = \tilde{x}^{(\texttt{Charlestown})}$ and $\tilde{x}_{\texttt{rooms}}^{(\texttt{Charlestown})} = 0.75$, meaning that 75 percent of all districts in Boston on average have four or less rooms. Moreover, suppose that $\overrightarrow{h}(y_{mean}^{(\texttt{Charlestown})}) = \tilde{y}^{(\texttt{Charlestown})} = 1.0$

Both components, the inverse coefficient of variation and the statistical feature extraction, are part of the construction of probabilistic examples for regression models. Similar to the classification case, Definition 4.23 models the relation between features and the regression target: The instance is instantiated by a unary predicate representing a position in the data set. Each feature is modeled as a binary predicate containing the instance name and the abstracted feature. The target predicate is built wrt. a specific target value of the feature extraction procedure. If the instance evaluates to the target value, its probability is replaced by the inverse coefficient of variation. Otherwise, the target probability is zero.

Definition 4.23 (Probabilistic Example for Regression Models (Raedt et al., 2015))**.** Let ϕ, the target predicate of a *probabilistic example for regression models*, be built by:

$$p::\phi(t(x^{(n)})) \text{ with } \phi = <\overrightarrow{h}>\_<name(y)>\_<\texttt{target\_value}>,$$

where $t(x^{(n)}) = \texttt{i\_}n$ translates the position of $x \in \mathcal{X}$ into ProbLog syntax, $name(y)$ returns the name of the target, $<\overrightarrow{h}>$ is the name of the feature extraction method, the <target_value> is the desired abstracted value wrt. $\overrightarrow{h}$, and:

$p = inv\_cv(y^{(n)})$ (Definition 4.21)
  if $\tilde{y}^{(n)} =$ <target_value> with $\tilde{y}^{(n)} = \overrightarrow{h}(y_{mean}^{(n)})$ (Definition 4.22) and
$p = 0$ otherwise.

A single probabilistic example for regression models e is then defined as follows, where the second component is repeated for each $i \in \mathcal{F}$:

$$\texttt{instance}\big(t(x^{(n)})\big). \qquad name(x_i)\big(t(x^{(n)}), \texttt{h}\tilde{x}_i^{(n)}\big). \qquad p::\phi\big(t(x^{(n)})\big),$$

where $\tilde{x}^{(n)} = \overrightarrow{h}(x^{(n)})$. Suppose now that ToProbEx$(x^{(n)}, y^{(n)}, \overrightarrow{h}, \phi)$ returns e.

Example 4.11. Assuming that `Charlestown` is the only instance in $\mathcal{X}$, a probabilistic example can be constructed as follows, taking the information of the Examples 4.8 to 4.10 into account:

`instance(i_1).` `rooms(i_1,h75).` `0.5:: quartile_price_100(i_1).`

Remark 4.7. Note that unique or multiple feature extraction procedures can abstract each feature. In those cases, using distinct prescripts of the abstracted feature value is important – in particular, replace `h`.

Remark 4.8. Practically, multiple examples per target feature extraction procedure ϕ are constructed and aggregated to a target predicate set Φ – for instance, replacing <target_value> by 0, 25, 50, 75, and 100 in the quartile example.

Definition 4.24 (Metric Inferences)**.** Suppose that the success probability $Pr_S(q|R)$ for a query q (Definition 4.2) is calculated for each query predicate $\phi \in \Phi$ with corresponding rule set $R \in \mathcal{R}$. If $Pr_S(q|R) > \alpha$, the *metric inference* is $\hat{y} = \overleftarrow{h}_y(\tilde{y})$ and `nil` otherwise, where $\overleftarrow{h}_y$ reverses the transformation, `nil` stands for a missing value, and $\alpha \in [0, 1]$ is a prediction threshold.

Example 4.12. Suppose $\alpha = 0.5$ and assume the following rule sets:

$$\mathcal{R} = \Big\{ R_1 = \{0.9 :: \texttt{quartile\_price\_100}(X) \text{ :- } \texttt{rooms}(X, \texttt{h75})\},$$
$$R_2 = \{0.9 :: \texttt{quartile\_price\_25}(X) \text{ :- } \texttt{rooms}(X, \texttt{h25})\} \Big\}.$$

The success probability $Pr_S(\texttt{quartile\_price\_100}(\texttt{Charlestown}) | R_1) = 0.9$, the one for R_2 evaluates to zero. Reversing the transformation in the running example means replacing the quartile value with the quartile boundary, e.g., assume for Charlestwon, $\overleftarrow{h}_y(1.0) = 800,000$ (Example 4.10, Figure 4.21).

PHAL (Algorithm 4.6) induces a set of rules per target predicate (line 2). For each target predicate, it iterates over the feature data set (line 4), where each instance is predicted (line 5). Using the predictions, probabilistic examples are generated (line 6), which are together with the target predicate input for the rule induction (line 7).

Compared to PROBFOIL$^+$ (Raedt et al., 2015), Algorithm 4.6 is executed iteratively for abstracted features and targets. This also holds for the inference step (Definition 4.24), during which each target predicate is evaluated. If the success probability exceeds a prior specified prediction threshold, the transformation step is reversed such that continuous values replace the logical inferences. A drawback of this procedure is that logical inferences might produce missing values and the reverse transformation is deterministic.

Algorithm 4.6: PHAL(f, $\overrightarrow{h}$, Φ, $\mathcal{X}_{\text{Exp}}$)

Input: Regression model f, statistical feature extraction procedure $\overrightarrow{h}$, set of target predicates Φ, explanation data set $\mathcal{X}_{Exp}$
Output: Set of probabilistic logic rule sets $\mathcal{R}$

1: $\mathcal{R} \leftarrow \emptyset$
2: **for** $\phi \in \Phi$ **do** ▷ *Definition 4.23, Remark 4.8*
3: $\quad E \leftarrow \emptyset$
4: $\quad$ **for** $x \in \mathcal{X}_{Exp}$ **do**
5: $\quad\quad y \leftarrow f(x)$ ▷ *Notation 4.3*
6: $\quad\quad E \cup \text{TOPROBEX}(x, y, \overrightarrow{h}, \phi)$ ▷ *Definitions 4.21, 4.22 and 4.23*
7: $\quad \mathcal{R} \cup \text{INDUCE}(\phi, E)$ ▷ *Raedt et al. (2015)*
8: **return** $\mathcal{R}$

Experiments

Setup PHAL is evaluated in comparison to GridEx (Sabbatini, Ciatto, and Omicini, 2021) regarding its fidelity (Definition 4.25), stability (Definition 4.26), and complexity, measured as the average number of rules and the average number of predicates per rule on the Boston Housing (Housing) and Wine Quality (Wine) data sets[31]. GridEx[32] is a knowledge extraction algorithm for tabular data with continuous target variables. It searches for rules consisting of lower and upper bounds of features that describe a continuous outcome. The lower and upper bounds span an area in the feature space. Multiple areas can be visualized as cubical shapes, which is why the rule components of GridEx are termed hypercubes (Sabbatini, Ciatto, and Omicini, 2021).

Definition 4.25 (Fidelity (Rosenfeld, 2021))**.** Let *fidelity* be the performance difference wrt. classification metrics denoted as Δ, in this case the false positive (fp-) and false negative rate (fn-rate). The predictions of the regression model are abstracted by $\tilde{y} = \overrightarrow{h}(y_{mean})$ (Definition 4.22), where $(y_{mean}, y_{cov}) = f(x)$ (Notation 4.3).

Definition 4.26 (Stability (Rosenfeld, 2021))**.** Let $\mathcal{P}^{(t)}$ be the set of unique predicates of all rules in $\mathcal{R}$ in experimental iteration t. The *stability* (S) of predicates between two experimental iterations t and t' is:

$$S(\mathcal{P}^{(t)}, \mathcal{P}^{(t')}) = \frac{|\mathcal{P}^{(t)} \cap \mathcal{P}^{(t')}|}{|\mathcal{P}^{(t)} \cup \mathcal{P}^{(t')}|}.$$

Remark 4.9. To compute the stability of GridEx (Sabbatini, Ciatto, and Omicini, 2021), the set of unique hypercubes is used.

Fidelity is measured by classification instead of regression metrics. One can easily reduce the information degree of the regression model's predictions, but increasing the information degree of rule outcomes such as the reverse transformation step in PHAL most certainly introduces systematic differences between PHAL and GridEx. Therefore, all outcomes are abstracted into quartile values. The stability metric is based on the Jaccard index and divides the intersection of unique predicates by their union. The stability results are the average stability estimates comparing each of the five experimental iterations to every other experimental iteration.

From each data set, 50 percent of instances are used to train the Gaussian Process regression model[33] and 25 percent of instances are used to obtain the surrogate models. The remaining 25 percent of instances are test data, where the abstracted first moment of the predictions of the Gaussian Process regression are treated as ground truth. This step is crucial because the surrogate models are designed to mimic the regression model rather than to maximize the predictive performance.

Results PHAL is superior in terms of fidelity and stability (Table 4.10). GridEx, on the contrary, has slight benefits in terms of the complexity.

[31] Housing: `https://www.kaggle.com/datasets/schirmerchad/bostonhoustingmlnd`, Wine: `https://www.kaggle.com/datasets/danielpanizzo/wine-quality`, 25 July 2024.

[32] `https://github.com/psykei/psyke-python`, 25 July 2024.

[33] `https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.GaussianProcessRegressor.html`, 25 July 2025.

Table 4.10: PHAL experimental results. Comparison of fidelity determined by the performance difference (Δ) in false positive (fp) and false negative (fn) rate, complexity measured as the average number of rules and the average number of features per rule, and stability (S) between two **Method**s and **Data** sets wrt. a regression model.

Data	Method	Δ-fp-rate	Δ-fn-rate	# Rules	∅ Features	S
Housing	PHAL	**0.04** (0.03)	**0.11** (0.09)	7.20 (0.84)	2.87 (0.51)	**0.85** (0.04)
	GridEx	0.09 (0.05)	0.39 (0.40)	**5.00** (0.00)	**1.40** (0.55)	0.19 (0.01)
Wine	PHAL	**0.04** (0.02)	**0.07** (0.04)	**2.80** (0.84)	3.68 (0.29)	**0.87** (0.06)
	GridEx	0.09 (0.05)	0.47 (0.20)	4.40 (1.14)	**1.40** (0.89)	0.26 (0.03)

Section Summary

Summary PHAL (Algorithm 4.6) is a knowledge extraction procedure for regression models that returns a probabilistic logical theory specifically for a target predicate, which is a statistical property of the regression target. Users construct the shape of the surrogate model by defining the feature extraction procedures. The extracted features are converted into logical predicates, where PROBFOIL$^+$ (Raedt et al., 2015) induces the surrogate model. PHAL is beneficial regarding fidelity and stability in relation to GridEx (Sabbatini, Ciatto, and Omicini, 2021), a state-of-the-art knowledge extraction procedure for continuous target variables (Table 4.10).

Limitations PHAL connects statistical feature extraction and probabilistic rule learning, which is otherwise more common for classification tasks. Hence, there exist some conceptual limitations, which will be discussed. Moreover, the experimental evaluation is shallower compared to other evaluations in this thesis. Still, there exist some improvement opportunities.

- **Conceptual limitations**:
  Probabilistic logic, in general, has its benefits in an uncertain environment that consists of many complex relations in the feature space (Raedt, Kimmig, and Toivonen, 2007). An open question is whether tabular data regression tasks are scenarios, where the benefits of probabilistic logic apply. It might be the case that more intuitive and computationally efficient rule learning algorithms such as decision trees achieve comparative results to PROBFOIL$^+$ (Raedt et al., 2015). On the upside, PHAL increases knowledge extraction opportunities. For instance, when leveraging the second moment, PHAL is capable of returning theories that inform users where highly unstable results are probable. The inverse coefficient of variation (Definition 4.21) requires a model that estimates the uncertainty, which is a restriction to a specific class of models. Perspectively, when integrating PHAL into the CAIPI framework (Teso and Kersting, 2019), CAIPI's model invariance property would be jeopardized. Ensemble or Bayesian models at least extend the set of suitable models for PHAL.
- **Experimental setup**:
  The experiments should be extended towards more data sets and baseline methods. Regression and optimization have been treated as synonyms. Strictly, this chapter only provides evidence for regression problems. Critical in the broader scope of XIML is the data type, which determines the choice of the explanation and feedback algorithm. Therefore, it can be argued that the implications apply also in the optimization context, which is also underpinned by Chakraborty, Wirth, and Seifert (2024). The experimental setup systematically favors PHAL over GridEx because the regression target that serves as the ground truth for the fidelity is abstracted similarly to when

obtaining the surrogate model. GridEx, on the contrary, uses the raw feature and target space as input. For the sake of comparability, its predictions are abstracted afterwards. The randomness in the selection of the explanation data set might be the dominant reason for the instability of GridEx, which is more vulnerable than PHAL that transforms the feature and the target space a-priori and thus also removes possible outliers. The stability of GridEx is corrupted once the range of the feature target space changes, the one of PHAL only if quartile boundaries change drastically. PHAL explicitly induces a theory for each possible target predicate, which is similar to the possible test outcome cases. GridEx might model the majority of the feature target space accurately, but false negative predictions might result from a proportion of the target space, which is poorly represented in the explanation data set.

Implications for CAIPI PHAL (Algorithm 4.6) is an eligible algorithm to be integrated into CAIPI (Teso and Kersting, 2019) in the context of regressions. Even though it is no local explanation algorithm in the strict sense, it might be beneficial for CAIPI in two aspects: First, it captures the regression model's general behavior while containing rules that apply to a specific instance. Hence, PHAL belongs more into the category of global explanation procedures (e.g., Schwalbe and Finzel, 2023), but local explanations can be deduced from it. Second, the predicates in the PHAL surrogate are tailored to the users' needs, as the statistical feature extraction procedures can be arbitrarily substituted, as long as they suit the data type. To generate counterexamples, one can first select the rules that cover an instance – similar to local explanations. The rules can be revised afterwards – similar to the explanation revisions in the **RWR** case. The revised rules partition the unlabeled data set wrt. to the decisive features. Counterexamples would be random samples of the partitioned data set, where the indecisive features are randomly substituted.

Figure 4.22 showcases a possible explanatory interactive regression approach incorporating PHAL within the scope of the running example. Having a regression model selected the most-informative instance, a user evaluates its prediction and the decision-making mechanism depicted as the subset of PHAL rules covering the instance. In this case, the average house prices of districts in Boston are estimated by the districts' average room numbers. A PHAL rule covering the most-informative instance, the district Roxbury, suggests a relation between the 25 percent quartile of rooms and the 25 percent quartile of house prices with a 90 percent probability. The user corrects the association such that the 25 percent room quartile is associated with the 50 percent house price quartile. The user might also be certain about this association, thus modifying the probability to 100 percent. Three unlabeled instances are covered by the revised rule. A sampling procedure generates counterexamples by a random selection of instances from the restricted unlabeled data set. Their addition with random noise scaled by the rule probability prevents overfitting. The counterexamples strengthen the user's perception of the relation associated with the most-informative instance. Note that this kind of explanation revision does not deterministically locally reflect decision-making mechanisms of instances similar to the most-informative instance, yet generally refines the regression model. On the upside, the explanation revision is more flexible, making also generalizations and specifications feasible. The applicability of such an explanatory interactive regression model has not yet been experimentally evaluated, but it appears to be a promising future research direction.

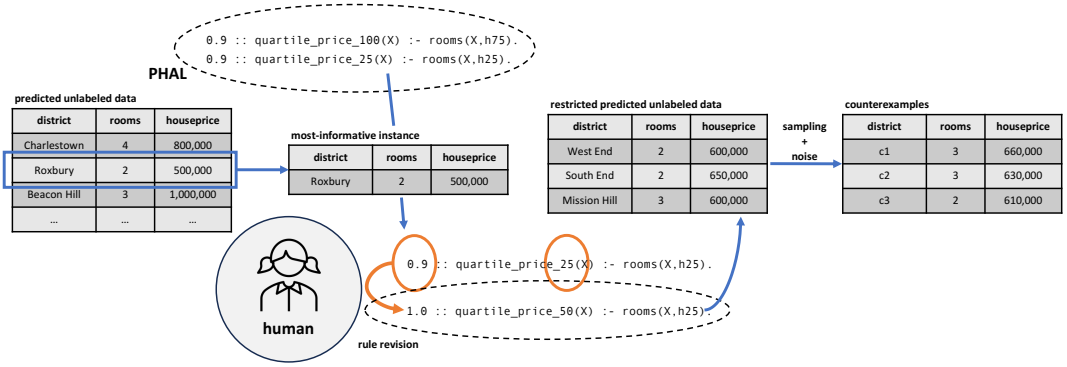

Figure 4.22: Explanatory interactive regression outline. A regression model selects the most-informative instance from the unlabeled data set. PHAL generates a rule set serving as a surrogate for the regression model. A human evaluates the prediction and explanation and revises the applicable rule. The revised rule restricts the unlabeled data set, from which instances are sampled and random noise is added.

Nevertheless, integrating PHAL into CAIPI also has some severe drawbacks: First, a certain proportion of the unlabeled data set must be restrained as an explanation data set. Second, obtaining a surrogate model in each **RWR** iteration is costly and might be inaccurate depending on the available examples, which are determined by the quality of the regression model. Especially in early CAIPI optimization iterations, where the regression model might have a poor predictive quality, positive or negative probabilistic examples might be rare.

Other methods with characteristics akin to PHAL also prove to be highly suitable for CAIPI. For example, integrating a decision tree with an initial abstraction step promises comparable efficacy. Counterfactual explainers (e.g., Guidotti, 2022, and references therein) offer a purely local explanation approach. However, for continuous target variables, establishing an alteration threshold analogous to the target class modification in classification scenarios is required.

4.3.2 Mixed-Data Clustering

The contents of this section are published in Amling et al. (2024) and rephrased. This section uses figures, tables, definitions, examples, and an algorithm of Amling et al. (2024), which are cited accordingly. Appendix C.5 contains detailed information about the author's contribution to the referenced publication.

XIML algorithms such as the ones reviewed in Table 1.1 focus exclusively on supervised ML problems. Consequently, they neglect clustering as the most prominent unsupervised ML task (Madhulatha, 2012). Clustering decomposes a data set into smaller sets with meaningful structures, also termed partitions or clusters. Various clustering algorithms exist with distinct conceptual approaches. For instance, *k*-means (Lloyd, 1982) partitions data sets wrt. representative instances – in this case, the expected instance. Contrary, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) (Ester et al., 1996) searches for density regions in a data space. Hence, different clustering algorithms applied to the same data set might result in a different outcome. Another crucial problem is that clustering benchmarks target primarily spatial data with continuous values (Gagolewski, 2022). They neglect hereby the unique challenges, e.g., for distance metrics, introduced by mixed-data scenarios, which are mostly present in tabular data. Existing post-hoc explanatory ML techniques for clustering are frequently based on surrogate models, e.g., Morichetta,

Casas, and Mellia (2019) use LIME (Ribeiro, Singh, and Guestrin, 2016), which, depending on their faithfulness, might introduce approximation errors and thus inaccurate explanations. Other post-hoc cluster explainers use feature importance values but restrict themselves to a single explanation type (e.g., Scholbeck, Funk, and Casalicchio, 2023), despite a diverse user group might benefit from multimodal explanations. Revisit Amling et al. (2024) for an exhaustive literature review of explanatory ML algorithms for clustering.

Problem Local explanations are a prerequisite for an iterative revision of clustering algorithms by XIML (Teso and Kersting, 2019). For clustering, there exists a research gap in developing post-hoc explanation methods that directly explain partitions without surrogate models or prior knowledge about the clustering algorithm.

Solution Categorical and continuous definitions of the entropy that measures the information concentration lead to local feature importance scores for clusters decomposing a data space with different scales of measurement. The local feature importance score is the basis for various modes of explanations: precisely, cluster-specific prototypes and rules and global feature importance scores for the entity of clusters.

Contribution This section contributes (i) a mathematical derivation of a model-agnostic explanation approach for mixed-data clustering based on local feature importance values as well as (ii) a deduction of various explanation types from the local feature importance scores. Related to this chapter is (iii) the clusterExplainR library[34] implemented in R, which generates multimodal explanations for a clustering process with mixed data.

The explanation types generated by clusterExplainR are visualized by Figure 4.23. Evaluating clustering algorithms is challenging due to the lack of ground truth in real-world applications. Hence, this section compares a subset of the explanation types to established methods on clustering benchmark data sets (Gagolewski, 2022): precisely, the global feature importance score to SHAP (Lundberg and Lee, 2017), and the rules to Cluster Analysis with Multidimensional Prototypes (ClAMP) (Bobek et al., 2022). For additional experiments on real-world data sets, refer to Amling et al. (2024).

Subsequently, the explanation types of clusterExplainR will be derived in a first and evaluated in a second part. The section summary will target the question of how clusterExplainR can be incorporated into a variation of CAIPI for clustering.

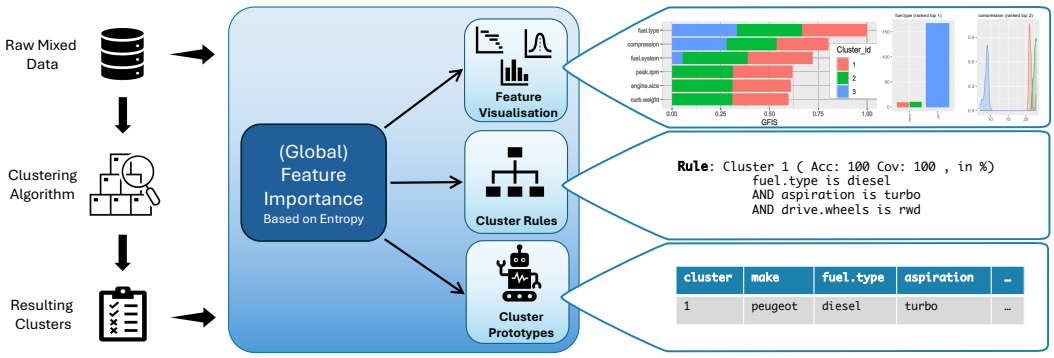

Figure 4.23: clusterExplainR overview (Amling et al., 2024).

[34] https://github.com/imperonas/clusterExplainR, 29 July 2024.

Methods

Clustering is an unsupervised ML task that divides a feature data space into smaller subsets. Similar to other domains of this thesis, mixed-data clustering on tabular data has it own notational conventions formalized in Notation 4.4.

Notation 4.4 (Mixed-Data Clustering (Amling et al., 2024))**.** Let $(x_1, ..., x_i, ..., x_n)^T \in \mathcal{X}$ be an instance of a feature data set with identifier set $\mathcal{F} = \{1, ..., i, ..., n\}$, of which $\mathcal{C} \subseteq \mathcal{F}$ is the subset of categorical and $\mathcal{M} = \mathcal{F} \setminus \mathcal{C}$ is the subset of metric identifiers. Let hard clustering decompose $\mathcal{X}$ into the partitions $\{B^{(1)}, ..., B^{(k)}\} = \mathcal{B}$. Let X denote the random variable behind a feature dimension in $\mathcal{X}$ with a outcome of a possible outcome set $z \in Z$, where X_B and Z_B express the random variable and the set of possible outcome values wrt. partition B. Finally, let $\mathbf{X} = (X_1, ..., X_i, ..., X_n)$ be the sequence of random processes corresponding to the feature dimensions in $\mathcal{X}$.

Definition 4.27 (Entropy (MacKay, 2003; Marsh, 2013; Amling et al., 2024))**.** Let *entropy* measure the information concentration of a feature wrt. its level of measurement. Let there be a categorical and a continuous case:

Shannon (categorical) entropy:
Suppose a discrete random variable X of a feature from $\{x_i | i \in \mathcal{C}, x \in \mathcal{X}\}$. Let $p_z = Pr(X = z)$. Then, the Shannon entropy is:

$$H(X) = -\sum_{z \in Z} p_z \log_2(p_z).$$

Continuous entropy:
Now, suppose a continuous random variable of a standardized feature from $\{x_i | i \in \mathcal{M}, x \in \mathcal{X}\}$ with $Z =]-\infty, \infty[$ and probability density function p. Then, the continuous entropy is defined as:

$$H(X) = -\int p(z) \log_2 p(z) dz.$$

Definition 4.28 (Local Feature Importance Score (Amling et al., 2024))**.** Let the *local feature importance score* ($lFIS$) be defined as follows:

$$lFIS(X, B) = 1 - \min(H(X_B) \cdot H(X)^{-1}, 1).$$

Each explanation type of clusterExplainR leverages entropy (Definition 4.27) as a mathematical concept that quantifies the information concentration of a feature within a corresponding random process (MacKay, 2003). Features that are split heterogeneously across clusters maximize the entropy, while features with homogeneous distributions, e.g., one value is only contained by one cluster, minimize the entropy. Deduced explanations are model-agnostic and mathematically sound, as clusterExplainR accommodates categorical and continuous scales of measurement, both present in tabular data. By comparing the entropy of a feature within a certain cluster to its global entropy, the information concentration of a feature of a cluster is contextualized in relation to the population. A feature equally distributed across the population will have an entropy close to one, whereas the same feature will have an entropy close to zero if it is concentrated in a cluster. In this case, the feature is said to be locally important for the cluster (Definition 4.28).

Example 4.13. Suppose the following clusters partitioning the gender variable of a data set (Amling et al., 2024).

gender	$\mathcal{X}$	$B^{(1)}$	$B^{(2)}$	$B^{(3)}$
female	60%	87.5%	50%	0%
male	40%	12.5%	50%	100%

The local feature importance score for the first cluster is calculated as:

$$H(\texttt{gender}) = -\left(0.6 \cdot \log_2 0.6 + 0.4 \cdot \log_2 0.4\right) \approx 0.97$$
$$H(\texttt{gender}_{B^{(1)}}) = -\left(0.875 \cdot \log_2 0.875 + 0.125 \cdot \log_2 0.125\right) \approx 0.54$$
$$lFIS(\texttt{gender}, B^{(1)}) = 1 - \min\left(\frac{H(\texttt{gender}_{B^{(1)}})}{H(\texttt{gender})}, 1\right) = 1 - \frac{0.54}{0.97} \approx 0.44.$$

The global feature importance score (Definition 4.29) quantifies the impact of a feature on the clustering process in general. It is calculated as the average of all clusters' local feature importance scores.

Definition 4.29 (Global Feature Importance Score (Amling et al., 2024))**.** Suppose k partions $B \in \mathcal{B}$. Then, the *global feature importance score* ($gFIS$) is defined as:

$$gFIS(X, \mathcal{B}) = \left(\sum_{B \in \mathcal{B}} lFIS(X, B)\right) k^{-1}.$$

Example 4.14. Calculating $lFIS$ for the remaining clusters of Example 4.13 results in $lFIS(\texttt{gender}, B^{(2)}) = 0.0$ and $lFIS(\texttt{gender}, B^{(3)}) = 1.0$. Hence, $gFIS(\texttt{gender}, \mathcal{B}) = (0.44 + 0 + 1) \times \frac{1}{3} = 0.48$ (Amling et al., 2024).

The entity cluster matching score (Definition 4.30) assesses the representativeness of the important features of an instance for a cluster. It unifies two components: a component quantifying the general representativeness of the instance and the local feature importance score. Intuitively, the entity cluster matching score normalizes a feature's local importance within a cluster relative to its importance across all clusters, thereby scaling the representative measure for a feature value.

Definition 4.30 (Entity Cluster Matching Score (Amling et al., 2024))**.** Let the *entity cluster matching score* ($ECMS$) be defined as follows:

$$ECMS(x, \mathbf{X}, B) = \frac{\sum_{x_i \in x, X \in \mathbf{X}} m(x_i, X_B) \cdot lFIS(X, B)}{\sum_{X \in \mathbf{X}} lFIS(X, B)}, \text{ where}$$
$$m(x_i, X_B) = \begin{cases} Pr(X_B = x_i) \cdot \max\left(\{Pr(X_B = z) | z \in Z_B\}\right)^{-1} & \text{if } i \in \mathcal{C}, \\ p_{X_B}(x_i) \cdot \max\left(p_{X_B}(z)\right)^{-1} & \text{otherwise,} \end{cases}$$

including the assumption that the outcomes of p_{X_B} for x_i and z are feasible.

Prototypes (Definition 4.31) are representative instances per cluster. The instance of a cluster that maximizes the entity cluster matching score can be considered as the prototypical instance.

Definition 4.31 (Prototype (Amling et al., 2024))**.** Let $\arg\max_{x \in B} ECMS(x, \mathbf{X}, B)$ return the *prototype* of partition B.

Example 4.15. Consider the example of Amling et al. (2024). Suppose the random variable `weight` with realization 75kg. Further, assume that $lFIS(\texttt{weight}, B^{(1)}) = 0.33$ with $p_{\texttt{weight}_{B^{(1)}}}(75) = 0.05$ and $\text{mode}(p_{\texttt{weight}_{B^{(1)}}}) = 0.05$. Finally, assume a single woman (`w`) and a single man (`m`) from cluster $B^{(1)}$, both with `weight` = 75kg. Then $ECMS$ can be computed by:

$$ECMS(\texttt{w}, (\texttt{gender}, \texttt{weight}), B^{(1)}) = \frac{1.0 \cdot 0.44 + 1.0 \cdot 0.33}{0.44 + 0.33} = 1.0,$$
$$ECMS(\texttt{m}, (\texttt{gender}, \texttt{weight}), B^{(1)}) \approx \frac{0.1429 \times 0.44 + 1.0 \times 0.33}{0.44 + 0.33} \approx 0.51.$$

Hence, the instance (`w`) is the chosen prototype for cluster $B^{(1)}$.

Finally, the local feature importance score can also be used to generate rules that describe a cluster (Algorithm 4.7). The rules are logical conjunctions of ranges for continuous features and feature values for categorical features. A rule set, in this case, are conjunctions of predicates for each feature. The RULESEARCH algorithm starts with the most general rule (line 1). It refines the rule as long as false positives exist (line 2). It selects the most important feature of a cluster (line 5) and evaluates randomly sampled proposal rules (line 6) according to a rule selection heuristic (Definition 4.32) wrt. the F1-score. The selected proposal rule extends the preliminary rule if it generates at least one true negative (line 7). Otherwise, the feature is omitted and the procedure continues as long as features are available. Figure 4.24 is a graphical representation of the RULESEARCH algorithm.

Definition 4.32 (Rule Search Heuristic (Amling et al., 2024))**.** Let each rule $r \in R$ decompose $\mathcal{X}$ into a partition $B \in \mathcal{B}$ denoted as $B = r(\mathcal{X})$. Let $\text{F1}(r(\mathcal{X}))$ return the corresponding F1-score. Let $R_{B,X}$ be a set of proposal rules for partition B and random variable X. Then, the *rule search heuristic* is: $\arg\max_r \{\text{F1}(r(\mathcal{X})) | r \in R_{B,X}\}$.

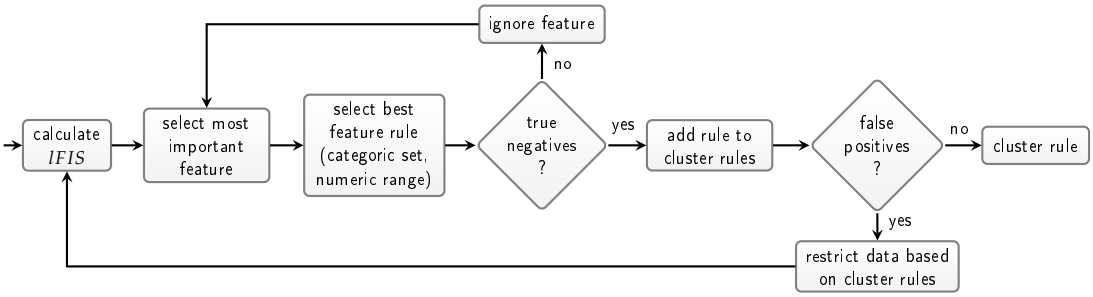

Figure 4.24: clusterExplainR rule search algorithm (Amling et al., 2024).

Example 4.16. Consider the following exemplary rules obtained by Algorithm 4.7 (Amling et al., 2024):

```
Cluster 1 (Accuracy: 80, Coverage: 87.5):
    Gender is  Female
    AND Weight is between  64.4 and 85.1
Cluster 2 (Accuracy: 60, Coverage: 100):
    Weight is between  56.1 and 92.9
Cluster 3 (Accuracy: 100, Coverage: 100):
    Gender is  Male
    AND Weight is between  43.4 and 45.6
```

Algorithm 4.7: RULESEARCH(B, $\mathcal{X}$, $\mathbf{X}$) (Amling et al., 2024)

Input: Partition B, dataset $\mathcal{X}$, sequence of random variables $\mathbf{X}$
Output: Rule set R describing partition B

$R \leftarrow \emptyset$
while $\exists_{x \in R(\mathcal{X})} x \notin B$ **do**
  $r \leftarrow \varnothing; \mathbf{X}' \leftarrow \mathbf{X}$
  while $r = \varnothing \wedge |\mathbf{X}'| > 0$ **do**
    $X \leftarrow \arg\max_X \{lFIS(X, B) | X \in \mathbf{X}'\}$ ▷ *Definition 4.28*
    $r' \leftarrow \arg\max_r \{\mathrm{F1}(r(\mathcal{X})) | r \in R_{B,X}\}$ ▷ *Definition 4.32*
    $r \leftarrow r'$ **if** $r'(\mathcal{X}) \neq \mathcal{X}$ **else** $\mathbf{X}' \leftarrow \mathbf{X}' \setminus X$
  $R \leftarrow R \cup r$
return R

This section has proposed a local feature importance score, which compares the entropy of feature dimensions within a cluster to the overall population. More explanation types have been deduced from the local feature importance score: a global feature importance score assessing the general importance of features for a clustering process, prototypical instances for clusters, and rules describing a cluster.

Experiments

Setup The entropy-based clusterExplainR approach is evaluated on four cluster-specific yet numerical benchmark data sets, named wingnut, mk1, mk2, and isolation (Gagolewski, 2022). All of which define spatial clustering tasks, which Figure 4.25 visualizes. Important to understand is that cluster benchmarks provide a cluster ground truth and, therefore, mimic a perfect clustering outcome without applying a clustering algorithm. Readers interested in evaluations on real-world data sets, which have been clustered manually, are referred to Amling et al. (2024).

This section has contributed the first model-agnostic, entropy-based, multimodal explanatory ML approach for mixed-data clustering. As a consequence, appropriate benchmark methods with the same characteristics as clusterExplainR are scarce. Therefore, the evaluation concentrates on the global feature importance score and the cluster rules and compares them to the well-established methods SHAP (Lundberg and Lee, 2017) and ClAMP (Bobek et al., 2022). SHAP is applied similarly to Definition 4.11 but uses a random forest[35] as a surrogate model, which is trained with the cluster identifiers as labels. ClAMP leverages the Anchor explainer (Ribeiro, Singh, and Guestrin, 2018) on a clustered data set as a surrogate model and generates rules containing bounding boxes with a range in the continuous case and a set of possible values for categorical features. The rules are evaluated in terms of fidelity, now measured by the accuracy and the coverage metric, and simplicity, calculating the average number of rules per cluster and the average number of features per rule. The results can be reproduced using the clusterExplainR R library[36].

Results Figure 4.25 compares RULESEARCH (Algorithm 4.7) to ClAMP (Bobek et al., 2022) and shows a similar result in terms of spatial clustering tasks. Both explainers are capable of finding profound rules for tasks with independent feature dimensions (wingnut, mk1). The rules of both algorithms do not suffice for data

[35] `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`, 30 July 2024.
[36] `https://github.com/imperonas/clusterExplainR`, 29 July 2024.

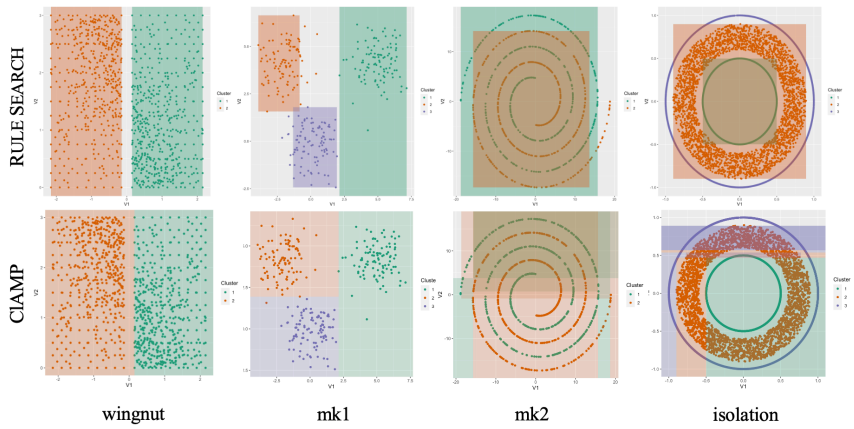

Figure 4.25: clusterExplainR rule-based cluster explanations (Amling et al., 2024).

sets with correlated feature dimensions (mk2, isolation). This conveys also in the accuracy and coverage metrics of each cluster, where RULESEARCH tends to have minor benefits even for benchmark data sets with correlated feature dimensions in comparison to ClAMP (Table 4.11). Interesting is the complexity evaluation, where RULESEARCH by definition (Algorithm 4.7) generates a single rule per cluster. Yet, also the average number of features per cluster point to simplicity advantages of RULESEARCH compared to ClAMP (Table 4.12).

The global feature importance score behaves similarly to SHAP (Figure 4.26) in terms of the feature ranking and the attribution magnitude. The only exception is the isolation data set, where SHAP retrieves an almost uniform attribution over the clusters for each feature and the global feature attribution score concentrates almost its entire attribution magnitude to the first cluster for each feature.

Table 4.11: clusterExplainR rule fidelity evaluation (Amling et al., 2024). Comparison of the fidelity between RULESEARCH and ClAMP measured by the **Accuracy** and **Coverage** metrics for each **Cluster** and **Data** set.

		RULESEARCH		ClAMP	
Data	**Cluster**	**Accuracy**	**Coverage**	**Accuracy**	**Coverage**
wingnut	Cluster 1	100%	100%	100%	100%
	Cluster 2	100%	100%	100%	100%
mk1	Cluster 1	100%	100%	100%	100%
	Cluster 2	97%	91%	99%	98%
	Cluster 3	98%	94%	99%	100%
mk2	Cluster 1	51.3%	99.8%	49%	99%
	Cluster 2	58.2%	96.2%	49%	98%
isolation	Cluster 1	98.7%	100%	66%	90%
	Cluster 2	53%	99.9%	69%	30%
	Cluster 3	-	-	65%	33%

Table 4.12: clusterExplainR rule complexity evaluation (Amling et al., 2024). Comparison of the rule complexity between RULESEARCH and ClAMP by the average number of rules per cluster (∅ **Rules**), and the average number of features per rule (∅ **Features**) across **Data** sets.

	RULESEARCH		ClAMP	
Data	∅ **Rules**	∅ **Features**	∅ **Rules**	∅ **Features**
wingnut	1.0	1.0	4.0	3.25
mk1	1.0	1.67	3.33	3.5
mk2	1.0	2.0	5.0	3.5
isolation	1.0	2.0	5.0	3.93

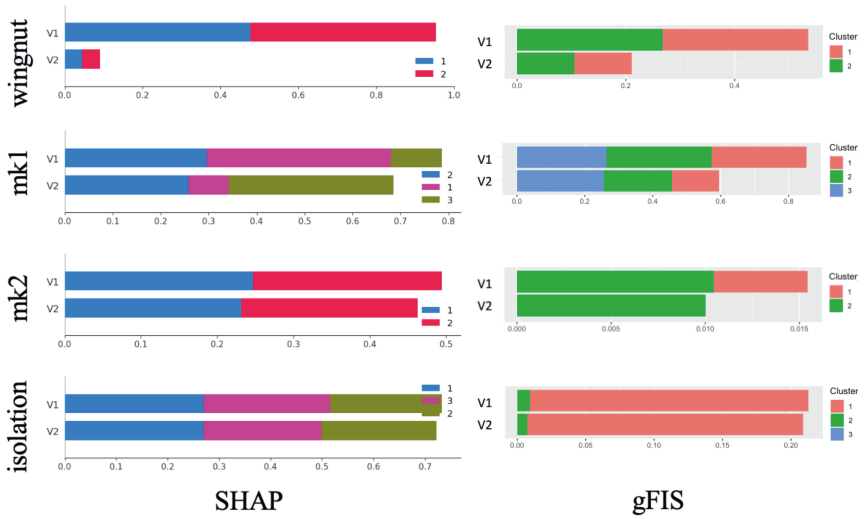

Figure 4.26: clusterExplainR global feature importance score SHAP comparison (Amling et al., 2024).

Section Summary

Summary The clusterExplainR framework proposes multiple explanation types – in particular, local and global feature importance scores, prototypes, and rules – which are mathematically derived solely from the information concentration of features in clusters. Hence, clusterExplainR is a mathematically sound, model-agnostic, and multimodal explainer, which specifically targets the challenges related to mixed data. The evaluation shows that explanations derived from clusterExplainR behave similarly to state-of-the-art explanation strategies, which is why clusterExplainR is a profound, novel, and beneficial way to target the opaqueness frequently encountered in unsupervised mixed-data clustering.

Limitations The evaluation setup reveals the necessity for particular mixed-data clustering benchmarks. The evaluation in Amling et al. (2024) suggests a trade-off between validated benchmark data, which are frequently continuous spatial clustering tasks, and complex mixed-data clustering tasks, where even state-of-the-art clustering approaches encounter problems when trying to find stable solutions. Despite the latter is no problem for the application of explanatory ML techniques, the noisy results might amplify in surrogate models of XAI techniques, making a stable experimental evaluation challenging. It can be expected that this drawback will diminish if mixed-data clustering benchmarks and more model-agnostic explanatory ML approaches for mixed-data clustering become available.

The clusterExplainR framework faces some open challenges: First, the obtained rules are unsuitable to capture feature interactions (Figure 4.25). Second, the tendency of the categorical entropy to converge to zero is higher than for the continuous entropy (Definition 4.27). Hence, categorical features might be favored over continuous features. The discrepancy between categorical and continuous feature dimensions is a known issue to distance metrics, which combine both scales of measurement (e.g., Gower, 1971). Finally, RULESEARCH (Algorithm 4.7) tends to produce specific rules with small intervals in the continuous case (Figure 4.25, mk1), potentially counteracting the interpretability and generalizability.

Implications for CAIPI In practice, even if clustering is considered as being an unsupervised ML task, it is a highly interactive approach where practitioners might incrementally revise the feature set or adapt the parameters to obtain a stable yet meaningful clustering result. Whereas the numerical stability can be measured, the meaningfulness of clusters is context-dependent. Consider a scenario where a data set with thousands of features shall be explored by clustering. Based on distance metrics, the most-heterogeneous cluster can be identified – the cluster equivalent of the most-informative instance. The local feature importance score estimates the relevance of a feature in the most-heterogeneous cluster. A practitioner might possess the knowledge that a high importance value has erroneously been associated with a specific feature. The feature weight for the instances in the most-heterogeneous cluster can be incrementally decreased to have a context-specific adjustment, which maximizes the expected overall stability improvement. This setup can be used, for instance, to counteract potential biases in clusters similar to Heidrich et al. (2023). In general, the feedback loop in the clustering case needs to be more sophisticated than the added counterexamples by CAIPI (Teso and Kersting, 2019) in the classification case. The overarching goal of clustering is to split a high-dimensional data set into meaningful subsets. An additional increase in complexity appears to be inviable, potentially having severe drawbacks for the computational efficacy of clustering algorithms. Therefore, the incremental weight adjustment is a promising feedback opportunity, which is worth exploring in the future.

Figure 4.27 further illustrates a potential integration of the derived local feature importance score into an explanatory interactive clustering framework. Envision a data space consisting of the features weight and gender similar to the running example of this section and suppose the necessity to derive meaningful subsets of the data space by clustering. Assume that a clustering algorithm has found three clusters: two clusters partitioning instances wrt. their weight and one heterogeneous cluster, which can be considered as the most-informative cluster according to the enrolled interaction framework. The local feature importance score attributes a slightly elevated importance value to the gender feature in the most-informative cluster. A human practitioner has the knowledge to identify the instances' gender as being indecisive in the specific context of the most-informative cluster. The user revises the explanation, resulting in a weight reduction of the gender variable of the instances that belong to the annotated cluster for the clustering algorithm. Figure 4.27 puts forward that local weight adaptations by explanation revisions are beneficial for obtaining stable and meaningful partitions of the data space. Two clusters partitioning instances according to their weight appear to be meaningful subsets for the data space in the specific context of the outlined example. The evaluation of this hypothesis is a promising future research direction, expanding the applicability of XIML even further.

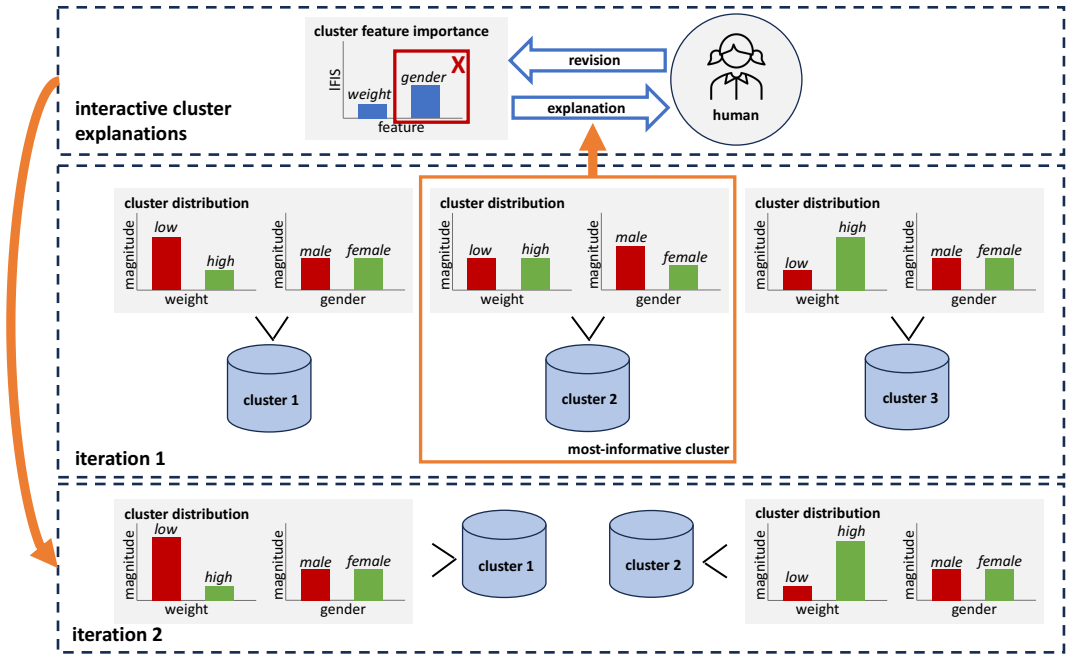

Figure 4.27: Explanatory interactive clustering outline. An iterative clustering procedure is capable of selecting the currently most-informative cluster. Its explanation – here, the local feature importance score – is presented to a human user who identifies the gender feature as being indecisive. The explanation revision refines the clustering process incrementally and ultimately finds more expressive and stable clusters.

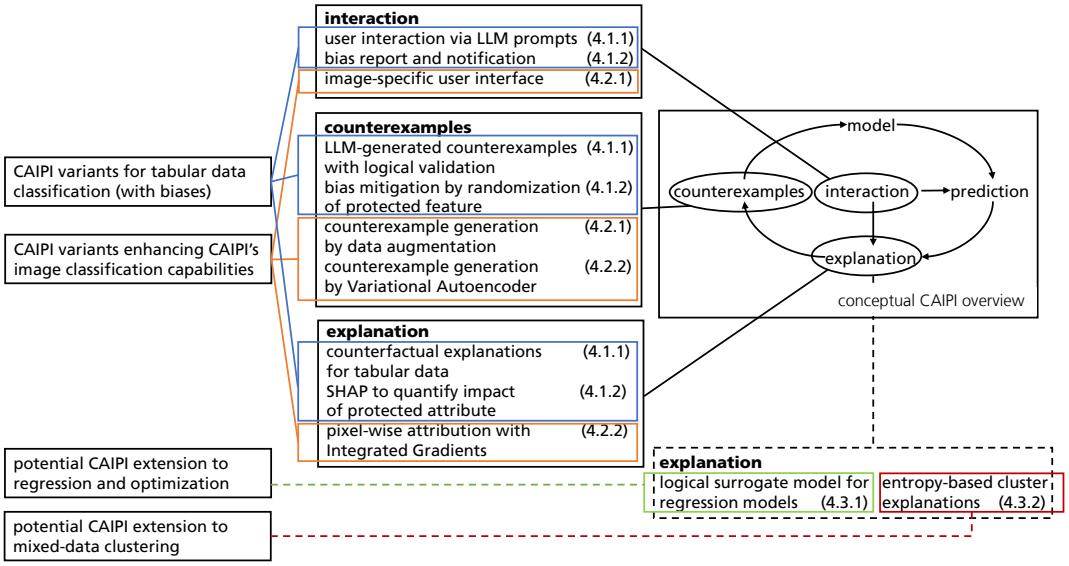

Figure 4.28: Summary of CAIPI adaptations.

4.4 Chapter Summary

Figure 4.28 summarizes the CAIPI component adaptations discussed in this chapter. Based on the conceptual CAIPI overview (right), the adaptations are listed and color-coded (middle): Orange belongs to adaptations that in summary improve CAIPI's performance in its original, primary application area image classification (Teso and Kersting, 2019). Encoded in blue are modifications that expand CAIPI's applicability towards tabular data classification tasks (left). Furthermore, this chapter has introduced two explanation frameworks, that potentially broaden CAIPI's application spectrum to regression, optimization, and clustering (bottom), color-coded green and red. The latter adaptations need to be paired with modifications of the interaction and counterexample generation, or more broadly feedback injection, components. Each modification contains section numbers in brackets. Please revisit the section summaries for each adaptation's specific derivations, results, and limitations. Note that the prediction component has not been modified at this stage. Its adaptation will be addressed in the next chapter, which, in combination with the results from this chapter, will answer **R2**.

In summary, this chapter has proposed CAIPI variants that either surpass CAIPI (Teso and Kersting, 2019) in pre-defined aspects of image classification tasks or transform CAIPI to tabular data classification tasks, a previously unconquered application area of CAIPI. Moreover, the proposed regression, optimization, and clustering explanation frameworks expand CAIPI's applicability even further when paired with the outlined interaction and feedback injection components. All methods have been rigorously derived and evaluated, leading to the conclusion that this chapter enhances CAIPI both in terms of novel application domains and improved performance metrics.

Chapter 5

Catastrophic Feedback Forgetting

The contents of this chapter have been published in Slany, Scheele, and Schmid (2024c). This chapter contains materials – precisely, figures, tables, definitions, and algorithms –, which are cited accordingly. The text has been rephrased. Refer to Appendix C.6 for an author contribution statement for the referenced publication.

ML models have proven as being effective tools for the medical domain (Qayyum et al., 2021). The corrigibility of the model's predictions, the ability to revise the model's decision-making mechanism, and thus enriching the optimization phase with the physician's annotations are crucial for a trustworthy application of ML in the medical domain (Holzinger, 2016). All prerequisites are met by XIML such as shown by Slany et al. (2022), who apply CAIPI (Teso and Kersting, 2019) to the classification of CT images. The previous chapter has transferred CAIPI to various ML tasks, such as the classification of tabular data like in the following scenario:

Consider the use case of assisting physicians during diagnosis, specifically in assessing the diabetes risk of patients, which serves as the running example in this chapter (Figure 5.1). Suppose a corrigible ML model in both regards, the prediction and the decision-making mechanism, optimized by CAIPI (Teso and Kersting, 2019). Imagine the model associates Marie, a female patient with a high body mass index (BMI), with a high diabetes risk. A counterfactual explainer alters Marie's gender, resulting in a low diabetes risk. The physician observes that an equivalent male patient with a high BMI is not prevented from having a high diabetes risk, prompting the inference that the ML model's decision boundary is erroneously influenced by the gender variable. The physician's explanation revision generates counterexamples that randomize the gender feature while maintaining the relationship between a high BMI and a high diabetes risk. Statistical learning frameworks such as optimizing a ML model with CAIPI might encounter a subsequent iteration in which a novel female patient with high BMI, Sandra, is again correctly predicted to have a high diabetes risk. Yet, the counterfactual explainer might reveal the identical decision making error for Sandra like previously for Marie. Apparently, the physician's revision has not prevented the error in the decision-making mechanism. Dealing with sensitive domains such as medical diagnosis, this behavior of CAIPI might have severely dangerous consequences if left undetected. This phenomenon, a form of catastrophic forgetting particular to XIML, is termed catastrophic feedback forgetting in this chapter (Slany, Scheele, and Schmid, 2024c).

Problem CAIPI (Teso and Kersting, 2019) is prone to catastrophic feedback forgetting. Counterexamples alone do not ensure that the user feedback on the decision-making mechanism persists throughout the CAIPI optimization cycle, expressed formally by Remark 5.1.

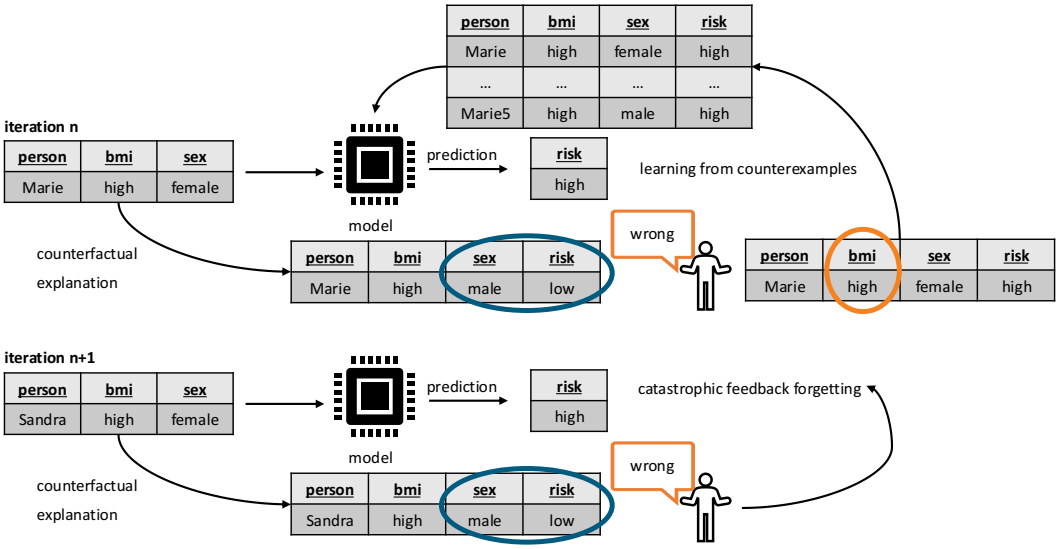

Figure 5.1: HYXIML problem statement.

Remark 5.1. The induction of an underproportional amount of counterexamples wrt. a feature data set causes catastrophic feedback forgetting in the sense that

$$\lim_{\frac{|\mathcal{X}'|}{|\mathcal{X}|}\to 0} \mathcal{X} \cup \mathcal{X}' = \mathcal{X}.$$

Even if the induced amount of counterexamples is sufficient, statistically learned models do not guarantee that (i) the decision-making mechanism of counterexamples is learned as intended and (ii) the decision-making mechanism of a single counterexample is preserved over the course of XIML iterations.

Remark 5.1 reveals, in relation to Remark 3.1, a deadlock scenario, where an underproportional amount of counterexamples is not guaranteed to adjust the decision boundary – causing catastrophic feedback forgetting. Yet, an overproportional amount of counterexamples corrupts the training data set, leading to catastrophic forgetting. The latter has drastic consequences for the training in general – inhibiting generalizing ML models –, which is why catastrophic feedback forgetting must be mitigated without additionally increasing the mass of counterexamples.

Solution Predictions of statistical ML models are accompanied by inferences from probabilistic logic rules. The user annotations in explanation revisions are conserved in logical programs from which rule sets are induced. Given a distance metric, logical inferences substitute ML predictions for instances similar to instances that have received explanation revisions. Highly precise rules ensure that novel yet similar instances receive the same decision-making mechanism, which conserves the user feedback and thus counteracts catastrophic feedback forgetting.

Contribution This chapter contributes HYXIML (Figure 5.2) – a hybrid XIML approach, the first for medical diagnosis according to the literature review (Table 1.1). HYXIML extends the traditional CAIPI framework (Teso and Kersting, 2019) by probabilistic logic inferences with rules induced by PROBFOIL$^+$ (Raedt et al., 2015). Using the Gower distance (Gower, 1971), an appropriate distance metric for tabular data that accommodates categorical and continuous features, similar instances, such as the ones for which the user has revised the explanation, are identified among novel

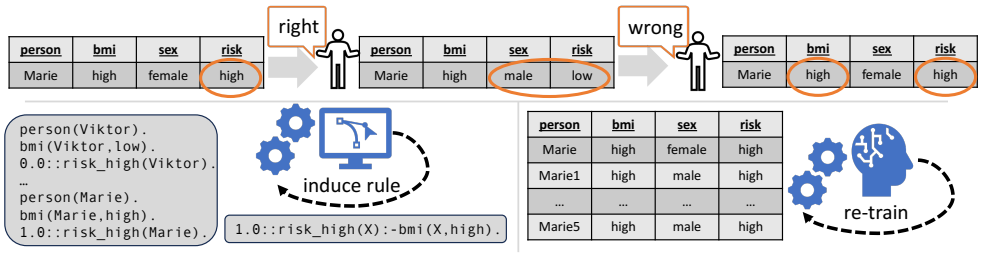

Figure 5.2: HYXIML overview (Slany, Scheele, and Schmid, 2024c). HYXIML depicts the decision-making mechanism by counterfactual explanations. In **RWR** iterations, HYXIML adapts the decision boundary by counterexamples from explanation revisions and conserves the user annotations in a logical program. Logical inferences overrule ML predictions for instances similar to ones that have received explanation revisions.

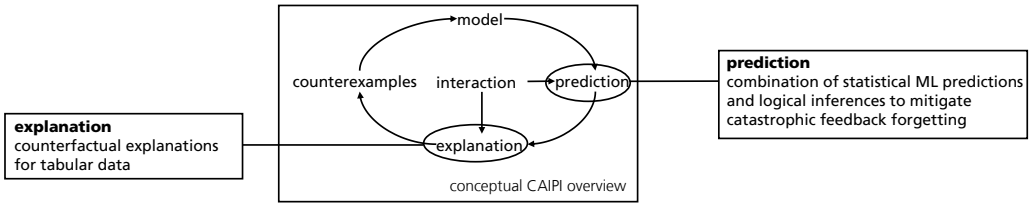

Figure 5.3: HYXIML component adaptations.

instances. Logical inferences substitute ML predictions for similar instances. The implementation of HYXIML is available in a GitHub repository[37].

In comparison to CAIPI (Teso and Kersting, 2019), HYXIML substitutes the explanation component with counterfactual explanations (Wachter, Mittelstadt, and Russell, 2017) (Figure 5.3). Prominent and novel in HYXIML is the adjustment of the prediction component, which combines statistical ML predictions with logical inferences given a distance metric.

Subordinated research questions The research questions (Slany, Scheele, and Schmid, 2024c) compare HYXIML to CAIPI in its ability to unlearn a spurious correlation on two tabular data sets from the medical domain. They also take the predictive performance into account because the correct decision-making mechanism is conditioned on correct predictions, as this thesis supposes the mechanism behind wrong predictions to be incorrect.

R2.14 Which is superior for unlearning a spurious correlation, HYXIML or CAIPI?

R2.15 Does HYXIML compromise the predictive performance?

This chapter contains subordinated research questions. It continues the consecutive enumeration from the previous chapter. In the chapter summary, first, the contents of this section will be summarized and discussed, during which the subordinated research questions will be answered. This chapter concludes with an answer to the second main research question of this thesis for which it also considers the results of the previous chapter. Apart from a summary section, this chapter is split into two sections, one that derives and one that evaluates the HYXIML approach.

[37] `https://github.com/emanuelsla/HybridXIML/`, 01 August 2024.

5.1 Hybrid Explanatory Interactive Machine Learning

This section formalizes a training and a prediction algorithm, denoted as TRAIN (Algorithm 5.1) and PREDICT (Algorithm 5.2). The prediction component of HYXIML accommodates probabilistic logic inferences and ML predictions and therefore requires an algorithmic specification.

The prerequisites for the training algorithm have already been defined over the course of this thesis. In the domain of Notation 2.1, a statistical ML model, specifically a random forest (Breiman, 2001), is incrementally trained to select the most-informative instance (Definition 3.9). Similarly to previous sections that have trained a random forest within variations of the CAIPI framework (e.g., Section 4.1.1), HYXIML depicts the decision-making mechanism by a counterfactual explainer (Definition 3.11). The novelty of HYXIML is that the user feedback is conserved in probabilistic logic examples to induce a set of probabilistic logic rules with PROBFOIL$^+$ (Raedt et al., 2015). Probabilistic logic and probabilistic examples have also been defined in the Definitions 4.1 and 4.3 and can be put into the context of the domain medical diagnosis by the following examples:

Example 5.1. Assuming that Marie is the first instance in the data set ($n = 1$) and her BMI is the only decisive feature, her probabilistic example is given as follows (Slany, Scheele, and Schmid, 2024c):

$$\texttt{instance(i\_1).} \qquad \texttt{bmi(i\_1,high).} \qquad 1.0{::}\,\texttt{risk\_high(i\_1).}$$

Remark 5.2. In addition to Definition 4.3, only decisive features $i \in v$ (Definition 2.3) are incorporated into probabilistic examples to ensure that the induced probabilistic rule set contains only the relation between decisive features and the target.

Example 5.2. Using probabilistic examples such as the one in Example 5.1 and PROBFOIL$^+$ (Raedt et al., 2015), a probabilistic logic rule set can be obtained. Suppose that the following rule is the only rule in R (Slany, Scheele, and Schmid, 2024c):

$$1.0{::}\,\texttt{risk\_high}(X)\texttt{:-bmi}(X,\texttt{high}).$$

The former rule expresses that a high BMI certainly causes a high diabetes risk.

HYXIML will be evaluated in its ability to unlearn an artificial spurious correlation (Definition 4.6). The counterexample generation procedure is equivalent to its basic formalization (Definition 2.5). All regular features are treated as being decisive; the spurious correlation feature is said to be indecisive. Hence, counterexamples repeat the instance but replace the spurious correlation with random draws from $\mathcal{Y}$.

The components are assembled to HYXIML's training algorithm (Algorithm 5.1). Its skeleton is equal to the basic CAIPI pipeline (Algorithm 2.1): It trains a model (line 3), selects the most-informative instance (line 4), evaluates its prediction (line 6) and, if it is predicted correctly, also its explanation (line 10). The XIML outcome case handling in terms of the labeled data set is also identical: In case of wrong predictions, the instance appends the labeled data set with its corrected label (line 7). Instances with correct predictions based on the correct reasons are added to the labeled data set without further actions (line 11). Correctly predicted instances with erroneous decision making are added together with counterexamples to the labeled data set (line 13). At the end of each iteration, the currently most-informative instance is removed from the unlabeled data set (line 17). The training procedure of HYXIML extends the **RWR** case: The user annotations are conserved in a probabilistic logic program (line 15), from which a set of probabilistic logic rules is induced (line 16).

Algorithm 5.1: TRAIN($\mathcal{L}, \mathcal{U}, c, n$) (Slany, Scheele, and Schmid, 2024c)

Input: Data sets $\mathcal{L}$ and $\mathcal{U}$, number of counterexamples c, iteration budget n
Output: Model f, set of probabilistic rules R, example feature set $\mathcal{X}_E$

1: $E \leftarrow \emptyset; \mathcal{X}_E \leftarrow \emptyset$
2: **for** $1 : n$ **do**
3: $\quad f \leftarrow \text{FIT}(\mathcal{L})$ ▷ *Notation 2.1*
4: $\quad m \leftarrow \text{MII}(f, \mathcal{U})$ ▷ *Definition 3.9*
5: $\quad \hat{y}^{(m)} \leftarrow f(x_{\mathcal{U}}^{(m)})$
6: $\quad$ **if** $\hat{y}^{(m)} \neq l(x_{\mathcal{U}}^{(m)})$ **then**
7: $\quad\quad \mathcal{L} \leftarrow \mathcal{L} \cup \{(x_{\mathcal{U}}^{(m)}, l(x_{\mathcal{U}}^{(m)}))\}$ ▷ *Case:* ***W***
8: $\quad$ **else**
9: $\quad\quad \bar{x} \leftarrow \text{EXP}(f, x_{\mathcal{U}}^{(m)})$ ▷ *Definition 3.11*
10: $\quad\quad$ **if** $\bar{x}_{\text{spurious}} = x_{\mathcal{U}_{\text{spurious}}}^{(m)}$ **then** ▷ *Definition 4.6*
11: $\quad\quad\quad \mathcal{L} \leftarrow \mathcal{L} \cup \{(x_{\mathcal{U}}^{(m)}, \hat{y}^{(m)})\}$ ▷ *Case:* ***RRR***
12: $\quad\quad$ **else**
13: $\quad\quad\quad \mathcal{L} \leftarrow \mathcal{L} \cup \{(x_{\mathcal{U}}^{(m)}, \hat{y}^{(m)})\} \cup \{\text{GEN}(x_{\mathcal{U}}^{(m)}, \hat{y}^{(m)}, c)\}$ ▷ *Case:* ***RWR****, Def. 2.5*
14: $\quad\quad\quad \mathcal{X}_E \leftarrow \mathcal{X}_E \cup \{x_{\mathcal{U}}^{(m)}\}$
15: $\quad\quad\quad E \leftarrow E \cup \{\text{TOPROBEX}(x_{\mathcal{U}}^{(m)}, \hat{y}^{(m)}, \phi)\}$ ▷ *Definition 4.3*
16: $\quad\quad\quad R \leftarrow \text{INDUCE}(\phi, E)$ ▷ *Raedt et al. (2015)*
17: $\quad \mathcal{U} \leftarrow \mathcal{U} \setminus x_{\mathcal{U}}^{(m)}$
18: **return** $f, R, \mathcal{X}_E$

The rule set serves as an additional prediction tool and is therefore returned additionally to the optimized model (line 18). The subsequent prediction step requires knowledge about the instances that have undergone an explanation revision. Such instances are aggregated in a data set (line 14), which is also returned.

Definition 5.1 (Logical Inference (Kimmig et al., 2011; Slany, Scheele, and Schmid, 2024c))**.** Let $P_S(q|R)$ be the success probability of a query q wrt. R (Definition 4.2). Let a procedure INFER take R and x as input and return a *logical inference* $\hat{y} = y$ corresponding to the target predicate if $P_S(q|R) > \alpha$, where α is a decision threshold, and $\hat{y} = \bar{y} \neq y$ otherwise.

Definition 5.2 (Unscaled Gower Distance (Gower, 1971; Slany, Scheele, and Schmid, 2024c))**.** Let the procedure DIST take a feature vector x and any other feature vector x^{other} as input and return the *unscaled Gower distance*, which is defined as follows:

$$\Delta(x, x^{\text{other}}) = \frac{|x_{\mathcal{M}}|}{|x|} \sqrt{\sum_{i \in \mathcal{M}} (x_i - x_i^{\text{other}})^2} + \frac{|x_{\mathcal{C}}|}{|x|} \left(1 - \frac{\sum_{i \in \mathcal{C}} \mathcal{I}_{[x_i = x_i^{\text{other}}]}}{|x_{\mathcal{C}}|}\right),$$

Let $\mathcal{M} \subseteq \mathcal{F}$ and $\mathcal{C} = \mathcal{F} \setminus \mathcal{M}$ be locally defined subsets of metric and categorical identifiers. The indicator function $\mathcal{I}$ returns 1 if its condition is met and 0 otherwise.

Algorithm 5.2: PREDICT(x, f, R, $\mathcal{X}_E$, δ) (Slany, Scheele, and Schmid, 2024c)

Input: Instance x, model f, set of probabilistic rules R, example feature set $\mathcal{X}_E$, threshold δ
Output: Prediction $\hat{y}$

1: $\hat{y} \leftarrow \text{INFER}(R, x)$ **if** $\min(\{\text{DIST}(x, x_E) | x_E \in \mathcal{X}_E\}) < \delta$ **else** $f(x)$ ▷ *Defs. 5.1, 5.2*
2: **return** $\hat{y}$

The prediction component of HYXIML (Algorithm 5.2) decides between a logical inference (Definition 5.1) and a ML prediction based on a distance metric (Definition 5.2). If the minimal distance between a novel instance and the instances in the collected explanation revision feature set is smaller than a pre-defined threshold, a logical inference substitutes the ML prediction. If the success probability (Definition 4.2) for a query given a probabilistic rule set exceeds a preliminary defined decision threshold, the instance receives the positive label, otherwise the negative label. HYXIML leverages the Gower distance (Gower, 1971), which combines similarity measures for categorical and continuous variables.

In summary, HYXIML strengthens the importance of the user's annotations by collecting the explanation revisions in a probabilistic logic program. Highly precise probabilistic logic rules overrule ML predictions of novel instances, similar to the instances that have undergone an explanation revision. This way, it can be ensured that novel yet similar instances receive the user-intended decision-making mechanism. Thus, HYXIML is designed to effectively counteract catastrophic feedback forgetting, otherwise present in CAIPI (Teso and Kersting, 2019).

5.2 Experimental Evidence for a Hybrid Approach

Setup HYXIML is evaluated in comparison to CAIPI (Teso and Kersting, 2019) in its ability to unlearn a spurious correlation (Definition 4.6) on two tabular data sets from the medical domain, Diabetes and Diagnostic[38]. HYXIML replaces ML predictions by logical inferences given a similarity measure using the Gower distance (Definition 5.2). A superiority of HYXIML over CAIPI implies a positive impact of including logical inferences. This can most easily be assessed by directly comparing ML predictions and logical inferences during the CAIPI optimization phase.

During five experimental iterations, a random forest with balanced class weights[39] is trained with CAIPI with 250 optimization iterations and five counterexamples per **RWR** iteration for each of the two data sets. The **RWR** data set is extracted in each tenth iteration, from which a set of probabilistic logic rules is induced. A grid search within an interval ranging from 0.25 to 5.0 in 0.25 increments determines the optimal Gower distance threshold. The decision threshold is set to 0.5. All experiments start with 100 labeled instances with features and targets randomly sampled from each variable's set of possible values. The goal is to start with an uninformative model such that optimization results can be attributed to CAIPI or HYXIML. Figure 5.4 depicts the experimental setup. The GitHub repository[40] contains the code to reproduce the experiments.

Two comments on the experimental setup: First, random forests are rule-based models. A baseline study has found that random forests are a viable but not an optimal choice for the investigated data sets (Table 4.1). Depending on the definition (e.g. Azevedo, Rocha, and Pereira, 2024), a connection of two rule-based approaches, even if one is statistical and the other is logical, might not be a hybrid approach in the strict sense. This section still refers to the approach as *hybrid* as it accommodates statistical and logical optimization (Rüden et al., 2023; Azevedo, Rocha, and Pereira, 2024). Moreover, it needs to be pointed out that HYXIML is model-agnostic

[38] Diabetes: `https://www.kaggle.com/datasets/akshaydattatraykhare/diabetes-dataset`, Diagnostic: `https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data`, 01 August 2024.

[39] `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`, 01 August 2024.

[40] `https://github.com/emanuelsla/HybridXIML/`, 01 August 2024.

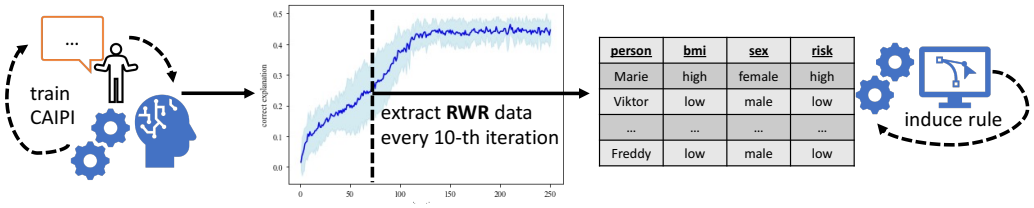

Figure 5.4: HYXIML experimental setup (Slany, Scheele, and Schmid, 2024c). Instances, which have been predicted correctly with erroneous decision-making mechanism, are withheld and extracted in each tenth CAIPI optimization iteration. Probabilistic logic rules are induced from each extracted data set.

such that the random forest can be replaced with other ML algorithms such as neural networks. Second, the experiments do not address the amount of counterexamples. This has been done in several previous works (Teso and Kersting, 2019; Slany et al., 2022; Slany, Scheele, and Schmid, 2024a) with high transferability to this evaluation. Instead, the experiments in this chapter focus on the distance threshold evaluation, which is a novel component for model-agnostic XIML.

Results Table 5.1 compares the weighted averages of precision, recall, and ratio of correct explanations given correct predictions between PROBFOIL$^+$ inferences and ML predictions on a subset of the test data sets determined by the optimal Gower distance threshold obtained by a grid search. Early-stage results are the optimum values within the first 50 CAIPI optimization iterations. Late-stage results are the optimum values within iterations 51 to 250. Probabilistic examples contain only the relation between decisive features and the target (Remark 5.2). Hence, they always utilize the correct decision-making mechanism and have, therefore, a deterministically perfect ratio of correct explanations. Regarding the predictive performance, there tend to exist slight benefits in the first 50 optimization iterations, an effect that diminishes in the later stage when the ML model has been provided with more training data. The decision-making mechanism behind ML predictions also improves in the later CAIPI optimization phase when more counterexamples from explanation revisions have been added. In summary, probabilistic logic inferences are constructed to be superior in utilizing the correct decision-making mechanism. Their predictive capability is not strictly lower compared to ML predictions. Probabilistic logic inferences are valuable in earlier optimization iterations.

Figure 5.5 provides graphical insights into the role of the Gower distance threshold on the Diabetes data set. It reveals that the predictive performance behaves mostly similar for logical inferences and ML predictions, where the latter tends to have minor benefits. Remarkable is that the anticipated drop in the predictive performance of logic inferences does not occur, even when applying the rules on all test data without a similarity restriction. For the explanatory performance, probabilistic logic inferences are superior to ML predictions and obtain a perfect solution given a sufficient amount of instances in the logic program to obtain a stable rule set. The positive impact diminishes in the later optimization iterations, where a ML model optimized with CAIPI has an almost equivalent capability of following the correct decision-making mechanism.

Finally, Figure 5.6 shows that the magnitude of the overall differences between HYXIML and CAIPI is small, even if probabilistic logic inferences in HYXIML overrule approximately 25 percent of instances at the end of the optimization cycle. This can be seen as a justification for the utilized experimental framework. It has been demonstrated that logical inferences are especially superior in the early optimization

stage. However, generally, one can expect more prediction corrections in the early and more explanation revisions in the later CAIPI optimization iterations (Heidrich et al., 2023; Slany, Scheele, and Schmid, 2024a). The consequence is that probabilistic logic inferences effectively overrule ML predictions, which might have been already predicted out of the correct reasons. Nevertheless, the HYXIML approach is beneficial for weak ML models and for explanation revisions, which seldomly occur but are important to be conserved, such as in medical diagnosis.

Table 5.1: ML predictions and logical inferences (Slany, Scheele, and Schmid, 2024c). Comparison of the weighted averages (Ø) of **Precision**, **Recall**, and ratio of correct explanations (**Corr. Expl.**) on two medical **Data** sets across the early (iteration <= 50) and late (iteration >50) **Stage** CAIPI optimization phase for an optimal Gower distance threshold δ.

Data	δ	**Stage**	**Ø Precision** PROBFOIL$^+$	ML	**Ø Recall** PROBFOIL$^+$	ML	**Ø Corr. Expl.** PROBFOIL$^+$	ML
Diabetes	1.5	early	0.6649	**0.6737**	**0.7326**	0.7022	**1.0000**	0.4716
			(0.4023)	(0.3988)	(0.3547)	(0.3402)	(0.0000)	(0.4009)
		late	0.6445	**0.6874**	0.7199	**0.7232**	**1.0000**	0.7780
			(0.2745)	(0.2485)	(0.2162)	(0.2004)	(0.0000)	(0.2688)
Diagnostic		early	**0.9091**	0.7273	**0.9091**	0.5455	**1.0000**	0.6250
			(0.3015)	(0.4671)	(0.3015)	(0.4539)	(0.0000)	(0.5175)
	0.75	late	0.7229	**0.8920**	**0.7700**	0.7438	**1.0000**	0.8263
			(0.4083)	(0.2893)	(0.3620)	(0.3258)	(0.0000)	(0.3340)

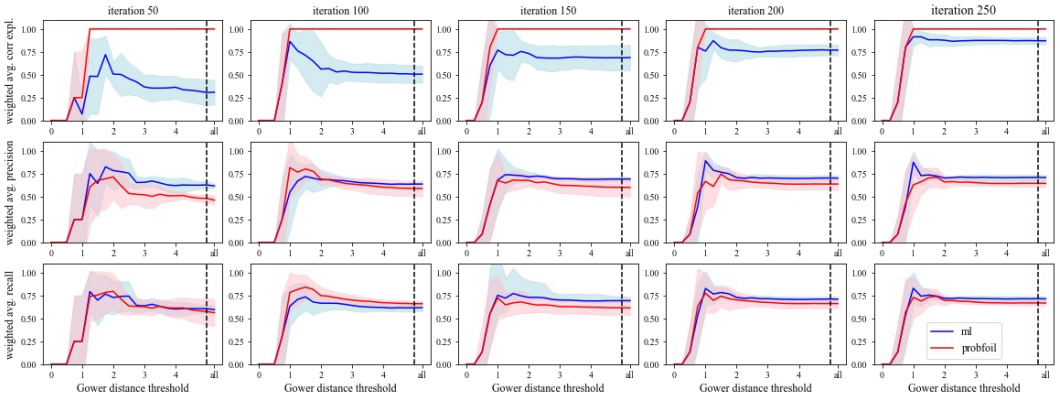

Figure 5.5: ML predictions and logical inferences on Diabetes (Slany, Scheele, and Schmid, 2024c). Comparison of the weighted averages of correct explanations, precision, and recall for ML predictions and logical inferences across CAIPI optimization iterations on various proportions of the test data set determined by the Gower distance.

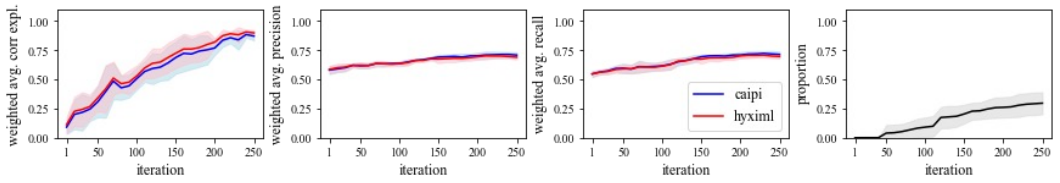

Figure 5.6: HYXIML CAIPI comparison on Diabetes (Slany, Scheele, and Schmid, 2024c).

5.3 Chapter Summary

Summary ML models for sensitive and safety-critical use cases such as medical diagnosis must be corrigible in their predictive behavior and their decision-making mechanism (Holzinger, 2016; Slany et al., 2022). It is crucial that user annotations persist over the course of an interactive optimization cycle such as CAIPI (Teso and Kersting, 2019). In other words, if a physician once revises an explanation to express that specific symptoms imply a particular disease, it must be guaranteed that the model does not conduct the identical decision making error in the future. Errors in the model's decision-making mechanism that re-occur despite being corrected are termed catastrophic feedback forgetting. It is shown that counterexamples added during CAIPI optimization in the context of statistical ML models alone do not prevent catastrophic feedback forgetting (Table 5.1).

HYXIML is a hybrid XIML approach, combining probabilistic logic inferences and statistical ML predictions. Optimizing a ML model with CAIPI, it collects the explanation revisions in a probabilistic logic program to induce highly precise rules by PROBFOIL$^+$ (Raedt et al., 2015) that conserve the user's explanation revisions. Tabular-data use cases frequently combine features with several scales of measurement. The Gower distance (Gower, 1971) accommodates distance measures for categorical and continuous variables and estimates the similarity between novel instances and instances that have undergone an explanation revision. Probabilistic logic inferences overtake the prediction phase in HYXIML for instances with a high similarity. Statistical ML predictions are used for the remaining instances.

The experimental results show that probabilistic logic inferences, which are constructed to utilize decisive features exclusively, are strictly superior to ML predictions in terms of the correct decision-making mechanism (Table 5.1, Figure 5.5). ML predictions, on the contrary, have slight benefits regarding the predictive performance and tend to utilize the correct decision-making mechanism after a sufficient amount of explanation revisions. Overall, the effect of a hybrid approach that includes probabilistic logic inferences into CAIPI is marginal yet noticeable and necessary in the context of medical diagnosis as the user knowledge persists even if ML models are not saturated or explanation revision are unique (Figure 5.6).

Answers to subordinated research questions The experimental evidence leads to the following answers to the research questions formulated in this chapter (Slany, Scheele, and Schmid, 2024c):

R2.14 Which is superior for unlearning a spurious correlation, HYXIML or CAIPI?
Given a random forest and two tabular data sets for medical diagnosis, HYXIML *is superior to* CAIPI *for unlearning a spurious correlation.*

R2.15 Does HYXIML compromise the predictive performance?
In the context of the experimental evaluation, HYXIML *does not compromise the predictive performance.*

Related Results Similar to Slany et al. (2022), who have classified CT images, HYXIML focuses on the medical domain but on tabular data for medical diagnosis instead of images. As has been done in Slany et al. (2022) and Slany, Scheele, and Schmid (2024a), HYXIML is a CAIPI (Teso and Kersting, 2019) variant that generates multiple, precisely five, counterexamples per **RWR** iteration. In contrast to the referenced

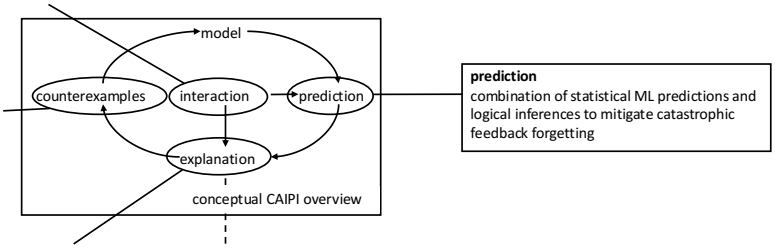

Figure 5.7: Extension of CAIPI adaptations summary.

publications, it does not explicitly evaluate the amount of counterexamples. HYXIML is a hybrid approach similar to knowledge-informed ML methods that post-process or replace ML predictions by logical inferences (Rüden et al., 2023). Unique for HYXIML is that it incrementally optimizes two models within a single CAIPI optimization cycle. Apart from the ML model, it induces a set of probabilistic logic rules from a probabilistic logic program with examples containing the user's explanation revisions in each **RWR** iteration. HYXIML is also the first CAIPI variant with a specific prediction algorithm (Algorithm 5.2), which decides between a ML prediction and a logical inference based on a similarity measure. In this regard, the evaluation of HYXIML fosters unique experiments in the XIML literature, as it conducts a grid search to identify the optimal distance threshold.

Limitations The broader implications of HYXIML need also to be set into context to some conceptual and experimental limitations:

- **Conceptual limitations**:
  By applying PROBFOIL$^+$ (Raedt et al., 2015), the computational complexity increases, leading to the question of whether the benefits of probabilistic logic are worth a reduced computational efficacy. A possible hypothesis states: Probabilistic rule learning is suitable because a probabilistic decision-making mechanism similar to statistical ML can be modeled with a small amount of data in the explanation revision feature set. This hypothesis has not been evaluated. This chapter has been motivated by the use case medical diagnosis with tabular data. An open question is whether and how the HYXIML framework can be transferred to other data types such as image data. A possible solution is to extract the labels of features from images either with classical image processing techniques or ML models. The labels might then be transformed into probabilistic examples to induce a set of rules (e.g., Manhaeve et al., 2018). Regarding the experimental results, Figure 5.6 reveals an overall low improvement magnitude of using HYXIML compared to CAIPI (Teso and Kersting, 2019). The beneficial properties of HYXIML have been thoroughly discussed. Using a probabilistic background theory in addition to induced rules promises to improve HYXIML especially in expert-knowledge-intensive tasks such as medical diagnosis.
- **Experimental setup**:
  Even though the experiments contain some additional evaluations such as for the optimal distance threshold, they lack equally important evidence: For instance, more ML models and other probabilistic rule learning frameworks can be incorporated into a more sophisticated evaluation that considers tabular

data sets beyond the medical domain and other data types. The implementation of HYXIML[41] is capable of categorical classifications, although only binary classification tasks have been evaluated. This leads to further conceptual challenges – in particular, revised definitions to construct probabilistic examples in the categorical case.

Answer to research question In conclusion, this chapter has provided a novel, additional CAIPI adaptation compared to Figure 4.28. Figure 5.7 puts the modification of the prediction component by HYXIML into the context of the conceptual CAIPI overview (Figure 4.28), indicating that now all CAIPI components mandatory in the sense of **R2** have been modified, which leads to the following answer:

R2 How do modifications in the prediction, explanation, interaction, and counterexample generation components affect CAIPI 's applicability to ML tasks?
Within the context of the different experimental evaluations of this thesis, CAIPI*'s* ***explanation*** *and* ***counterexample generation*** *components have been adapted such that (i)* CAIPI*'s capability for correct decision making for image classification tasks has been improved (Section 4.2.2). Furthermore, by adapting the aforementioned components, (ii)* CAIPI *has been transferred to tabular data classification tasks (Chapter 3, Section 4.1). In regard to classification scenarios, this thesis has presented a modification of the* ***prediction*** *component (iii) to overcome catastrophic feedback forgetting, which had been identified as a vulnerability for* CAIPI *(Chapter 5). For both domains, image and tabular data classification, (iv) user interfaces that allow human users without ML expertise to* ***interact*** *with* CAIPI *have been proposed (Chapter 3, Section 4.2.1). Additionally, this thesis has presented and evaluated explanation algorithms connected to possible human revision and feedback injection approaches that (v) potentially expand* CAIPI *to regression and optimization as well as to clustering tasks (Section 4.3). Overall, the component modifications have enhanced* CAIPI*'s applicability in both regards: They have increased* CAIPI*'s capabilities in its original primary application area image classification and expanded* CAIPI*'s application spectrum by conquering novel ML tasks.*

[41] `https://github.com/emanuelsla/HybridXIML/`, 01 August 2024.

Chapter 6

Conclusion

6.1 Summary

Model-agnostic XIML (Teso and Kersting, 2019) combines the interactive and iterative corrigibility of model decisions by coactive learning (Shivaswamy and Joachims, 2015) with revisable local explanations (e.g., Schwalbe and Finzel, 2023). CAIPI (Teso and Kersting, 2019) is a model-agnostic XIML framework, which leverages LIME (Ribeiro, Singh, and Guestrin, 2016) as a local explainer and injects the user feedback in the form of counterexamples that emerge from explanation revisions. CAIPI is tailored to image and text classification. The traditional evaluation objective of CAIPI for image data is unlearning a spurious correlation, which has been induced by colored decoy pixels correlated with the class label. Counterexamples are identical images with randomized decoy pixel color.

This thesis has identified four **limitations** in the context of CAIPI (Teso and Kersting, 2019): First, CAIPI's original formalization limits CAIPI's practical applicability as, for instance, the proposed counterexample generation procedure for images cannot be transferred to real-world image classification scenarios. Second, a fundamental theoretical motivation for XIML in general is that human users without ML expertise are put into the position to train ML models, corrigible in their predictions and revisable in their decision-making mechanism. The original publication does not contribute user interfaces, counteracting the motivational proposition. Third, the original formalization is not built upon mathematical definitions such that some components, e.g., the outcome case distinction, remain unspecified. Finally, related to the third limitation, is the inability to investigate theoretical research questions such as how counterexamples affect the optimization process, as the original publication does not exhaustively specify the components.

Driven by the identified shortcomings, this thesis has made two key **contributions**: First, all algorithms proposed by this thesis are built upon a strict mathematical formalization, which has allowed this thesis to explore theoretical concepts related to model optimization with model-agnostic XIML methods learning from counterexamples. This thesis has connected the procedural formalization of algorithms like utilized by Teso and Kersting (2019) with grounded mathematical concepts – every procedure is formally defined. According to the review of the literature in the XIML field, this approach to algorithmic formalizations is unique (Table 1.1). Specifically, this thesis has found that counterexamples are only beneficial for instances similar to instances in the reference data set. For others, an overproportional amount of counterexamples harms the model optimization with CAIPI variants. It can be shown that counterexamples increase the risk of catastrophic forgetting. Second, this thesis has proposed several CAIPI variants that either improve CAIPI's performance in one of its traditional application areas, image classification, or expand CAIPI's application spectrum beyond images or text. The major contributions

of this thesis are related to innovative ways to generate counterexamples in the image and tabular data classification context. For instance, generative ML approaches have been used in both domains. An important contribution is enhancing the statistical ML prediction component with probabilistic logic inferences. Probabilistic logic programs preserve user explanation revisions. By substituting ML predictions with logical inferences for instances similar to annotated ones, the risk of catastrophic feedback forgetting – a potential vulnerability of CAIPI– is mitigated.

Both contributions correspond to the overarching **research questions** of this thesis. The first one has been evaluated mathematically and experimentally. Several subordinated research questions in specific sections of this thesis have thoroughly assessed the second one. Their findings have been aggregated to a final answer.

R1 How do counterexamples affect the optimization of ML models?
Given random forests, binary classification tasks, and tabular data, counterexamples improve the probability that the decision-making mechanism is adjusted. This does not necessarily cause an improvement in the predictive performance nor for the classifier's ability to follow the correct decision-making mechanism (Chapter 3).

R2 How do modifications in the prediction, explanation, interaction, and counterexample generation components affect CAIPI 's applicability to ML tasks?
Within the context of the different experimental evaluations of this thesis, CAIPI*'s* ***explanation*** *and* ***counterexample generation*** *components have been adapted such that (i)* CAIPI*'s capability for correct decision making for image classification tasks has been improved (Section 4.2.2). Furthermore, by adapting the aforementioned components, (ii)* CAIPI *has been transferred to tabular data classification tasks (Chapter 3, Section 4.1). In regard to classification scenarios, this thesis has presented a modification of the* ***prediction*** *component (iii) to overcome catastrophic feedback forgetting, which had been identified as a vulnerability for* CAIPI *(Chapter 5). For both domains, image and tabular data classification, (iv) user interfaces that allow human users without ML expertise to* ***interact*** *with* CAIPI *have been proposed (Chapter 3, Section 4.2.1). Additionally, this thesis has presented and evaluated explanation algorithms connected to possible human revision and feedback injection approaches that (v) potentially expand* CAIPI *to regression and optimization as well as to clustering tasks (Section 4.3). Overall, the component modifications have enhanced* CAIPI*'s applicability in both regards: They have increased* CAIPI*'s capabilities in its original primary application area image classification and expanded* CAIPI*'s application spectrum by conquering novel ML tasks.*

6.2 Discussion

The intermediate summary parts of this thesis have discussed the limitations related to specific CAIPI adaptations, including their theoretical concepts and experimental evaluations, which is why the purpose of this section will be three-fold: First, it will put the main findings of this thesis into a broader context. Second, it will identify the main limitations of this thesis. Third, it will terminate with the question: What could have been done better, if the research topic could be explored again?

In a broader spectrum, this thesis is located in the research field model-agnostic XIML and conducts its contributions by modifying the CAIPI algorithm (Teso and Kersting, 2019). CAIPI is the original algorithm in the research area and induces the human feedback into the model by means of counterexamples, which are additional training data that contain only the correlation between a human annotation of the decision-making mechanism and the target. XIML, in general, is meant to

put humans without ML expertise into the position to train ML models and hereby maximize their control over the underlying ML model. The resulting model is supposed to conduct the correct decisions based on the correct decision-making mechanism from a human user's perspective. Such a behavior is especially important in domains where ethical considerations are relevant such as the Credit example (Section 4.1.2) or in safety-critical domains like medical diagnosis (Chapter 5).

An additional medical use case explored by this thesis is the classification of CT images (Section 4.2.1) – an example, where learning from counterexamples might be inefficient. Especially, state-of-the-art image classifiers like transformer models (e.g., Dosovitskiy et al., 2021) require a large amount of training data. User annotations and counterexamples for a small subset of instances have a small impact on the entire optimization procedure. These are exactly cases, where model-specific XIML procedures (e.g., Schramowski et al., 2020) have benefits, yet still do not solve the issue of costly and perhaps too seldom user annotations. A possible enhancement of model-agnostic XIML is replacing counterexamples with local weight adaptations. Both solutions are beyond the scope of this thesis but illustrate that CAIPI variants – or, more general, *learning from counterexamples* algorithms – are restricted to use cases, where labeled data sets are rare and the human domain expertise is high.

Although a major goal of this thesis is a thorough mathematical formalization, it still has formal limitations. Sometimes, methods are only partially derived. For instance, apart from the random forest (Breiman, 2001), k-means (Lloyd, 1982), and the Variational Autoencoder (Kingma and Welling, 2014; Doersch, 2016), no ML model has been mathematically defined. This thesis occasionally utilizes mathematical assumptions or simplifications. Examples are the erroneous association of categorical variables and the l2-norm (Chapter 3) or the point value retrieval of continuous probability distributions without a specification of the approximation algorithm (Section 4.3.2). Especially challenging is the exploration of many ML domains, all having different notational conventions. The solution has been to establish different notational domains. A superior solution would be to obtain a unified and consistent notation, which also holds across the domains. The experimental evaluation of this thesis can be improved and is occasionally incomparable between sections. Even though the utilized data sets are mostly consistent, even if some sections use only a subset compared to others, the experimental setup is widely deviating. For instance, Chapter 3 controls the entire data-generating process, Section 4.2.1 uses human annotations, and Chapter 5 unlearns spurious correlations. Despite all experimental setups can be justified individually, their difference hinders the comparability between results. A unified experimental evaluation holds a major improvement potential for this work.

In general, many improvement capabilities can be attributed to the circumstance that the work for this thesis has started with various CAIPI adaptations, all making their individual small contributions such as a user interface (Slany et al., 2022) or substituting the counterexample generation procedure (Slany, Scheele, and Schmid, 2024a). Over the course of the publications, theoretical questions have led to the major contributions of this thesis, such as the general impact of counterexamples on the optimization process (Chapter 3) or the hybrid prediction approach to mitigate catastrophic feedback forgetting (Chapter 5). Evaluating this work from the current standpoint, the thesis would have benefited from an initial mathematical formalization of CAIPI and a thorough investigation of its optimization framework. Afterwards, this thesis should have structurally explored with a consistent experimental setup the necessary CAIPI adaptations that transfer CAIPI to novel application areas and improve hereby its theoretical limitations and practical performance.

6.3 Open Potentials

While each adaptation and evaluation of CAIPI (Teso and Kersting, 2019) in relation to this thesis can be improved, as the intermediate limitations paragraphs indicate, there exist three future research directions which appear to be particularly promising: This thesis postulates that counterexamples are beneficial for model optimization for some instances but not for others. However, this thesis does not provide a set of axioms describing the contexts of this finding. Moreover, the interesting question in this context is the optimal amount of counterexamples given an instance, which is one potential future research direction. Catastrophic feedback forgetting is a potential vulnerability in the context of XIML and causes revisions of the decision-making mechanism to diminish during the optimization phase. This thesis has proposed a solution that combines probabilistic logic and statistical ML. The proposed framework, however, still contains some shortcomings such as higher computational costs or a poor transferability to other ML tasks like image classification. Overcoming the identified limitations appears to be especially relevant for concepts acting against catastrophic feedback forgetting in the future. Finally, this thesis has not gone the step beyond classification scenarios. Especially the research direction of explanatory interactive clustering is promising. Counterexamples are unsuitable for clustering as they further increase the complexity of algorithms constructed to reduce the dimensionality. Distinct approaches, for instance, local and incremental weight adaptations, need to be explored.

Appendix A

Mathematical Appendix

A.1 Proof of Lemma 3.3

Lemma (Decision Trees and Decisive Features)**.** Any split $\theta = (i, \alpha) \in \Theta$ of a decision tree $h(\mathcal{X}, \Theta)$ is more likely to be conducted along decisive features v than on indecisive features. That is: $\forall \theta \in \Theta, Pr_\Theta(i \in v) > Pr_\Theta(i \notin v)$.

Proof. Assume that the feature set can be split into a decisive set $\mathcal{X}_v = \{x_i \text{ if } i \in v | x \in \mathcal{X}\}$ and an indecisive set $\mathcal{X}_{\bar{v}} = \{x_i \text{ if } i \notin v | x \in \mathcal{X}\}$. Further, assume that the absence of a causal relation in the indecisive set can be expressed by a uniform random distribution in the sense that: $\mathcal{X}_{\bar{v}} \sim U_{|\bar{v}|}((a, b)^{|\bar{v}|})$. The impurity function wrt. θ determines the split of a decision tree (the lower the impurity function, the more probable a split). It suffices to show that the impurity function wrt. any proposal θ, is expected to be lower for decisive than for indecisive features:

$$\mathtt{E}_\Theta \left[Pr(imp((\mathcal{X}_v \times \mathcal{Y})_\theta) < imp((\mathcal{X}_{\bar{v}} \times \mathcal{Y})_\theta)) \right] > 0.5.$$

The truth of the above statement directly follows from the weak learner condition (Kearns and Valiant, 1994; Schapire, 1990) that states that any machine-learned model on correlated sets $\mathcal{X}_v$ and $\mathcal{Y}$ is superior to random guessing, which is equal to $\mathcal{X}_{\bar{v}}$ and $\mathcal{Y}$. It holds that $\mathtt{E}_{\mathcal{X},\mathcal{Y}} \left[Pr(Pr_\Theta(i \in v) > Pr_\Theta(i \notin v)) \right] > 0.5$. □

A.2 Proof of Proposition 3.1

Proposition (Decision Trees and Counterexamples)**.** A positive number of counterexamples does not decrease the probability that a split of a decision tree is conducted along decisive features. Formally: $\forall c \in \mathbb{N}^+, Pr_{\Theta_c}(i \in v) \geq Pr_\Theta(i \in v)$, where Θ_c is a set of split parameters under the influence of c counterexamples.

Proof. Revisit how different XIML outcome cases affect the training set of a ML model in each iteration. **RRR** and **W** iterations are essentially equal because in **RRR** cases holds that $\hat{y} = y = l(x)$. In **W** iterations, the label is corrected such that $y = l(x)$. The only difference occurs in **RWR** iterations, where counterexamples are added additionally to the most-informative instance.

$$\begin{aligned} \textbf{RWR} \text{ iterations: } & \mathcal{X} \cup \{x\} \cup \mathcal{X}' \times \mathcal{Y} \cup \{y\} \cup \mathcal{Y}', \\ & \text{where } |\mathcal{X}'| = |\mathcal{Y}'| = c \\ \textbf{RRR} \text{ and } \textbf{W} \text{ iterations: } & \mathcal{X} \cup \{x\} \times \mathcal{Y} \cup \{y\} \end{aligned}$$

The probability of any proposal split θ can be determined by the impurity function such that $Pr_{\Theta_c}(i \in v) \geq Pr_\Theta(i \in v)$ corresponds to:

$$imp((\mathcal{X} \cup \{x\} \cup \mathcal{X}' \times \mathcal{Y} \cup \{y\} \cup \mathcal{Y}')_\theta) \leq imp((\mathcal{X} \cup \{x\} \times \mathcal{Y} \cup \{y\})_\theta).$$

The extensive form of the impurity function $imp(\mathcal{Q}_\theta)$ with $\mathcal{Q} = \mathcal{X} \times \mathcal{Y}$ wrt. θ is:

$$imp(\mathcal{Q}_\theta) = \frac{|\mathcal{Q}^{(left)}|}{|\mathcal{Q}|}\left[1 - \left[\left(\frac{1}{|\mathcal{Q}^{(left)}|}\sum \mathcal{I}_{[y=0]}\right)^2 + \left(\frac{1}{|\mathcal{Q}^{(left)}|}\sum \mathcal{I}_{[y=1]}\right)^2\right]\right] + \frac{|\mathcal{Q}^{(right)}|}{|\mathcal{Q}|}\left[1 - \left[\left(\frac{1}{|\mathcal{Q}^{(right)}|}\sum \mathcal{I}_{[y=0]}\right)^2 + \left(\frac{1}{|\mathcal{Q}^{(right)}|}\sum \mathcal{I}_{[y=1]}\right)^2\right]\right].$$

Suppose two cases: a perfect split and a random split. For each of the two cases, the impact of additional instances, either after **RWR** or after **RRR** or **W** iterations, is investigated. Lemma 3.3 states that it is more probable that splits are conducted along decisive features. Each additional instance reflects the decision-making mechanism. Hence, additional instances are supposed to decrease the impurity function.

Let $imp(\mathcal{Q}_\theta)_{\text{perfect}}$ denote the impurity function in the case of a perfect split, where each side contains a single class. It simplifies to:

$$imp(\mathcal{Q}_\theta)_{\text{perfect}} = \frac{|\mathcal{Q}^{(left)}|}{|\mathcal{Q}|}[1 - [1+0]] + \frac{|\mathcal{Q}^{(right)}|}{|\mathcal{Q}|}[1 - [0+1]] = 0$$

RWR:

$$imp(\mathcal{Q}_\theta)_{\text{perfect, }\mathbf{RWR}} = \frac{|\mathcal{Q}^{(left)}| + c + 1}{|\mathcal{Q}| + c + 1}[1 - [1+0]] + \frac{|\mathcal{Q}^{(right)}|}{|\mathcal{Q}| + c + 1}[1 - [0+1]] = 0$$

RRR and **W**:

$$imp(\mathcal{Q}_\theta)_{\text{perfect, }\mathbf{RRR,W}} = \frac{|\mathcal{Q}^{(left)}| + 1}{|\mathcal{Q}| + 1}[1 - [1+0]] + \frac{|\mathcal{Q}^{(right)}|}{|\mathcal{Q}| + 1}[1 - [0+1]] = 0.$$

If a split is perfect, counterexamples have no minimizing impact on the impurity function: $imp(\mathcal{Q}_\theta)_{\text{perfect, }\mathbf{RWR}} = imp(\mathcal{Q}_\theta)_{\text{perfect, }\mathbf{RRR,W}}$ for any $c \in \mathbb{N}_0^+$.

Random splits $imp(\mathcal{Q}_\theta)_{\text{random}}$ are cases, where the classes are equally distributed across the subsets. The impurity function changes to:

$$\begin{aligned} imp(\mathcal{Q}_\theta)_{\text{random}} &= \frac{|\mathcal{Q}^{(left)}|}{|\mathcal{Q}|}\left[1 - \left[\left(\frac{1}{|\mathcal{Q}^{(left)}|}\frac{|\mathcal{Q}^{(left)}|}{2}\right)^2 + \left(\frac{1}{|\mathcal{Q}^{(left)}|}\frac{|\mathcal{Q}^{(left)}|}{2}\right)^2\right]\right] \\ &+ \frac{|\mathcal{Q}^{(right)}|}{|\mathcal{Q}|}\left[1 - \left[\left(\frac{1}{|\mathcal{Q}^{(right)}|}\frac{|\mathcal{Q}^{(right)}|}{2}\right)^2 + \left(\frac{1}{|\mathcal{Q}^{(right)}|}\frac{|\mathcal{Q}^{(right)}|}{2}\right)^2\right]\right] \\ &= \frac{|\mathcal{Q}^{(left)}|}{|\mathcal{Q}|}\left[1 - \left[\left(\frac{|\mathcal{Q}^{(left)}|}{2|\mathcal{Q}^{(left)}|}\right)^2 + \left(\frac{|\mathcal{Q}^{(left)}|}{2|\mathcal{Q}^{(left)}|}\right)^2\right]\right] \\ &+ \frac{|\mathcal{Q}^{(right)}|}{|\mathcal{Q}|}\left[1 - \left[\left(\frac{|\mathcal{Q}^{(right)}|}{2|\mathcal{Q}^{(right)}|}\right)^2 + \left(\frac{|\mathcal{Q}^{(right)}|}{2|\mathcal{Q}^{(right)}|}\right)^2\right]\right] \end{aligned}$$

RWR:

$$imp(\mathcal{Q}_\theta)_{\text{random, }\mathbf{RWR}} =$$
$$\frac{|\mathcal{Q}^{(left)}|+c+1}{|\mathcal{Q}|+c+1}\left[1-\left[\left(\frac{|\mathcal{Q}^{(left)}|+c+1}{2(|\mathcal{Q}^{(left)}|+c+1)}\right)^2+\left(\frac{|\mathcal{Q}^{(left)}|}{2(|\mathcal{Q}^{(left)}|+c+1)}\right)^2\right]\right]$$
$$+\frac{|\mathcal{Q}^{(right)}|}{|\mathcal{Q}|+c+1}\left[1-\left[\left(\frac{|\mathcal{Q}^{(right)}|}{2|\mathcal{Q}^{(right)}|}\right)^2+\left(\frac{|\mathcal{Q}^{(right)}|}{2|\mathcal{Q}^{(right)}|}\right)^2\right]\right]$$

RRR and **W**:

$$imp(\mathcal{Q}_\theta)_{\text{random, }\mathbf{RRR,W}} =$$
$$\frac{|\mathcal{Q}^{(left)}|+1}{|\mathcal{Q}|+1}\left[1-\left[\left(\frac{|\mathcal{Q}^{(left)}|+1}{2(|\mathcal{Q}^{(left)}|+1)}\right)^2+\left(\frac{|\mathcal{Q}^{(left)}|}{2(|\mathcal{Q}^{(left)}|+1)}\right)^2\right]\right]$$
$$+\frac{|\mathcal{Q}^{(right)}|}{|\mathcal{Q}|+1}\left[1-\left[\left(\frac{|\mathcal{Q}^{(right)}|}{2|\mathcal{Q}^{(right)}|}\right)^2+\left(\frac{|\mathcal{Q}^{(right)}|}{2|\mathcal{Q}^{(right)}|}\right)^2\right]\right].$$

For random splits, counterexamples have an optimization impact on the impurity function: $imp(\mathcal{Q}_\theta)_{\text{random, }\mathbf{RWR}} < imp(\mathcal{Q}_\theta)_{\text{random, }\mathbf{RRR,W}}$ for any $c \in \mathbb{N}^+$.

Thus, it can be inferred that $imp(\mathcal{Q}_\theta)_{\mathbf{RWR}} \leq imp(\mathcal{Q}_\theta)_{\mathbf{RRR,W}}$ for any $c \in \mathbb{N}^+$, from which follows that $Pr_{\Theta_c}(i \in v) \geq Pr_\Theta(i \in v)$ for any $c \in \mathbb{N}^+$. □

A.3 Proof of Theorem 3.1

Theorem (Generalization Error and Counterexamples)**.** Considering the entire feature target space $\mathcal{X}, \mathcal{Y}$, there exists a non-zero probability that counterexamples do not reduce the upper bound of the generalization error. That is: $Pr_{\mathcal{X},\mathcal{Y}}(PE_c^* \geq PE^*) \neq 0$, where PE_c^* and PE^* indicate the upper bound of the generalization error in the presence and absence of counterexamples.

Proof. The upper bound of the generalization error of random forests PE^*

$$PE \leq \bar{\rho}(1-str^2)/str^2 = PE^*$$

is determined by the expected inter-tree correlation $\bar{\rho}$ and the expected strength of decision trees within random forests str (Lemma 3.2)

$$\bar{\rho} = \mathrm{E}_{\Theta,\Theta'}[\rho(h(.,\Theta),h(.,\Theta'))]$$
$$str = \mathrm{E}_{\mathcal{X},\mathcal{Y}}[Pr_\Theta(h(\mathcal{X},\Theta)=\mathcal{Y})].$$

To prove Theorem 3.1, it suffices to show that there exist cases where neither the inter-tree correlation decreases nor the strength increases with counterexamples.

Inter-tree correlation:
Counterexamples enforce the correlation between trees

$$\mathrm{E}_{\Theta_c,\Theta_c'}[\rho(h(.,\Theta_c),h(.,\Theta_c'))] \geq \mathrm{E}_{\Theta\Theta'}[\rho(h(.,\Theta),h(.,\Theta'))],$$

which can be justified by the fact that counterexamples are built with the finite set of decisive features of a single instance. Inducing the identical split criteria in each decision tree of an otherwise equal sequence does not lower the inter-tree correlation.

Strength:
Suppose a sample $(x^*, y) \sim \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}'$ is generated from x^*.

$$\lim_{\mathtt{E}[\{\|x,x^*\|_2 | x \in \mathcal{X}\}] \to 0} Pr(\mathtt{E}_{\mathcal{X},\mathcal{Y}}\left[Pr_{\Theta_c}(h(\mathcal{X}, \Theta_c) = \mathcal{Y})\right] \geq \mathtt{E}_{\mathcal{X},\mathcal{Y}}\left[Pr_{\Theta}(h(\mathcal{X}, \Theta) = \mathcal{Y})\right]) = 1$$

$$\lim_{\mathtt{E}[\{\|x,x^*\|_2 | x \in \mathcal{X}\}] \to \infty} Pr(\mathtt{E}_{\mathcal{X},\mathcal{Y}}\left[Pr_{\Theta_c}(h(\mathcal{X}, \Theta_c) = \mathcal{Y})\right] \geq \mathtt{E}_{\mathcal{X},\mathcal{Y}}\left[Pr_{\Theta}(h(\mathcal{X}, \Theta) = \mathcal{Y})\right]) = 0$$

From the preliminary derivation follows that it can not be guaranteed that

$$\mathtt{E}_{\mathcal{X},\mathcal{Y}}\left[Pr_{\Theta_c}(h(\mathcal{X}, \Theta_c) = \mathcal{Y})\right] > \mathtt{E}_{\mathcal{X},\mathcal{Y}}\left[Pr_{\Theta}(h(\mathcal{X}, \Theta) = \mathcal{Y})\right].$$

It can be inferred that none of the subsequent equations hold certainly

$$\mathtt{E}_{\Theta_c,\Theta'_c}[\rho(h(.,\Theta_c), h(.,\Theta'_c))] < \mathtt{E}_{\Theta\Theta'}[\rho(h(.,\Theta), h(.,\Theta'))]$$

$$\mathtt{E}_{\mathcal{X},\mathcal{Y}}\left[Pr_{\Theta_c}(h(\mathcal{X}, \Theta_c) = \mathcal{Y})\right] > \mathtt{E}_{\mathcal{X},\mathcal{Y}}\left[Pr_{\Theta}(h(\mathcal{X}, \Theta) = \mathcal{Y})\right].$$

It follows that $Pr_{\mathcal{X},\mathcal{Y}}(PE_c^* < PE^*) \neq 1$ and thus $Pr_{\mathcal{X},\mathcal{Y}}(PE_c^* \geq PE^*) \neq 0$. □

Appendix B

Additional Results

Plausibility of Counterfactual Explanations

A counterfactual explanation by means of a counterfactual instance (Definition 3.10) is supposed to satisfy two criteria: (i) it is valid – the classification outcome has changed – and (ii) the changes in the feature space are minimal – ideally, only decisive features are altered (Definition 2.3). Consider the example:

Customer: *Why was my credit request rejected?*
Model: *Considering your income, the loan term must be 70 years.*
Customer: *But I am 30 years old. Am I supposed to work until 100?*

Although counterfactual explanations have proven their persuasive power (van der Waa et al., 2021), the initial example shows their drawback wrt. plausibility as it suggests an infeasible action. This issue is an underrepresented problem in the counterfactual explanation literature (Guidotti, 2022). *Plausibility* refers to whether the generated counterfactual instance is realistic or probable given a reference population (Guidotti, 2022). Some counterfactual explainers (e.g., Kanamori et al., 2020) inherently account for the plausibility of counterfactual instances by statistical distance metrics. Others apply distance metrics post-hoc to identify implausible counterfactual instances (Laugel et al., 2019) or to include constraints that increase the plausibility of generated counterfactual explanations (Artelt et al., 2021)[42].

This chapter proposes an alternative approach to increase the plausibility of counterfactual explanations: It is built upon the assumption that a reference population is still only an observed subset of the true population, which is why statistical plausibility post-processors alone might still struggle to identify the plausible among the proposed counterfactual instances. Given this proposition, this chapter proposes logical constraints obtained from human background knowledge. Technically, the internal decision mechanism of a surrogate model – a decision tree – is enriched with logical rules that restrict the set of counterfactuals to plausible instances. The surrogate model ensures that the post-processor is model-agnostic and invariant of the counterfactual explainer.

Problem Counterfactual explanations might be implausible. Existing statistical post-processors bear the risk that the reference population is insufficient to assess the counterfactual explanations' plausibility. Domain experts must have the ability to detect and reject implausible counterfactual explanations.

Solution This chapter enriches a statistical surrogate model with a logical specification to create a post-processor that restricts a set of counterfactual instances to plausible ones only. A decision tree extracted from a classification model is translated

[42]This section is based on joint work with Stephan Scheele. The idea, realization, the experiments, their evaluation, and the writing can entirely be attributed to the author. Exceptions are that Stephan Scheele contributed to the formalization of the method by re-writing, editing, and constructing Figure B.1.

into probabilistic logic rules and extended with logical constraints from background knowledge. If any constraint is violated, the counterfactual instance is rejected.

Contribution This chapter makes two contributions: First, it proposes a model-agnostic and counterfactual-explainer-invariant plausibility post-processor for counterfactual instances and, second, it demonstrates the practicality of post-processing counterfactual explanations regardless of their cardinality.

Methods

The derivations of this chapter belong to the domain of Notation 2.1. Definition B.1 converts the paths of a decision tree that serves as a surrogate model for an arbitrary binary classification model into a probabilistic logic rule set.

Definition B.1 (Decision Tree ProbLog Conversion)**.** Suppose an explanation set $\mathcal{X}_{Exp} \subseteq \mathcal{X}$ similar to Notation 4.3. Let a decision tree h (Definition 3.1) be a surrogate model for an arbitrary binary classification model f such that $h(\mathcal{X}_{Exp}, f(\mathcal{X}_{Exp}))$ postulates the training step. Let CONVERT take a decision tree surrogate model as input and return a probabilistic rule set R (Definition 4.1), where the rule probabilities are the tree's class probabilities. Suppose the following substitution operations to transform the splits of a decision tree into unary probabilistic predicates:

$<\rightarrow$ `l` (lower),	$\leq\rightarrow$ `le` (lower/equal),
$=\rightarrow$ `e` (equal),	$\neq\rightarrow$ `ne` (not equal),
$\geq\rightarrow$ `ge` (greater/equal),	$>\rightarrow$ `g` (greater).

Example B.1. The rule $0.9 :: \texttt{risk\_low}(X)$:- $\texttt{duration\_g\_30}(X)$ expresses that a person has a low credit risk with a probability of 90 percent if its loan duration is greater than 30 years. Suppose that the body predicate has been present as a split criterion in a decision tree, yielding a leaf with a purity of 90 percent.

Logical constraints formulate minimal plausibility conditions. Domain experts know that if these conditions are violated, instances are implausible. Note that also reference data sets can contain implausible instances as outliers might corrupt them. Definition B.2 distinguishes between univariate constraints that apply to a single feature, explicit multivariate constraints that affect multiple features, and implicit multivariate constraints, which affect multiple features such that a condition of one or multiple features is met.

Definition B.2 (Logical Constraint)**.** Suppose that a and b are the lower and upper bounds of an interval $[a, b]$. A *logical constraint* γ for an instance $x \in \mathcal{X}$ is defined as:

- Univariate constraint: $[a, b]_{x_i}$.
- Explicit multivariate constraint: $[a, b]_{x_i}$ if $[a, b]_{x_j}$, where $i \neq j$.
- Implicit multivariate constraint: $[a, b]_{x_i}$, $[a, b]_{x_j}$ such that [condition], where $i \neq j$ and [condition] is a prerequisite.

Let Γ be a finite set of constraints. For simplicity reasons, each $\gamma \in \Gamma$ will be logically encoded as a unary definite predicate.

Example B.2. Each constraint type can be illustrated as follows:

- Univariate: *Customers are at least* 18 *and at most* 100 *years old*: $[18, 100]_{age}$; $\texttt{age\_g\_18\_le\_100}(X)$.
- Explicit multivariate: *If duration* $\leq$ 10 *years, then yearly salary* $>$ \$50,000: $[50000, \infty[_{salary}$ if $[0, 10]_{duration}$; $\texttt{salary\_g\_50k\_IF\_duration\_le\_10}(X)$.

- Implicit multivariate: *The sum of age and duration in years is at most* 80:
  $[\text{contraint}] = x_{age} + x_{duration} \leq 80$; `sum_age_duration_le_80`$(X)$.

Remark B.1. For the sake of brevity, the multivariate cases in Definition B.2 are bivariate. Both sides of the if-conditions can be extended.

A constrained probabilistic rule set (Definition B.3) emanates if the conjunction of body predicates of each rule of a rule set is extended with each logical constraint. Intuitively, each constraint is necessary for each rule to be evaluated successfully. Hence, each constraint has a veto capability.

Definition B.3 (Constrained Probabilistic Rule Set (Raedt, Kimmig, and Toivonen, 2007))**.** A *constrained probabilistic rule set* R_Γ emerges from a ProbLog rule set R (Definition 4.1) by extending each rule $r \in R$ by all constraints $\gamma \in \Gamma$ to yield r_Γ:

$$r_\Gamma = p :: H \text{ :- } B_1, \ldots, B_n, \gamma \mid \gamma \in \Gamma.$$

In the following, suppose the short notation $r_\Gamma = r, \Gamma$ for a constrained rule.

Example B.3. Consider the rule from Example B.1. Example B.2 presents several constraints. Assume that Γ consists solely of `sum_age_duration_le_80`(X). Then:

$$0.9 :: \texttt{risk\_low}(X) \text{ :- } \texttt{duration\_g\_30}(X), \texttt{sum\_age\_duration\_le\_80}(X).$$

In words: A loan duration greater than 30 years yields a low credit risk with a probability of 90 percent as long as the sum of the customer's age and the loan duration in years is smaller than or equal to 80.

The POSTPROCESSOR (Algorithm B.1) first obtains the decision tree surrogate model, which is converted into a probabilistic rule set (l. 2). Each rule is extended with the logical constraints (l. 3). The success probability of each counterfactual instance on the constrained rule set evaluates whether a counterfactual instance reflects the decision-making mechanism of the classification model according to the surrogate model and does not violate a human-defined plausibility constraint (l. 4). As the post-processor takes a set of counterfactual explanations as input, it is invariant from the counterfactual explanation algorithm. Furthermore, the fact that the decision paths of a decision tree are translated into probabilistic logic makes the post-processor model-agnostic.

Figure B.1 visualizes the intuition about plausibility post-processing: First, a counterfactual explanation is obtained using a suitable counterfactual explanation algorithm. The original classification model is transformed into a surrogate model and converted into a probabilistic logic rule set, which is extended by constraints. The logical program performs a sanity check on whether the counterfactual instance yields the counterfactual outcome and ensures that each validity constraint is met. As soon as one constraint is violated, the success probability evaluates to zero and the counterfactual instance is rejected.

Algorithm B.1: POSTPROCESSOR$(\bar{\mathcal{X}}, f, \mathcal{X}_{Exp}, \Gamma)$

Input: Counterfactual data $\bar{\mathcal{X}}$, classifier f, explanation data $\mathcal{X}_{Exp}$, constraints Γ
Output: Counterfactual probabilities $Pr_{\bar{\mathcal{X}}}$

1: $R_\Gamma \leftarrow \emptyset$; $Pr_{\bar{\mathcal{X}}} \leftarrow \emptyset$
2: $R \leftarrow$ CONVERT$(h(\mathcal{X}_{Exp}, f(\mathcal{X}_{Exp}))$ ▷ *Definitions 4.1 and B.1*
3: $R_\Gamma \leftarrow R_\Gamma \cup \{r_\Gamma = r, \Gamma\}$ **for** $r \in R$ ▷ *Definition B.3*
4: $Pr_{\bar{\mathcal{X}}} \leftarrow Pr_{\bar{\mathcal{X}}} \cup \{Pr_{\bar{x}} = Pr_S(H(t(\bar{x}))|R_\Gamma)\}$ **for** $\bar{x} \in \bar{\mathcal{X}}$ ▷ *Definition 4.2*
5: **return** $Pr_{\bar{\mathcal{X}}}$

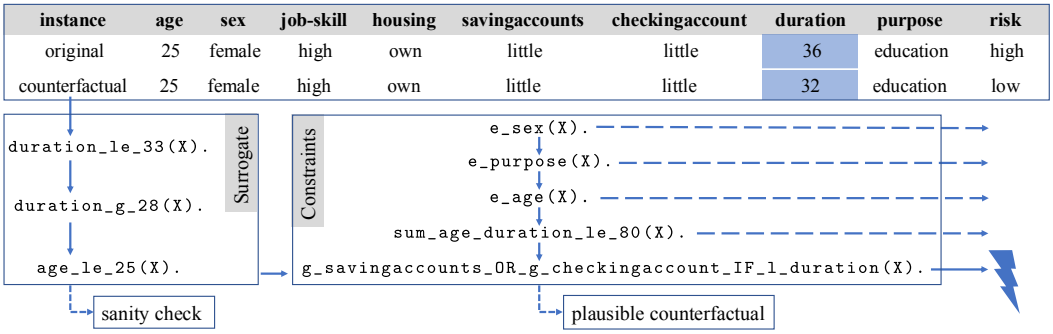

Figure B.1: Plausibility post-processing for counterfactual explanations.

Experiments

Setup This section provides experimental evidence for the post-processing procedure on the German Credit Risk (Credit) and Adult data sets[43] in cases where counterfactuals explainers return multiple counterfactual explanations. The experiments leverage the counterfactual explainer of Mothilal, Sharma, and Tan (2020) such as formalized in Definition 3.11. The amount of queried counterfactual explanations is set to 100 to increase the reliability of the quantitative evaluation. The implementation of the utilized counterfactual explainer[44] is capable of handling actionability constraints. Actionability ensures that some features, e.g., socio-demographic variables like age and gender, cannot be modified (Guidotti, 2022). Those extensions are neglected in the evaluation context, as actionability is a special case of a univariate plausibility constraint (Definition B.2). It can be shown that actionability is inherently part of the post-processor. Credit, which inspires the running example, distinguishes between high and low credit risk. Adult classifies instances along their yearly salary into the categories above and at most $50,000. In Adult, the following feature columns are used: *educationalnum*, *occupation*, *race*, *gender*, and *hoursperweek*. The explanation data and test set ratios are set to 0.25. The classification model is a support vector classifier with radial basis transformer[45]. The presented results are mean values of five experimental iterations with standard deviations in brackets.

The false positive (fp) rates of the classification model are 0.4 (0.04) for Credit and 0.31 (0.02) for Adult. The corresponding false negative (fn) rates are 0.33 (0.12) and 0.22 (0.00). The fidelity estimates (Definition 4.25, without the abstraction step) are: Δ-fp-rates of 0.01 (0.01) for Credit and 0.00 (0.00) for Adult and corresponding Δ-fn-rates of 0.05 (0.04) and 0.01 (0.01).

Table B.1 summarizes the evaluation metrics (Guidotti, 2022). The minimality criterion is estimated by proximity (P) and sparsity (Sp) where the first refers to the cosine distance and the latter counts changed features. Diversity is operationalized by the identical metrics (P_{sim}, Sp_{sim}) but compares the counterfactual instances among each other rather than the counterfactual explanation with the original instance. A 1-Nearest-Neighbor classifier[46] that is supposed to distinguish counterfactual from original instances assesses the discriminative power (DP). The stability (S)

[43] Credit: `https://www.kaggle.com/datasets/kabure/german-credit-data-with-risk`, Adult: `https://www.kaggle.com/datasets/wenruliu/adult-income-dataset`, 06 August 2024.

[44] `https://github.com/interpretml/DiCE/tree/main`, 06 August 2024.

[45] `https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`, 06 August 2024.

[46] `https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.NearestNeighbors.html`, 06 August 2024.

Table B.1: Quality metrics for counterfactual explanations (Guidotti, 2022).

Ratio R**:**	Ratio of returned to queried counterfactuals.
Proximity P**:**	Average cosine similarity of counterfactual explanations to the original instance.
Sparsity Sp**:**	Average ratio of unmodified features in counterfactual explanations compared to the original instance.
Discriminative Power DP**:**	Accuracy of 1-Nearest-Neighbor classifier, distinguishing counterfactual explanations from the original instance wrt. 100 test instances.
Actionability A**:**	Average ratio of actionable features among the modified features.
Proximity Similarity P_{sim}**:**	Average proximity among the counterfactuals.
Sparsity Similarity Sp_{sim}**:**	Average sparsity among the counterfactuals.
Stability S**:**	Comparison of average cosine distances of counterfactual explanations generated from most similar original instances.
Time T**:**	In seconds.

is measured by the similarity between the counterfactual instances generated from the original and its most similar instance. The metrics are enriched by the ratio of queried to returned counterfactuals (R) and the the time (T) in seconds.

Finally, to control the evaluation of the post-processor, the subsequent list of data-set-specific handcrafted constraints is added:

Credit:

$$\texttt{e\_age}(X).\ \texttt{e\_sex}(X).\ \texttt{e\_purpose}(X).\ \texttt{sum\_age\_duration\_le\_80}(X).$$
$$\texttt{g\_savingaccounts\_OR\_g\_checkingaccount\_IF\_l\_duration}(X).$$

Adult:

$$\texttt{e\_gender}(X).\ \texttt{e\_race}(X).\ \texttt{hoursperweek\_g\_0\_le\_70}(X).$$
$$\texttt{g\_educationalnum\_IF\_l\_hoursperweek}(X).$$

For instance, the constraint $\texttt{e\_age}(X)$ expresses that age is immutable, and $\texttt{sum\_age\_duration\_le\_80}(X)$ expresses that the sum of age and duration must be less than or equal to 80. Example B.2 illustrates their construction in more detail.

Table B.2: Evaluation of plausibility post-processor. Comparison of the presence and absence of the post-processor (**post.**) for counterfactual explanations on two **Data** sets wrt. predefined quality metrics (Table B.1). Standard deviations are given in brackets.

Data	**post.**	R	P	Sp	DP	A	P_{sim}	Sp_{sim}	S	T
Credit	True	0.11	0.90	0.74	0.92	1.00	0.96	0.76	0.96	4.21
		(0.07)	(0.03)	(0.03)	(0.05)	(0.00)	(0.02)	(0.05)	(0.01)	(2.62)
	False	1.00	0.88	0.79	0.92	0.95	0.97	0.74	0.97	0.56
		(0.00)	(0.03)	(0.02)	(0.04)	(0.02)	(0.01)	(0.03)	(0.01)	(0.04)
Adult	True	0.09	0.99	0.56	0.58	1.00	0.99	0.74	0.99	29.57
		(0.07)	(0.01)	(0.07)	(0.20)	(0.00)	(0.01)	(0.08)	(0.00)	(5.84)
	False	0.92	0.87	0.54	0.70	0.93	0.99	0.56	0.99	4.05
		(0.17)	(0.17)	(0.15)	(0.26)	(0.02)	(0.00)	(0.12)	(0.00)	(1.40)

Results Table B.2 highlights two drawbacks of the post-processor: First, post-processing does not guarantee the queried amount of counterfactual explanations. The second drawback is a drastically increased computational time. The major benefit of the post-processor is that it has a perfect actionability value. The remaining results are nearly stable with two exceptions: First, the discriminative power is lower for post-processed counterfactual instances in the Adult setting, whereas, second, the sparsity similarity among the counterfactuals is higher.

Section Summary

Summary Counterfactual explanations have emerged as an intuitive interpretation technique for ML models and represent a state-of-the-art local explanation approach in the field of XAI (Wachter, Mittelstadt, and Russell, 2017; Guidotti, 2022). This chapter has provided a practical extension for counterfactual explanations to improve their plausibility post-hoc, despite missing control over the explanation algorithm or the initial classification model. Existing post-processors estimate the plausibility of counterfactual instances with statistical distance metrics wrt. a reference data set (Laugel et al., 2019; Artelt et al., 2021). The reference data set might be unrepresentative for the population, which is why plausibility violations might not be detected by existing methods. The proposed post-processor builds a surrogate in the form of a decision tree, extracts its rules recursively and adds the class probabilities. The former is used to construct a probabilistic logic rule set extended by deterministic constraints. Each counterfactual instance is propagated through the constrained rule set to ensure their validity and to identify plausibility violations. The experiments show that despite minor deviations, the counterfactual explanation quality remains stable after post-processing.

Limitations Discussions on the proposed counterfactual post-processing technique should focus on three dimensions: post-processing in general, the concept of logical constraints to increase the plausibility of counterfactual explanations, and the experimental setup.

- **Post-processing**:
  From Table B.2 can be deduced: Apart from the ratio of returned to queried counterfactual explanations and the computational time, the quantitative behavior of counterfactual instances after post-processing remains stable. Both exceptions are post-processing-inherent problems: When counterfactual instances are deleted, the ratio of returned to queried counterfactual instances will not improve. Neither will the computational time if operations are added. Nevertheless, a more precise inspection of the subsequent differences – the plummeted discriminative power and the increased sparsity similarity for Adult – indicates another problem: A poor discriminative power implies that the counterfactual explanations can be hardly differentiated from the original instance. A high sparsity similarity exacerbates this circumstance. Another general problem is that actually two distinct surrogate models are used in the proposed framework: first, a decision tree and subsequently, a probabilistic logic program. The decision tree is an efficient and interpretable surrogate model, yet redundant and should be replaced with probabilistic rule learning (e.g., Raedt et al., 2015) with probabilistic examples (Definition 4.3).

- **Logical constraints**:
  On a theoretical level, the post-processor depends on the awareness of plausibility constraints and on the ability to express them logically. Both restrictions are necessary, as background knowledge depends on each individual and implicit knowledge cannot be formalized directly. Furthermore, the post-processor puts forward that each individual has sufficient domain knowledge to distinguish plausible from implausible instances. A possible enhancement is the connection of statistical post-processors with the proposed one. Another opportunity is to extract the causal relationships of the data set with structured causal models (e.g., Pearl, 2009) as a baseline for the plausibility constraints.
- **Experimental setup**:
  Three remarks on the experimental setting: First, the Adult data set is truncated to force more heterogeneous counterfactual explanations. This crucial operation has not been evaluated. Second, the support vector classifier does not use balancing methods for simplicity reasons, a poor-performing decision, which should be corrected in future evaluations. In general, however, it can be argued that the predictive power is of subsequent importance with a high fidelity. Indeed, error rate deltas are low when the surrogate approximates the classifier. Nevertheless, more classifiers and data sets should be explored. Ideally, the evaluations are even expanded to multi-class classifications.

Implications for CAIPI Obviously, the implications of this chapter on CAIPI (Teso and Kersting, 2019) are under the premise that CAIPI leverages counterfactual instances as a local explanation component. Then, an improved plausibility of counterfactual explanations might improve the users' understanding of the decision boundary. Consider implausible counterfactual instances such as in the introductory example: If some feature alternations are exaggerated or the indecisive features are altered to an over-proportional extent instead of the decisive features to adjust the model's decision, users might evaluate the explanation as being incorrect, despite the decision boundary might, in fact, be located correctly. Such problems can be prevented with higher quality – in this case, more plausible – counterfactual explanations.

Appendix C

Peer-Reviewed Publications

C.1 Counterexamples from Constrained LLMs

Full reference:

Slany, Emanuel, Stephan Scheele, and Ute Schmid (2024). "Explanatory Interactive Machine Learning with Counterexamples from Constrained Large Language Models". In: KI 2024: Advances in Artificial Intelligence. Ed. by Andreas Hotho and Sebastian Rudolph. Cham: Springer Nature Switzerland, pp. 324–331. DOI: 10.1007/978-3-031-70893-0_26.
Type: conference article with talk

My scientific contribution:

I conceptualized the paper. I was responsible for the formalization and implementation of the method. I conducted and evaluated all experiments. Ute Schmid provided feedback on the conceptualization and Stephan Scheele on the formalization.

My writing contribution:

I wrote the entire paper. Stephan Scheele provided feedback on the structure and helped arranging the materials regarding the page limit.

C.2 FairCAIPI

Full reference:

Heidrich, Louisa, Emanuel **Slany**, Stephan Scheele, and Ute Schmid (2023). "FairCaipi: A Combination of Explanatory Interactive and Fair Machine Learning for Human and Machine Bias Reduction". In: *Machine Learning and Knowledge Extraction* 5.4, pp. 1519–1538. DOI: 10.3390/make5040076.
Type: journal article

My scientific contribution:

I conceptualized and structured the paper. I developed a formal notation for the FAIRCAIPI approach. I implemented parts of the code to reproduce and optimize the results of the experiments, which originate in Louisa Heidrich's MSc thesis. Further, I curate the open-source FAIRCAIPI project linked to this article. I had help with the mathematical notation from Stephan Scheele and Ute Schmid helped during the revision.

My writing contribution:

I wrote the paper entirely, while I recycled some parts, especially in the literature review, from Louisa Heidrich's MSc thesis. Stephan Scheele rewrote minor parts, specifically the description of the SHAP approach as well as the discussion of FAIR-CAIPI's time complexity. Figures 1, 2, and 4 stem from Stephan Scheele.

C.3 CAIPI in Practice

Full reference:

Slany, Emanuel, Yannik Ott, Stephan Scheele, Jan Paulus, and Ute Schmid (2022). "CAIPI in Practice: Towards Explainable Interactive Medical Image Classification". In: *Artificial Intelligence Applications and Innovations. AIAI 2022 IFIP WG 12.5 International Workshops - MHDW 2022, 5G-PINE 2022, AIBMG 2022, ML@HC 2022, and AIBEI 2022, Hersonissos, Crete, Greece, June 17-20, 2022, Proceedings.* Ed. by Ilias Maglogiannis et al. Vol. 652. IFIP Advances in Information and Communication Technology. Springer, pp. 389–400. DOI: 10.1007/978-3-031-08341-9_31.
Type: workshop article with talk

My scientific contribution:

This article originates in Yannik Ott's BSc thesis, which Ute Schmid and I jointly supervised. I conceptualized and structured the paper. I optimized Yannik Ott's technical realization and conducted the experiments together with him.

My writing contribution:

I wrote this paper entirely. Stephan Scheele, Jan Paulus, and Ute Schmid all offered their review comments, which I have integrated.

C.4 Bayesian CAIPI

Full reference:

Slany, Emanuel, Stephan Scheele, and Ute Schmid (2024). "Bayesian CAIPI: A Probabilistic Approach to Explanatory and Interactive Machine Learning". In: *Artificial Intelligence. ECAI 2023 International Workshops*. Ed. by Sławomir Nowaczyk et al. Cham: Springer Nature Switzerland, pp. 285–301. DOI: 10.1007/978-3-031-50396-2_16.
Type: workshop article with talk

My scientific contribution:

From idea, over conceptualization, technical realization, to evaluation, the content of this paper is entirely my own. Except, I had help with the algorithmic formalization of Bayesian CAIPI by Stephan Scheele.

My writing contribution:

This paper is entirely written by my own. Except, Stephan Scheele wrote the start of the Explanatory Interactive Machine Learning section. Moreover, Stephan Scheele guided me through the editing process.

C.5 Cluster XAI

Full reference:

Amling, Jonas, Stephan Scheele, Emanuel **Slany**, Moritz Lang, and Ute Schmid (2024). "Explainable AI for Mixed Data Clustering". In: Explainable Artificial Intelligence. Ed. by Luca Longo, Sebastian Lapuschkin, and Christin Seifert. Cham: Springer Nature Switzerland, pp. 42–62. DOI: 10.1007/978-3-031-63797-1_3.
Type: conference article with talk (held by Jonas Amling)

My scientific contribution:

The work is based on the MSc thesis of Jonas Amling. I assisted him during the conceptualization and provided continuous feedback regarding the statistical methods parts. Furthermore, I was responsible for the formalization of the method.

My writing contribution:

I authored the notation, definitions, explanations, algorithms, and technical parts related to SHAP. Thereby, I modified existing materials from Jonas Amling's MSc thesis. Furthermore, I revised the entire text.

C.6 Hybrid Explanatory Interactive Machine Learning

Full reference:

Slany, Emanuel, Stephan Scheele, and Ute Schmid (2024). "Hybrid Explanatory Interactive Machine Learning for Medical Diagnosis". In: Artificial Intelligence Applications and Innovations. Ed. by Ilias Maglogiannis et al. Cham: Springer Nature Switzerland, pp. 105–116. DOI: 10.1007/978-3-031-63211-2_9.
Type: conference article with talk

My scientific contribution:

The content of this paper is entirely my own. Except, the algorithmic formalization emerged from discussions with Stephan Scheele, who also refined the notation.

My writing contribution:

I wrote the paper entirely, while Stephan Scheele and Ute Schmid supported me during the revision process.

Appendix D

Scientific Activities

D.1 Non-Peer-Reviewed Publications

Slany, Emanuel (2024). "Zusammenarbeit von Mensch und Künstlicher Intelligenz in der Chemieindustrie". *"perspectives" magazine by fazit, an FAZ company*. To be published.
Type: interview

Schmid, Ute, Emanuel **Slany**, and Stephan Scheele (2023). "Understanding the Why and How of Trustworthy AI". *Smart Sensing Insights*. URL: `https://websites.fraunhofer.de/smart-sensing-insights/trustworthy-ai/`.
Type: blog post

D.2 Talks

Slany, Emanuel (2024). "Zusammenarbeit von Mensch und Künstlicher Intelligenz in der Chemieindustrie". *ACHEMA 2024*. Frankfurt (Germany), 13 June 2024.

Slany, Emanuel (2023). "PHAL: Post-Hoc model Approximation with Logic". *Nordic Probabilistic AI Summer School 2023*. Trondheim (Norway), 15 June 2023.

Schmid, Ute, Emanuel **Slany**, and Stephan Scheele (2022). "Hybrid Explanatory Interactive Machine Learning – Towards Human-AI Partnerships". *Airbus Tech Talk*. Virtual, 21 November 2022.

Slany, Emanuel (2022). "CAIPI in Practice: Towards Explainable Interactive Medical Image Classification". *Heinz-Nixdorf Symposium 2022*. Paderborn (Germany), 16 September 2022.

Slany, Emanuel (2022). "Explainable Gaussian Process Regression with Probabilistic Influence and Logic". *Heinz-Nixdorf Symposium 2022*. Paderborn (Germany), 15 September 2022.

D.3 Reviews

"DT-PPO: Interpretable Proximal Policy Optimization using Decision Trees". *IJCAI 2024*.

"Deep Differentiable Symbolic Regression Neural Networks". *AAAI 2023*.

"Assessing the Performance Gain on Retail Article Categorization at the Expense of Explainability and Resource Efficiency". *KI 2022*.

"Agnostic Explanation of Model Change based on Feature Importance". *KI - Künstliche Intelligenz 2022*.

D.4 Advised Thesis

Gernlein, Lukas (2024). "An Explanatory and Interactive Machine Learning Approach for Multi-Label Classification". *MSc thesis*. Supervision by Emanuel **Slany** and Stephan Scheele.

Hempel, Felix (2023). "Explainable and Interactive Machine Learning with Counterfactuals and Ordinal Data". *MSc thesis*. Supervision by Emanuel **Slany** and Stephan Scheele.

Rabshal, Solveig (2023). "Exploring the Impact of Scale of Measurement on Counterfactual Explanations". *BSc thesis*. Supervision by Emanuel **Slany** and Stephan Scheele.

Ott, Yannik (2022). "An explanatory interactive machine learning approach for image classification in medical engineering". *BSc thesis*. Supervision by Emanuel **Slany** and Ute Schmid.

Serdarov, Oraz (2022). "Explainable Unsupervised Learning for Fraud Detection". *MSc thesis with HUK-Coburg*. Supervision by Emanuel **Slany** and Ute Schmid.

D.5 Proposals

Hauenstein, Thomas, Andreas Ernst, Emanuel **Slany**, Ute Schmid, and Dominik Seuß (2024, for Fraunhofer IIS). "Intelligente, multimodale Info-Displays (IMID)". *ZIM, BMWK Germany*.

Slany, Emanuel and Ute Schmid (2024, for Fraunhofer IIS). "Artificial Intelligence assisted Technical Information Model (AITIM)". *Bayern Innovativ, Germany*.

Slany, Emanuel and Ute Schmid (2023, for Fraunhofer IIS). "Extension: Humanzentrierte Künstliche Intelligenz in der Chemischen Industrie (hKI-Chemie)". *BMBF Germany*.

Bibliography

Adadi, Amina and Mohammed Berrada (2018). "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)". In: *IEEE Access* 6, pp. 52138–52160. DOI: 10.1109/ACCESS.2018.2870052.

Ahn, Yongsu et al. (2022). "Tribe or Not? Critical Inspection of Group Differences Using TribalGram". In: *ACM Trans. Interact. Intell. Syst.* 12.1, 5:1–5:34. DOI: 10.1145/3484509.

Ali, Sajid et al. (2023). "Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence". In: *Inf. Fusion* 99, p. 101805. DOI: 10.1016/J.INFFUS.2023.101805.

Amershi, Saleema et al. (2014). "Power to the People: The Role of Humans in Interactive Machine Learning". In: *AI Mag.* 35.4, pp. 105–120. DOI: 10.1609/AIMAG.V35I4.2513.

Amling, Jonas et al. (2024). "Explainable AI for Mixed Data Clustering". In: *Explainable Artificial Intelligence*. Ed. by Luca Longo, Sebastian Lapuschkin, and Christin Seifert. Cham: Springer Nature Switzerland, pp. 42–62. DOI: 10.1007/978-3-031-63797-1_3.

Arenas, Marcelo et al. (2023). "On the Complexity of SHAP-Score-Based Explanations: Tractability via Knowledge Compilation and Non-Approximability Results". In: *Journal of Machine Learning Research* 24.63, pp. 1–58. URL: http://jmlr.org/papers/v24/21-0389.html.

Artelt, André et al. (2021). "Evaluating Robustness of Counterfactual Explanations". In: *IEEE Symposium Series on Computational Intelligence, SSCI 2021, Orlando, FL, USA, December 5-7, 2021*. IEEE, pp. 1–9. DOI: 10.1109/SSCI50451.2021.9660058.

Artelt, André (2019). *CEML - Counterfactuals for Explaining Machine Learning models - A Python toolbox*. DOI: 10.4119/UNIBI/2936468.

Azevedo, Beatriz Flamia, Ana Maria A. C. Rocha, and Ana I. Pereira (2024). "Hybrid approaches to optimization and machine learning methods: a systematic literature review". In: *Mach. Learn.* 113.7, pp. 4055–4097. DOI: 10.1007/S10994-023-06467-X.

Baniecki, Hubert and Przemyslaw Biecek (2019). "modelStudio: Interactive Studio with Explanations for ML Predictive Models". In: *J. Open Source Softw.* 4.43, p. 1798. DOI: 10.21105/JOSS.01798.

— (2020). *The Grammar of Interactive Explanatory Model Analysis*. arXiv: 2005.00497.

Baniecki, Hubert et al. (2021). "dalex: Responsible Machine Learning with Interactive Explainability and Fairness in Python". In: *J. Mach. Learn. Res.* 22, 214:1–214:7. URL: http://jmlr.org/papers/v22/20-1473.html.

Beckh, Katharina et al. (2021). *Explainable Machine Learning with Prior Knowledge: An Overview*. arXiv: 2105.10172.

Bellamy, Rachel K. E. et al. (2018). *AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias*. arXiv: 1810.01943.

Bertrand, Astrid et al. (2023). "On Selective, Mutable and Dialogic XAI: a Review of What Users Say about Different Types of Interactive Explanations". In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, CHI 2023,*

Hamburg, Germany, April 23-28, 2023. Ed. by Albrecht Schmidt et al. ACM, 411:1–411:21. DOI: 10.1145/3544548.3581314.

Blei, David M., Alp Kucukelbir, and Jon D. McAuliffe (2017). "Variational Inference: A Review for Statisticians". In: *Journal of the American Statistical Association* 112.518, pp. 859–877. DOI: 10.1080/01621459.2017.1285773.

Bobek, Szymon et al. (2022). "Enhancing Cluster Analysis With Explainable AI and Multidimensional Cluster Prototypes". In: *IEEE Access* 10, pp. 101556–101574. DOI: 10.1109/ACCESS.2022.3208957.

Bock, Hans-Hermann (2007). "Clustering Methods: A History of k-Means Algorithms". In: *Selected Contributions in Data Analysis and Classification*. Springer Berlin Heidelberg, pp. 161–172. DOI: 10.1007/978-3-540-73560-1_15.

Breiman, Leo (2001). "Random Forests". In: *Machine Learning* 45.1, pp. 5–32. DOI: 10.1023/a:1010933404324.

Brown, Tom B. et al. (2020). "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle et al. URL: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Bunt, Andrea, Matthew Lount, and Catherine Lauzon (2012). "Are explanations always important?: a study of deployed, low-cost intelligent interactive systems". In: *17th International Conference on Intelligent User Interfaces, IUI 2012, Lisbon, Portugal, February 14-17, 2012*. Ed. by Carlos Duarte et al. ACM, pp. 169–178. DOI: 10.1145/2166966.2166996.

Buslaev, Alexander et al. (2020). "Albumentations: Fast and Flexible Image Augmentations". In: *Information* 11.2. DOI: 10.3390/info11020125.

Chakraborty, Shayok (2020). "Asking the Right Questions to the Right Users: Active Learning with Imperfect Oracles". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.04, pp. 3365–3372. DOI: 10.1609/aaai.v34i04.5738.

Chakraborty, Tanmay, Christian Wirth, and Christin Seifert (2024). "Post-hoc Rule Based Explanations for Black Box Bayesian Optimization". In: *Artificial Intelligence. ECAI 2023 International Workshops*. Ed. by Sławomir Nowaczyk et al. Cham: Springer Nature Switzerland, pp. 320–337. DOI: 10.1007/978-3-031-50396-2_18.

Chen, Jiahao et al. (2019). "Fairness Under Unawareness: Assessing Disparity When Protected Class Is Unobserved". In: *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, Atlanta, GA, USA, January 29-31, 2019*. Ed. by danah boyd and Jamie H. Morgenstern. ACM, pp. 339–348. DOI: 10.1145/3287560.3287594.

Chen, Tianqi and Carlos Guestrin (2016). "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. DOI: 10.1145/2939672.2939785.

Chouldechova, Alexandra (2017). "Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments". In: *Big Data* 5.2, pp. 153–163. DOI: 10.1089/big.2016.0047.

Chung, John Joon Young, Ece Kamar, and Saleema Amershi (2023). "Increasing Diversity While Maintaining Accuracy: Text Data Generation with Large Language Models and Human Interventions". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*. Vol. 1, pp. 575–593. URL: https://aclanthology.org/2023.acl-long.34.pdf.

Dietvorst, Berkeley J., Joseph P. Simmons, and Cade Massey (2018). "Overcoming Algorithm Aversion: People Will Use Imperfect Algorithms If They Can (Even

Slightly) Modify Them". In: *Manag. Sci.* 64.3, pp. 1155–1170. DOI: 10.1287/MNSC.2016.2643.

Doersch, Carl (2016). *Tutorial on Variational Autoencoders*. arXiv: 1606.05908.

Dosovitskiy, Alexey et al. (2021). "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. URL: https://openreview.net/forum?id=YicbFdNTTy.

Du, Jun and Charles X. Ling (2010). "Active Learning with Human-Like Noisy Oracle". In: *2010 IEEE International Conference on Data Mining*, pp. 797–802. DOI: 10.1109/ICDM.2010.114.

Dwork, Cynthia et al. (2012). "Fairness through awareness". In: *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*. Ed. by Shafi Goldwasser. ACM, pp. 214–226. DOI: 10.1145/2090236.2090255.

Ester, Martin et al. (1996). "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*. Ed. by Evangelos Simoudis, Jiawei Han, and Usama M. Fayyad. AAAI Press, pp. 226–231. URL: http://www.aaai.org/Library/KDD/1996/kdd96-037.php.

Friedrich, Felix et al. (2023). "A typology for exploring the mitigation of shortcut behaviour". In: *Nat. Mac. Intell.* 5.3, pp. 319–330. DOI: 10.1038/S42256-023-00612-W.

Fürnkranz, Johannes (2011). "Decision Tree". In: *Encyclopedia of Machine Learning*. Springer US, pp. 263–267. DOI: 10.1007/978-0-387-30164-8_204.

Gagolewski, Marek (2022). "A framework for benchmarking clustering algorithms". In: *SoftwareX* 20, p. 101270. DOI: 10.1016/J.SOFTX.2022.101270.

Gelfand, Alan E. (2000). "Gibbs Sampling". In: *Journal of the American Statistical Association* 95.452, pp. 1300–1304. DOI: 10.2307/2669775.

Gilpin, Leilani H. et al. (2018). "Explaining Explanations: An Overview of Interpretability of Machine Learning". In: *5th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2018, Turin, Italy, October 1-3, 2018*. Ed. by Francesco Bonchi et al. IEEE, pp. 80–89. DOI: 10.1109/DSAA.2018.00018.

Göpfert, Jan Philip et al. (2022). "Intuitiveness in Active Teaching". In: *IEEE Trans. Hum. Mach. Syst.* 52.3, pp. 458–467. DOI: 10.1109/THMS.2021.3121666.

Gower, J. C. (1971). "A General Coefficient of Similarity and Some of Its Properties". In: *Biometrics* 27.4, pp. 857–871. DOI: 10.2307/2528823.

Guidotti, Riccardo (2022). "Counterfactual explanations and how to find them: literature review and benchmarking". In: *Data Mining and Knowledge Discovery*. DOI: 10.1007/s10618-022-00831-6.

Guidotti, Riccardo et al. (2018). *Local Rule-Based Explanations of Black Box Decision Systems*. arXiv: 1805.10820.

Hagos, Misgina Tsighe, Kathleen M. Curran, and Brian Mac Namee (2022). "Impact of Feedback Type on Explanatory Interactive Learning". In: *Foundations of Intelligent Systems - 26th International Symposium, ISMIS 2022, Cosenza, Italy, October 3-5, 2022, Proceedings*. Ed. by Michelangelo Ceci et al. Vol. 13515. Lecture Notes in Computer Science. Springer, pp. 127–137. DOI: 10.1007/978-3-031-16564-1_13.

Hammond, Kristian J. and David B. Leake (2023). "Large Language Models Need Symbolic AI". In: *Proceedings of the 17th International Workshop on Neural-Symbolic Learning and Reasoning, La Certosa di Pontignano, Siena, Italy, July 3-5, 2023*. Ed. by

Artur S. d'Avila Garcez et al. Vol. 3432. CEUR Workshop Proceedings. CEUR-WS.org, pp. 204–209. URL: https://ceur-ws.org/Vol-3432/paper17.pdf.

Hardt, Moritz, Eric Price, and Nati Srebro (2016). "Equality of Opportunity in Supervised Learning". In: *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. Ed. by Daniel D. Lee et al., pp. 3315–3323. URL: https://proceedings.neurips.cc/paper/2016/hash/9d2682367c3935defcb1f9e247a97c0d-Abstract.html.

Heidrich, Louisa et al. (2023). "FairCaipi: A Combination of Explanatory Interactive and Fair Machine Learning for Human and Machine Bias Reduction". In: *Machine Learning and Knowledge Extraction* 5.4, pp. 1519–1538. DOI: 10.3390/make5040076.

Hoffman, Robert R. et al. (2023). "Evaluating machine-generated explanations: a "Scorecard" method for XAI measurement science". In: *Frontiers Comput. Sci.* 5. DOI: 10.3389/FCOMP.2023.1114806.

Holzinger, Andreas (2016). "Interactive machine learning for health informatics: when do we need the human-in-the-loop?" In: *Brain Informatics* 3.2, pp. 119–131. DOI: 10.1007/S40708-016-0042-6.

Huang, Lei et al. (2023). "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions". In: DOI: 10.48550/ARXIV.2311.05232.

Kamiran, Faisal and Toon Calders (2011). "Data preprocessing techniques for classification without discrimination". In: *Knowl. Inf. Syst.* 33.1, pp. 1–33. DOI: 10.1007/s10115-011-0463-8.

Kanamori, Kentaro et al. (2020). "DACE: Distribution-Aware Counterfactual Explanation by Mixed-Integer Linear Optimization". In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*. Ed. by Christian Bessiere. ijcai.org, pp. 2855–2862. DOI: 10.24963/IJCAI.2020/395.

Kearns, Michael J. and Leslie G. Valiant (1994). "Cryptographic Limitations on Learning Boolean Formulae and Finite Automata". In: *J. ACM* 41.1, pp. 67–95. DOI: 10.1145/174644.174647.

Kiefer, Sebastian (2022). "CaSE: Explaining Text Classifications by Fusion of Local Surrogate Explanation Models with Contextual and Semantic Knowledge". In: *Information Fusion* 77, pp. 184–195. DOI: 10.1016/j.inffus.2021.07.014.

— (2023). *Human-centered Interactions with Text Classifiers: Fusing Concept-based Knowledge with Local Surrogate Explanation Models*. URL: https://fis.uni-bamberg.de/entities/publication/3ecdb0a-0262-469d-b42f-d3efb4e2d726.

Kiefer, Sebastian, Mareike Hoffmann, and Ute Schmid (2022). "Semantic Interactive Learning for Text Classification: A Constructive Approach for Contextual Interactions". In: *Mach. Learn. Knowl. Extr.* 4.4, pp. 994–1010. DOI: 10.3390/MAKE4040050.

Kimmig, Angelika et al. (2011). "On the implementation of the probabilistic logic programming language ProbLog". In: *Theory Pract. Log. Program.* 11.2-3, pp. 235–262. DOI: 10.1017/S1471068410000566.

Kingma, Diederik P. and Jimmy Ba (2015). "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. arXiv: 1412.6980.

Kingma, Diederik P. and Max Welling (2014). "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB,*

Canada, April 14-16, 2014, Conference Track Proceedings. Ed. by Yoshua Bengio and Yann LeCun. arXiv: 1312.6114.

Kourou, Konstantina et al. (2015). "Machine learning applications in cancer prognosis and prediction". In: *Computational and Structural Biotechnology Journal* 13, pp. 8–17. DOI: 10.1016/j.csbj.2014.11.005.

Kulesza, Todd et al. (2015). "Principles of Explanatory Debugging to Personalize Interactive Machine Learning". In: *Proceedings of the 20th International Conference on Intelligent User Interfaces, IUI 2015, Atlanta, GA, USA, March 29 - April 01, 2015*. Ed. by Oliver Brdiczka et al. ACM, pp. 126–137. DOI: 10.1145/2678025.2701399.

Lapuschkin, Sebastian et al. (2019). "Unmasking Clever Hans predictors and assessing what machines really learn". In: *Nature Communications* 10.1, p. 1096. DOI: 10.1038/s41467-019-08987-4.

Laugel, Thibault et al. (2019). "The Dangers of Post-hoc Interpretability: Unjustified Counterfactual Explanations". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*. Ed. by Sarit Kraus. ijcai.org, pp. 2801–2807. DOI: 10.24963/IJCAI.2019/388.

Li, Peng, Elizabeth A. Stuart, and David B. Allison (2015). "Multiple Imputation: A Flexible Tool for Handling Missing Data". In: *JAMA* 314.18, pp. 1966–1967. DOI: 10.1001/jama.2015.15281.

Liu, Han, Vivian Lai, and Chenhao Tan (2021). "Understanding the Effect of Out-of-distribution Examples and Interactive Explanations on Human-AI Decision Making". In: *Proc. ACM Hum. Comput. Interact.* 5.CSCW2, 408:1–408:45. DOI: 10.1145/3479552.

Lloyd, S. (1982). "Least squares quantization in PCM". In: *IEEE Transactions on Information Theory* 28.2, pp. 129–137. DOI: 10.1109/tit.1982.1056489.

Lundberg, Scott M. and Su-In Lee (2017). "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al., pp. 4765–4774. URL: https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

Maaten, Laurens Van der and Geoffrey Hinton (2008). "Visualizing data using t-SNE". In: *Journal of Machine Learning Research* 9.11, pp. 2579–2605. URL: https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf.

MacKay, David J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press. ISBN: 978-0-521-64298-9.

Madhulatha, T. Soni (2012). "An overview on clustering methods". In: *IOSR Journal of Engineering* 02.04, 719–725. DOI: 10.9790/3021-0204719725.

Manhaeve, Robin et al. (2018). "DeepProbLog: Neural Probabilistic Logic Programming". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. Ed. by Samy Bengio et al., pp. 3753–3763. URL: https://proceedings.neurips.cc/paper/2018/hash/dc5d637ed5e62c36ecb73b654b05ba2a-Abstract.html.

Marsh, Charles (2013). "Introduction to Continuous Entropy". In: *Department of Computer Science, Princeton University* 1034. URL: https://crmarsh.com/pdf/Charles_Marsh_Continuous_Entropy.pdf.

Mehrabi, Ninareh et al. (2022). "A Survey on Bias and Fairness in Machine Learning". In: *ACM Comput. Surv.* 54.6, 115:1–115:35. DOI: 10.1145/3457607.

Morichetta, Andrea, Pedro Casas, and Marco Mellia (2019). "EXPLAIN-IT: Towards Explainable AI for Unsupervised Network Traffic Analysis". In: *Proceedings of the 3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks, Big-DAMA@CoNEXT 2019, Orlando, FL, USA, December 9, 2019*. ACM, pp. 22–28. DOI: 10.1145/3359992.3366639.

Mosqueira-Rey, Eduardo et al. (2023). "Human-in-the-loop machine learning: a state of the art". In: *Artif. Intell. Rev.* 56.4, pp. 3005–3054. DOI: 10.1007/S10462-022-10246-W.

Mothilal, Ramaravind K, Amit Sharma, and Chenhao Tan (2020). "Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations". In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 607–617. URL: https://par.nsf.gov/servlets/purl/10179945.

Müller, Dennis et al. (2022). "An Interactive Explanatory AI System for Industrial Quality Control". In: *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*. AAAI Press, pp. 12580–12586. DOI: 10.1609/AAAI.V36I11.21530.

Nakao, Yuri et al. (2022). "Toward Involving End-users in Interactive Human-in-the-loop AI Fairness". In: *ACM Trans. Interact. Intell. Syst.* 12.3, 18:1–18:30. DOI: 10.1145/3514258.

Nye, Maxwell I. et al. (2021). "Improving Coherence and Consistency in Neural Sequence Models with Dual-System, Neuro-Symbolic Reasoning". In: *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. Ed. by Marc'Aurelio Ranzato et al., pp. 25192–25204. URL: https://proceedings.neurips.cc/paper/2021/hash/d3e2e8f631bd9336ed25b8162aef8782-Abstract.html.

Pearl, Judea (2009). *Causality: Models, Reasoning and Inference*. 2nd. USA: Cambridge University Press. ISBN: 052189560X.

Pfeuffer, Nicolas et al. (2023). "Explanatory Interactive Machine Learning". In: *Business & Information Systems Engineering* 65.6, pp. 677–701. DOI: 10.1007/s12599-023-00806-x.

Qayyum, Adnan et al. (2021). "Secure and Robust Machine Learning for Healthcare: A Survey". In: *IEEE Reviews in Biomedical Engineering* 14, pp. 156–180. DOI: 10.1109/RBME.2020.3013489.

Quinlan, J. Ross (1990). "Learning Logical Definitions from Relations". In: *Mach. Learn.* 5, pp. 239–266. DOI: 10.1007/BF00117105.

Radford, Alec et al. (2019). "Language Models are Unsupervised Multitask Learners". In: URL: https://api.semanticscholar.org/CorpusID:160025533.

Raedt, Luc De, Angelika Kimmig, and Hannu Toivonen (2007). "ProbLog: A Probabilistic Prolog and Its Application in Link Discovery". In: *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pp. 2462–2467. URL: http://ijcai.org/Proceedings/07/Papers/396.pdf.

Raedt, Luc De et al. (2015). "Inducing Probabilistic Relational Rules from Probabilistic Examples". In: *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*. AAAI Press, pp. 1835–1843. URL: http://ijcai.org/Abstract/15/261.

Rasmussen, Carl Edward and Christopher K. I. Williams (2006). "Gaussian Processes for Machine Learning". In: Adaptive computation and machine learning. MIT Press. URL: https://gaussianprocess.org/gpml/.

Rezatofighi, Hamid et al. (2019). "Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, pp. 658–666. DOI: 10.1109/CVPR.2019.00075.

Ribeiro, Marco Túlio and Scott M. Lundberg (2022). "Adaptive Testing and Debugging of NLP Models". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Association for Computational Linguistics, pp. 3253–3267. DOI: 10.18653/V1/2022.ACL-LONG.230.

Ribeiro, Marco Túlio, Sameer Singh, and Carlos Guestrin (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*. Ed. by Balaji Krishnapuram et al. ACM, pp. 1135–1144. DOI: 10.1145/2939672.2939778.

— (2018). "Anchors: High-Precision Model-Agnostic Explanations". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. Ed. by Sheila A. McIlraith and Kilian Q. Weinberger. AAAI Press, pp. 1527–1535. DOI: 10.1609/AAAI.V32I1.11491.

Rosenfeld, Avi (2021). "Better Metrics for Evaluating Explainable Artificial Intelligence". In: *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021*. Ed. by Frank Dignum et al. ACM, pp. 45–50. DOI: 10.5555/3463952.3463962.

Ross, Andrew Slavin, Michael C. Hughes, and Finale Doshi-Velez (2017). "Right for the Right Reasons: Training Differentiable Models by Constraining their Explanations". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*. Ed. by Carles Sierra. ijcai.org, pp. 2662–2670. DOI: 10.24963/IJCAI.2017/371.

Rüden, Laura von et al. (2023). "Informed Machine Learning - A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems". In: *IEEE Trans. Knowl. Data Eng.* 35.1, pp. 614–633. DOI: 10.1109/TKDE.2021.3079836.

Rudin, Cynthia (2019). "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead". In: *Nat. Mach. Intell.* 1.5, pp. 206–215. DOI: 10.1038/S42256-019-0048-X.

Sabbatini, Federico, Giovanni Ciatto, and Andrea Omicini (2021). "GridEx: An Algorithm for Knowledge Extraction from Black-Box Regressors". In: *Explainable and Transparent AI and Multi-Agent Systems - Third International Workshop, EXTRAAMAS 2021, Virtual Event, May 3-7, 2021, Revised Selected Papers*. Ed. by Davide Calvaresi et al. Vol. 12688. Lecture Notes in Computer Science. Springer, pp. 18–38. DOI: 10.1007/978-3-030-82017-6_2.

Schallner, Ludwig et al. (2019). "Effect of Superpixel Aggregation on Explanations in LIME - A Case Study with Biological Data". In: *Machine Learning and Knowledge Discovery in Databases - International Workshops of ECML PKDD 2019, Würzburg, Germany, September 16-20, 2019, Proceedings, Part I*. Ed. by Peggy Cellier and Kurt Driessens. Vol. 1167. Communications in Computer and Information Science. Springer, pp. 147–158. DOI: 10.1007/978-3-030-43823-4_13.

Schapire, Robert E. (1990). "The Strength of Weak Learnability". In: *Mach. Learn.* 5, pp. 197–227. DOI: 10.1007/BF00116037.

Schmid, Ute and Britta Wrede (2022). "What is Missing in XAI So Far?" In: *KI - Künstliche Intelligenz* 36.3, pp. 303–315. DOI: 10.1007/s13218-022-00786-2.

Scholbeck, Christian A., Henri Funk, and Giuseppe Casalicchio (2023). "Algorithm-Agnostic Feature Attributions for Clustering". In: *Explainable Artificial Intelligence - First World Conference, xAI 2023, Lisbon, Portugal, July 26-28, 2023, Proceedings, Part I*. Ed. by Luca Longo. Vol. 1901. Communications in Computer and Information Science. Springer, pp. 217–240. DOI: 10.1007/978-3-031-44064-9_13.

Schramowski, Patrick et al. (2020). "Making deep neural networks right for the right scientific reasons by interacting with their explanations". In: *Nat. Mach. Intell.* 2.8, pp. 476–486. DOI: 10.1038/s42256-020-0212-3.

Schwalbe, Gesina and Bettina Finzel (2023). "A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts". In: *Data Mining and Knowledge Discovery*. DOI: 10.1007/s10618-022-00867-8.

Settles, Burr (2012). *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers. DOI: 10.2200/S00429ED1V01Y201207AIM018.

Shivaswamy, Pannaga and Thorsten Joachims (2015). "Coactive Learning". In: *J. Artif. Intell. Res.* 53, pp. 1–40. DOI: 10.1613/jair.4539.

Slany, Emanuel, Stephan Scheele, and Ute Schmid (2024a). "Bayesian CAIPI: A Probabilistic Approach to Explanatory and Interactive Machine Learning". In: *Artificial Intelligence. ECAI 2023 International Workshops*. Ed. by Sławomir Nowaczyk et al. Cham: Springer Nature Switzerland, pp. 285–301. DOI: 10.1007/978-3-031-50396-2_16.

— (2024b). "Explanatory Interactive Machine Learning with Counterexamples from Constrained Large Language Models". In: *KI 2024: Advances in Artificial Intelligence*. Ed. by Andreas Hotho and Sebastian Rudolph. Cham: Springer Nature Switzerland, pp. 324–331. DOI: 10.1007/978-3-031-70893-0_26.

— (2024c). "Hybrid Explanatory Interactive Machine Learning for Medical Diagnosis". In: *Artificial Intelligence Applications and Innovations*. Ed. by Ilias Maglogiannis et al. Cham: Springer Nature Switzerland, pp. 105–116. DOI: 10.1007/978-3-031-63211-2_9.

Slany, Emanuel et al. (2022). "CAIPI in Practice: Towards Explainable Interactive Medical Image Classification". In: *Artificial Intelligence Applications and Innovations. AIAI 2022 IFIP WG 12.5 International Workshops - MHDW 2022, 5G-PINE 2022, AIBMG 2022, ML@HC 2022, and AIBEI 2022, Hersonissos, Crete, Greece, June 17-20, 2022, Proceedings*. Ed. by Ilias Maglogiannis et al. Vol. 652. IFIP Advances in Information and Communication Technology. Springer, pp. 389–400. DOI: 10.1007/978-3-031-08341-9_31.

Smith, Alison et al. (2018). "Closing the Loop: User-Centered Design and Evaluation of a Human-in-the-Loop Topic Modeling System". In: *Proceedings of the 23rd International Conference on Intelligent User Interfaces, IUI 2018, Tokyo, Japan, March 07-11, 2018*. Ed. by Shlomo Berkovsky et al. ACM, pp. 293–304. DOI: 10.1145/3172944.3172965.

Smith, Peter (2020). *An Introduction to Formal Logic*. New York: Logic Matters. ISBN: 979-8-675-80394-1.

Snijders, Chris et al. (2023). "Humans and Algorithms Detecting Fake News: Effects of Individual and Contextual Confidence on Trust in Algorithmic Advice". In:

Int. J. Hum. Comput. Interact. 39.7, pp. 1483–1494. DOI: 10.1080/10447318.2022.2097601.

Spinner, Thilo et al. (2020). "explAIner: A Visual Analytics Framework for Interactive and Explainable Machine Learning". In: *IEEE Trans. Vis. Comput. Graph.* 26.1, pp. 1064–1074. DOI: 10.1109/TVCG.2019.2934629.

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan (2017). "Axiomatic Attribution for Deep Networks". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 3319–3328. URL: http://proceedings.mlr.press/v70/sundararajan17a.html.

Teso, Stefano (2019). "Toward Faithful Explanatory Active Learning with Self-explainable Neural Nets". In: URL: https://ceur-ws.org/Vol-2444/ialatecml_paper1.pdf.

Teso, Stefano and Kristian Kersting (2019). "Explanatory Interactive Machine Learning". In: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2019, Honolulu, HI, USA, January 27-28, 2019*. Ed. by Vincent Conitzer, Gillian K. Hadfield, and Shannon Vallor. ACM, pp. 239–245. DOI: 10.1145/3306618.3314293.

Teso, Stefano et al. (2023). "Leveraging explanations in interactive machine learning: An overview". In: *Frontiers in Artificial Intelligence* 6. DOI: 10.3389/frai.2023.1066049.

Thaler, Anna Magdalena and Ute Schmid (2021). "Explaining Machine Learned Relational Concepts in Visual Domains – Effects of Perceived Accuracy on Joint Performance and Trust". In: *Proceedings of the Annual Meeting of the Cognitive Science Society* 43. URL: https://escholarship.org/uc/item/8wr7s491.

van der Waa, Jasper et al. (2021). "Evaluating XAI: A comparison of rule-based and example-based explanations". In: *Artificial Intelligence* 291, p. 103404. DOI: 10.1016/j.artint.2020.103404.

Vaswani, Ashish et al. (2017). "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al., pp. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Vincent, Pascal et al. (2008). "Extracting and composing robust features with denoising autoencoders". In: *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*. Ed. by William W. Cohen, Andrew McCallum, and Sam T. Roweis. Vol. 307. ACM International Conference Proceeding Series. ACM, pp. 1096–1103. DOI: 10.1145/1390156.1390294.

Wachter, Sandra, Brent D. Mittelstadt, and Chris Russell (2017). "Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR". In: arXiv: 1711.00399.

Welc, Jacek and Pedro J. Rodriguez Esquerdo (2018). "Applied Regression Analysis for Business: Tools, Traps and Applications". In: Cham: Springer International Publishing, pp. 1–6. DOI: 10.1007/978-3-319-71156-0_1.

Wexler, James et al. (2020). "The What-If Tool: Interactive Probing of Machine Learning Models". In: *IEEE Trans. Vis. Comput. Graph.* 26.1, pp. 56–65. DOI: 10.1109/TVCG.2019.2934619.

White, Adam and Artur S. d'Avila Garcez (2020). "Measurable Counterfactual Local Explanations for Any Classifier". In: *ECAI 2020 - 24th European Conference on*

Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020). Vol. 325. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 2529–2535. DOI: 10.3233/FAIA200387.

Wirth, Christian, Ute Schmid, and Stefan Voget (2022). "Humanzentrierte Künstliche Intelligenz: Erklärendes interaktives maschinelles Lernen für Effizienzsteigerung von Parametrieraufgaben". In: *Digitalisierung souverän gestalten II*. Ed. by Ernst A. Hartmann. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 80–92. DOI: 10.1007/978-3-662-64408-9_7.

Yang, Zhun, Adam Ishay, and Joohyung Lee (2023). "Coupling Large Language Models with Logic Programming for Robust and General Reasoning from Text". In: *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 5186–5219. URL: https://aclanthology.org/2023.findings-acl.321.pdf.

Young, H. P. (1985). "Monotonic solutions of cooperative games". In: *International Journal of Game Theory* 14.2, pp. 65–72. DOI: 10.1007/BF01769885.