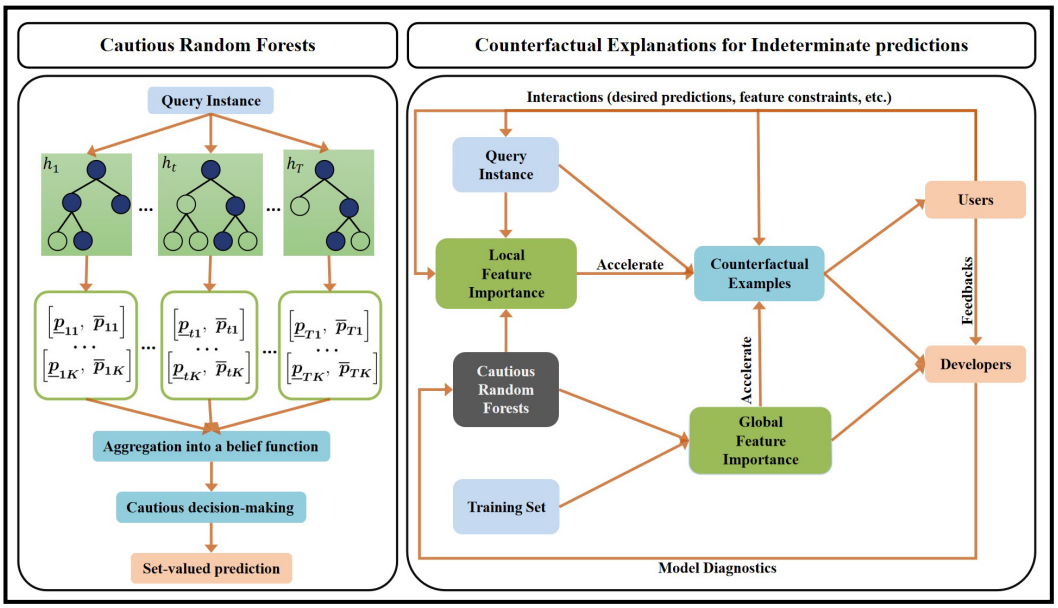
Explainable cautious classifiers

Haifei Zhang

Par **Haifei ZHANG**

Explainable cautious classifiers

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 30 novembre 2023
Spécialité : Informatique : Unité de recherche Heudyasic (UMR-7253)
D2777

Université de Technologie de Compiègne

UMR CNRS 7253, Heudiasyc Laboratory

Doctoral Thesis:

Submitted in fulfillment of the requirements for the degree of Doctor of Computer Science

---

Explainable cautious classifiers

Haifei ZHANG

Spécialité : Informatique

Defended on November 30, 2023

Jury:

Marie-Jeanne Lesot	Professor Sorbonne Université	Reviewer
Cassio De Campos	Professor Eindhoven University of Technology	Reviewer
Yves Grandvalet	CNRS Director of research Université de Technologie de Compiègne	Examiner
David Mercier	Professor Université d'Artois	Examiner
Benjamin Quost	Associate professor Université de Technologie de Compiègne	Supervisor
Marie-Hélène Masson	Associate professor Université de Picardie Jules Verne	Supervisor

Acknowledgements

I would like to give my sincerest acknowledgments to my two deer supervisors: Benjamin Quost and Marie-Hélène Masson. Three years ago, when the COVID-19 epidemic was striking, they accepted me to be their Ph.D. student, helped me apply for funding, and advanced the administrative procedures, which allowed me to start my research smoothly. During these three years, they guided me step by step to complete my dissertation with their endless work passion, solid academic knowledge, and well-defined time planning. When I encountered obstacles in my research, they always gave me the most insightful advice. It is their full trust, ample patience, selfless dedication, and warm encouragement that allowed me to conduct my research in a comfortable rhythm. They are the researchers I respect the most both academically and in terms of personality. Beyond work, their concern for my life was also heartwarming, especially during the epidemic. All in all, without them, the accomplishment of my dissertation would not have been possible at all.

I would also like to thank all the jury members: Prof. Marie-Jeanne Lesot (Sorbonne Université), Prof. Cassio De Campos (Eindhoven University of Technology), Prof. David Mercier (Université d'Artois) and Dr. Yves Grandvalet (Université de technologie de Compiègne). Thank you for spending your valuable time in your busy schedules to review and evaluate my manuscript and give insightful comments. It is your commitment that provides a solid guarantee for the quality of my thesis.

I would like to give special gratitude to Sébastien Destercke and Yves Grandvalet who, as members of my individual monitoring committee, have followed my research progress every year, providing me with research directions and suggestions with their extensive insights. I would also like to thank Vu Linh Nguyen for allowing me to participate in his research and broaden my collaborative relationships.

During the three years passed in the Heudiasyc lab, my colleagues have helped me a lot. Thank you, Bérengère Guermonprez, Nathalie Alexandre, and Véronique Moisan for your professional experience in administration, finance, and electronic

equipment. Their efforts have allowed me to focus on my research without worries about those affairs. Thanks to all the other colleagues in the lab. Forgive me for not being able to list all the names here but I will always remember you and the unforgettable time that I shared with you.

I would like to express my greatest appreciation to my family, especially my beloved parents, brother, and sister-in-law. It was you who strongly encouraged me to continue to pursue a higher degree and my dreams. During these years, being far away from home, it was also your continuous concern for me that made me feel less lonely and motivated me to move forward and keep persevering.

Finally, I would like to thank myself, for making the choice to pursue a doctoral degree, for persevering for three years, and for not wasting my golden ages.

Abstract

Machine learning classifiers have achieved impressive success in a wide range of domains such as natural language processing, image recognition, medical diagnosis, and financial risk assessment. Despite their remarkable accomplishments, their application to real-world problems still entails challenges.

Traditional classifiers make precise decisions based on estimated posterior probabilities; this becomes problematic when dealing with limited data and in complex, uncertain scenarios where making erroneous decisions is costly. As alternatives, cautious classifiers, also known as imprecise classifiers, provide subsets of classes as predictions. We propose in this thesis a cautious classifier called cautious random forest, within the framework of belief functions. It combines imprecise decision trees constructed by the imprecise Dirichlet model and aims at achieving a better compromise between the accuracy and the cautiousness of predictions. Cautious random forests can be regarded as generalizations of classical random forests, where the usual aggregation strategies (averaging and voting) are replaced with a cautious counterpart.

However, making imprecise predictions has a cost, since indeterminacy must be resolved via further analysis. Therefore, it seems crucial to understand what led to an indeterminate prediction, and what could be done to turn it into a determinate one. To address this problem, we propose in this thesis a framework for providing explanations so as to discover which features contribute the most to improving the determinacy of the cautious classifier and how we can modify the feature values so as to achieve a determinate prediction (counterfactual explanations).

Keywords: cautious classification, imprecise Dirichlet model, belief functions, ensemble learning, explainable AI, counterfactual explanation

Résumé

L'apprentissage automatique a connu un succès impressionnant dans des domaines variés comme le traitement du langage naturel, la reconnaissance d'images, ou le diagnostic médical. Malgré ces résultats remarquables, son application à certains problèmes réels soulève encore des questions.

Les classifieurs traditionnels choisissent une classe unique parmi un ensemble de classes possibles (prédiction déterminée), en se basant sur une estimation ponctuelle des probabilités des classes. Cette stratégie peut être problématique lorsque les données sont limitées et dans des scénarios complexes dans lesquels les décisions erronées sont coûteuses. Comme alternative, les classifieurs prudents (classifieurs imprécis) fournissent des sous-ensembles de classes comme prédictions. Nous avons proposé dans cette thèse un classifieur prudent appelé forêt aléatoire prudent développé dans le cadre des fonctions de croyance. Il combine des arbres de décision imprécis construits grâce au modèle de Dirichlet imprécis et vise à atteindre un meilleur compromis entre la précision et la prudence des prédictions. Les forêts aléatoires prudentes peuvent être considérées comme des généralisations des forêts aléatoires classiques, où les stratégies d'agrégation habituelles (calcul de la moyenne et vote) sont remplacées par leurs équivalentes prudentes.

Cependant, faire des prédictions indéterminées a un coût puisque l'indétermination doit être résolue par une analyse plus approfondie. Il semble donc essentiel de comprendre ce qui a conduit à une prédiction indéterminée et ce qui pourrait être fait pour la transformer en une prédiction déterminée. Pour résoudre ce problème, nous avons proposé dans cette thèse un cadre permettant de comprendre d'où provient l'imprécision dans les sorties de notre modèle. En particulier, nous avons proposé l'utilisation d'explications contrefactuelles pour les classifieurs prudents à déterminer comment modifier les entrées du classifieurs pour obtenir une sortie déterminée.

Mots-clés : classification prudente, modèle de Dirichlet imprécis, fonctions de croyance, apprentissage ensembliste, forêts aléatoires, IA explicable, explications contrefactuelles

Contents

List of Figures

List of Tables

Acronyms and notations

Acronyms

OWA	Ordered Weighted Average
IDM	Imprecise Dirichlet Model
DST	Dempster-Shafer Theory
BPA	Basic Probability Assignment
BBA	Basic Belief Assignment
TP	True Positive
FP	False Positive
FN	False Negative
TN	True Negative
ROC	Receiver Operating Characteristic curve
AUC	Area UUnder the ROC Curve
ANOVA	Analysis of Variance
KNN	K-Nearest Neighbors
SVM	Support Vector Machines
LDA	Linear Discriminant Analysis
QDA	Quadratic Discriminant Analysis
DU	Discounted Utility
NDC	NonDeterministic classifier
NCM	NonConformity Measure
CNN	Convolutional Neural Network
NCC	Naive Credal Classifier

CDT	Credal Decision Tree
ICDT	Imprecise Credal Decision Tree
MVA	Minimum Vote Against
CRF	Cautious Random Forest
AVE	AVErage model
MV	Majority Voting
RO	Reject Option
SSA	Single-Set Accuracy
EW	Equal Weight
OOBACC	Out-Of-Bag ACCuracy
OOBU65	Out-Of-Bag u_{65} score
IRF	Imprecise Random Forest
AW	Automatically-learned Weight
CDM_Ave	Cautious Decision-Making via Averaging
CDM_Vote	Cautious Decision-Making via Voting
XAI	eXplainable Artificial Intelligence
COMPAS	Correctional Offender Management Profiling for Alternative Sanctions
GDPR	General Data Protection Regulation
PFI	Permutation Feature Importance
LIME	Local Interpretable Model-agnostic Explanations
SHAP	SHapley Additive exPlanations
PDP	Partial Dependence Plot
ALE	Accumulated Local Effects
ICE	Individual Conditional Expectation
LEAFAGE	Local Example and Feature importance-based model AGnostic Explanations
CLEAR	Counterfactual Local Explanations via Regression
LORE	Local Rule-Based Explanation
FACE	Feasible and Actionable Counterfactual Explanations

DisCERN	Discovering Counterfactual Explanations using Relevance Features from Neighbourhoods
MACE	Model-Agnostic Counterfactual Explanation
DACE	Distribution-Aware Counterfactual Explanation
MDI	Mean Decrease Impurity
SHAP-FI	SHAP Feature Importance
MAD	Median Absolute Deviation
STD	STandard Deviation
MO	Minimum Observable
OFCC	One-Feature-Changed Counterfactual
LOR	Local Outlier Factor

Notations

$\mathcal{A}$	set of possible actions
Ω	set of states of nature, set of possible classes
c_k	k-th state of nature, k-th class
K	number of states of nature, number of classes
u	utility function
$\mathbf{U}$	utility matrix
$\boldsymbol{w}$	weight assignment vector
$\boldsymbol{p}$	probability vector
p	probability of an event
$\mathcal{P}$	credal set
$\mathcal{I}$	set of probability intervals
$\underline{p}$	lower probability
$\overline{p}$	upper probability
s	parameter of the imprecise Dirichlet model
$\underline{\mathbb{E}}_{\mathcal{P}}$	lower expectation according to credal set $\mathcal{P}$
$\overline{\mathbb{E}}_{\mathcal{P}}$	upper expectation according to credal set $\mathcal{P}$

$\mathcal{F}^*_{e\text{-}ad}$	set of maximal actions using the E-admissibility criterion
$\mathcal{F}^*_{max}$	set of maximal actions using the maximality criterion
$\mathcal{F}^*_{id}$	set of maximal actions using the interval dominance criterion
m	mass function
Bel	belief function
Pl	plausibility function
$BetP$	pignistic probability
$\mathcal{K}$	degree of conflict between two mass functions
$\mathbb{E}_{BetP}$	pignistic expectation
$\underline{\mathbb{E}}_m$	lower expectation according to mass function m
$\overline{\mathbb{E}}_m$	upper expectation according to mass function m
$\mathbb{E}_{m,\alpha}$	Hurwicz expectation with a pessimism index α
$\overline{R}_m$	expected maximal regret associated with mass function m
M	number of features
$\mathbf{X} = \{\mathrm{X}^1, \ldots, \mathrm{X}^M\}$	input random vector
$\boldsymbol{\mathcal{X}} = \{\mathcal{X}^1, \ldots, \mathcal{X}^M\}$	input space
$\boldsymbol{x}$	data point in $\boldsymbol{\mathcal{X}}$
$\mathcal{Y}$	output space
y	class label of $\boldsymbol{x}$
$\mathcal{D}_{train}$	set of training data
$\mathcal{D}_{test}$	set of testing data
$\boldsymbol{h}$	classifier
$\mathbf{H}$	ensemble of classifiers
$\mathscr{P}(\Omega)$	power set of Ω
$\hat{y}$	prediction of traditional classifiers
$\hat{Y}$	prediction of cautious classifiers, i.e., set-valued prediction
dr	discount ratio
u_{65}	u_{65} score
u_{80}	u_{80} score
F_β	F_β score
nc	nonconformity

$\underline{\boldsymbol{\delta}}(\boldsymbol{x})$	indicators of whether the lower probability of c_1 evaluated by each tree is greater than or equal to 0.5
$\overline{\boldsymbol{\delta}}(\boldsymbol{x})$	indicators of whether the upper probability of c_1 evaluated by each tree is greater than 0.5
$\mathcal{L}$	loss function
$\mathcal{L}_{sup}$	upper bound of the loss function
λ	regularization coefficient
H	Heaviside function
γ	utility of correct indeterminate predictions of cardinality 2
$A^\star$	subset of Ω that maximizes the lower expected utility
$\overline{M}$	upper bound of prediction cardinality
$f_{\mathbf{X}}$	probability distribution of $\mathbf{X}$
ϕ_j	local feature importance of feature X^j
Φ_j	global feature importance of feature X^j
Imp	imprecision measure for indeterminate predictions
$\boldsymbol{\xi}$	optimal surrogate model
$f_{\boldsymbol{h}}$	payoff function associated with classifier $\boldsymbol{h}$
L_1^{mad}	the Manhattan distance weighted with the inverse median absolute deviation
L_2^{std}	the Euclidean distance weighted with the inverse standard deviation
$\boldsymbol{L}$	set of leaves
SV	split value
SI	split interval
fd	distance along a given feature
d_{sup}	upper bound distance
$\boldsymbol{R}^M$	region in the form of M-dimensional box
cd	cumulative distance along a given set of features

Introduction

Background

Classifiers have demonstrated impressive success in a wide range of domains [130]. From natural language processing to image recognition, from medical diagnosis to financial risk assessment, classifiers are not only an important part of efficient automated decision-making processes, but also achieve accuracy beyond human capabilities in many tasks. However, the challenges they face when dealing with real-world problems still cannot be ignored.

Uncertainty modeling and cautiousness

A major challenge stems from the complexity and uncertainty inherent to real-world problems. Traditionally, classifiers make precise decisions, in the form of a single class, according to the posterior probabilities of classes estimated based on the available information. However, enforcing the assignment of a given instance to a single class is questionable when the available information from which the decision is made is scarce (insufficient evidence), because in this case, the estimated posterior probabilities may not be reliable. As well, in ensemble learning, a large conflict between the outputs of individual learners should lead to avoiding reaching a definitive conclusion.

Considering this issue, in some critical systems where wrong decisions may have serious consequences such as in medical diagnosis, an alternative is to produce imprecise predictions such as sets of plausible classes (or intervals in regression), to

reduce the risk of making erroneous predictions. This specific approach is referred to as cautious classification or imprecise classification [164]. Cautious classifiers can be based on precise probabilities, such as classification with a reject option [36, 81], conformal prediction [208] or the so-called nondeterministic classifier [48]. In addition, imprecise probabilities, which can better model the uncertainty in the available information and provide different imprecise decision-making criteria, can also be applied to constructing cautious classifiers [14]. Some of the most prominent cautious classification representatives are the naive credal classifier [226], imprecise Gaussian discriminant analysis [8], evidential K-nearest neighbors [55], evidential neural networks [197], imprecise credal decision trees used on their own or in an ensemble [2, 142], etc.

However, producing indeterminate predictions comes with a cost: the uncertainty associated with predictions involving multiple plausible classes requires human intervention to be resolved. Therefore, the challenge of achieving a balance between cautiousness (the ability to avoid making wrong decisions) and determinacy (the ability to make informative predictions) seems essential to cautious classification approaches.

Explainability

Another challenge is the explainability of classifiers. With the development of AI technology, machine learning models play a key role in an increasing number of fields, and their decisions may have a direct impact on human life. However, many modern machine learning models are often described as "black boxes" because their inner workings are concealed or elusive to users, and therefore difficult to be explained and understood, which poses serious challenges to the accountability and credibility of models [141]. In critical scenarios, such as medical diagnosis, financial decision-making, or legal judgments, model predictions often need to meet strict interpretation requirements to ensure transparency and fairness in the decision-making process. In these domains, the lack of explainability even becomes a barrier to their deployment and application [124, 140].

To overcome this problem, eXplainable Artificial Intelligence (XAI) aims to reveal the inner workings of models, facilitating a deeper understanding of their predictive principles. Through XAI, users may attain clearer and more transparent insights into both the model and its predictions, thus enhancing the model's credibility and accountability [13].

In traditional machine learning, a lot of explanation methods have been proposed and extensively applied. These methods can generally be categorized into approaches that establish connections between input features and model outputs (such as feature importance and feature visualization), case-based explanations (such as counterfactual and prototype explanations), and surrogate models with intrinsic interpretability (such as linear regression and decision trees) [5].

However, in the domain of cautious classification, very few works address the problem of explaining the indeterminacy of set-valued predictions, and none of them involve computing feature importance degrees or counterfactual explanations.

Contributions

In this thesis, we detail two main contributions that respectively address the aforementioned challenges in cautious classification.

First, we propose a strategy within the framework of belief functions where we combine imprecise decision trees induced by the imprecise Dirichlet model to construct a cautious classifier, called cautious random forest [234, 230]. This strategy aims to reach a better compromise between the accuracy and the cautiousness of predictions than state-of-the-art aggregation methods for imprecise trees. Additionally, we introduce a cost function specifically designed for binary cautious classifiers to assign weights to trees in the ensemble.

Second, this thesis aims to provide tools to reduce the cost of indeterminacy in imprecise (set-valued) predictions. We address this problem with XAI. More precisely, for any instance for which an indeterminate prediction has been made, we

propose to generate counterfactual examples with desired determinate predictions, which directly allows users to know how to modify certain feature values to resolve the indeterminacy. For this purpose, we propose a framework for generating counterfactual examples, taking into account all desirable properties (validity, proximity, plausibility, actionability, and efficiency) of counterfactual explanations. This framework also makes use of feature importance metrics to accelerate the generation of counterfactual examples.

Structure of the thesis

This thesis is divided into two parts. The first one addresses cautious classification and more particularly cautious random forests. It is structured as follows:

- in Chapter 1, we review the decision-making problem and its operation within different uncertainty modeling frameworks such as precise probabilities, imprecise probabilities, and belief functions.
- In Chapter 2, we give an overview of precise and imprecise (cautious) classification: we outline the problems, we introduce evaluation metrics for classifiers and model comparison, and we discuss some common classifiers for both traditional and cautious classification problems.
- In Chapter 3, we detail our first contribution, a new aggregation method to construct cautious random forests in a binary imprecise classification setting based on pre-trained traditional random forests, the imprecise Dirichlet model, and belief functions.
- In Chapter 4, we extend our contribution to multi-class cautious classification problems by generalizing the averaging and the voting strategies proposed for precise tree ensembles. In both cases, we aim at providing set-valued predictions by maximizing the lower expected utility.

The second part deals with providing explanations for cautious random forests:

- in Chapter 5, we review the concepts of explainable artificial intelligence. After a discussion on the importance of explanations and the desirable properties of explanation methods, we present the most prominent explanation methods and classify them from different perspectives. Additionally, we illustrate different ways to provide explanations for random forests in precise classification problems.

- In Chapter 6, we propose a framework that uses counterfactual examples to explain indeterminate predictions made by a cautious random forest model, which makes it possible to answer the question of why a given input instance is classified indeterminately and how feature values can be identified and modified to achieve a determinate prediction.

Finally, a chapter of Conclusion and perspectives summarizes our main results and presents possible future works.

Publications

Journal papers

1. Haifei Zhang, Benjamin Quost, and Marie-Hélène Masson. "Cautious weighted random forests." Expert Systems with Applications 213 (2023): 118883 [230].

International conference papers

1. Haifei Zhang, Benjamin Quost, and Marie-Hélène Masson. "Cautious Decision-Making for Tree Ensembles." The 17th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2023) [229].

2. Vu-Linh Nguyen, Haifei Zhang, and Sébastien Destercke. "Learning Sets of Probabilities through Ensemble Methods." The 17th European Conference on

Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2023) [150].

3. Haifei Zhang, Benjamin Quost, and Marie-Hélène Masson. "Explaining Cautious Random Forests via Counterfactuals." The 10th International Conference on Soft Methods in Probability and Statistics (SMPS 2022) [231].

4. Haifei Zhang, Benjamin Quost, and Marie-Helène Masson."Cautious Random Forests: a new decision strategy and some experiments." The 12th International Symposium on Imprecise Probability: Theories and Applications (ISIPTA 2021) [234].

French national conference papers

1. Haifei Zhang, Benjamin Quost, and Marie-Hélène Masson. "Explications contrefactuelles pour les forêts aléatoires prudentes." La 31ièmes Rencontres Francophones sur la Logique Floue et ses Applications (LFA 2022) [232].

2. Haifei Zhang, Benjamin Quost, and Marie-Hélène Masson. "Forêts aléatoires prudentes: une nouvelle stratégie de décision et quelques expériences." La 31ièmes Rencontres Francophones sur la Logique Floue et ses Applications (LFA 2022) [233].

Available codes

Explainable cautious random forest: the python codes with scikit-learn style for a cautious random forest model and its explainer including feature importance measures and counterfactual generators are accessible on GitHub via the link https://github.com/Haifei-ZHANG/Explainable-Cautious-Random-Forest

Part I

Cautious random forests

Chapter 1

Cautious decision-making under different frameworks

Many of the problems encountered in everyday life involve making choices and decisions. Decision-making is the process resulting in the selection of an action among several alternatives. Common decision-making examples include deciding what is the best medical treatment for a patient, the best product to recommend to a customer, the best train to take, etc An action may have different consequences (outcomes), depending on the state of nature, e.g., the actual condition of the patient. The best action is determined by the state of nature and by the desirability or the utility of its consequences. For example, recommending a treatment for a person in good health is clearly not appropriate.

In many real-life problems, the state of nature is not known with certainty or, even worse, it is sometimes completely ignored. These two situations are respectively referred to as decision-making under uncertainty and decision-making under complete ignorance. In order to address the problem of decision-making under uncertainty, the theoretical framework of probabilities is often used. However, in some situations, the precise probability of each state of nature is difficult or even impossible to evaluate. Therefore, theoretical frameworks such as imprecise probabilities [14], credal sets [119] and belief functions [50, 180] have been proposed to deal with quantifying the uncertainty in complex systems, in presence of limited data, under subjective judgments.

In this chapter, we present a review of decision-making strategies using various theoretical frameworks and introduce the notion of cautious decision-making, which is a central notion of this thesis. Section 1.1 presents the basic concepts of classical decision theory, for which we rely on the survey article [58]. Then, imprecise probabilities are introduced in Section 1.2, with a focus on credal sets, the imprecise Dirichlet model, and the corresponding decision-making principles. Finally, we introduce the theory of belief functions in Section 1.3, including the representation and the combination of evidence, and finally decision-making with belief functions.

1.1 Decision-making

A decision-making problem consists in selecting an action f from a set of finite possible alternatives $\mathcal{A} = \{f_1, \ldots, f_N\}$ according to a preference relationship [67]. The desirability of each action $f_i \in \mathcal{A}$ given each state of nature $c_k \in \Omega = \{c_1, \ldots, c_K\}$ is usually quantified via a utility function $u : \mathcal{A} \times \Omega \to \mathbb{R}$. Then, a utility matrix $\mathbf{U}$ of dimension $N \times K$ can be constructed, in which the general term $u_{ik} = u(f_i, c_k)$, $i = 1, \ldots, N$, $k = 1, \ldots, K$, represents the utility if action f_i is picked whereas c_k holds.

In real-world decision-making problems, the true state of nature is unknown. Therefore, besides the utility matrix, some description of the uncertainty about the state of nature is necessary to construct the preference relation over actions in $\mathcal{A}$. In the following sections, decision-making under complete ignorance (maximum uncertainty) and decision-making under probabilistic uncertainty will be presented. We refer to the work of Denœux [58] for this part of the literature review.

1.1.1 Preference relation among actions

In order to select the most desirable action, a preference relation should be built over the actions in $\mathcal{A}$. The notation $f \succcurlyeq f'$ means that the action f is at least as desirable as f' for the decision-maker. Respectively, strict preference (f is strictly more desirable than f') and indifference (f and f' are equally desirable) relations are written as $\succ$ and $\sim$.

If a preference relation $\succcurlyeq$ is reflexive (for any $f \in \mathcal{A}$, $f \succcurlyeq f$) and transitive (for any f, f', $f'' \in \mathcal{A}$, if $f \succcurlyeq f'$ and $f' \succcurlyeq f''$, then $f \succcurlyeq f''$), it is called a *preorder*. Additionally, if a preorder is anti-symmetric (for any f, $f' \in \mathcal{A}$, $f \succcurlyeq f'$ and $f' \succcurlyeq f$ imply $f = f'$), it becomes an *order*. A preorder is complete if, for any pair of actions f, $f' \in \mathcal{A}$, the preference between them is known, i.e., either $f \succcurlyeq f'$ or $f' \succcurlyeq f$. Otherwise, it is partial.

In a partial order, an action f is *the greatest element* if it is at least desirable as

any other action, i.e., $\forall f' \in \mathcal{A},\ f \succcurlyeq f'$. If there is a greatest element for a partial order, it must be unique. An action f is a *maximal element* if no other action in $\mathcal{A}$ is strictly preferred to (or dominates) f, i.e., $\nexists f' \in \mathcal{A}$, such that $f' \succ f$. For a partial order, there may be multiple maximal elements. Therefore, the greatest element is always a maximal element, but the converse is usually not true [58, 129].

1.1.2 Decision-making under ignorance

In some situations, the decision-maker may be totally ignorant of the uncertainty about the states of nature, which means that the decision-making is only based on the given utility matrix. We will enumerate some criteria and principles to construct a partial or a complete preference relationship among actions from $\mathcal{A}$ for this case.

Partial preorder

For a pair of actions f_i and f_j from $\mathcal{A}$, f_i is dominated by f_j if f_j is always at least as desirable than f_i, which means that $\forall c_k \in \Omega$, we have $u_{jk} \geq u_{ik}$ and $\exists c_k \in \Omega$, such that $u_{jk} > u_{ik}$. The *non-domination principle* is often used to build a partial preorder, according to which dominated actions should never be selected as desirable actions [191]. However, the preference among the remaining non-dominated actions is unknown. Thus, the preorder is said to be partial.

Complete preorder

We present here some criteria to establish a complete preorder among the non-dominated actions.

- The *maximax rule* [191] compares actions in terms of their most favorable utility across all states of nature, which reflects an extremely optimistic attitude of the decision-maker. Thus, $f_i \succcurlyeq f_j$ if and only if

$$\max_{k=1\ldots K} u_{ik} \geq \max_{k=1\ldots K} u_{jk}. \tag{1.1}$$

- The *maximin rule* [210] considers the least favorable utility across all states of nature for each action. It reflects an extremely pessimistic attitude of the decision-maker. Thus, $f_i \succcurlyeq f_j$ if and only if

$$\min_{k=1\ldots K} u_{ik} \geq \min_{k=1\ldots K} u_{jk}. \tag{1.2}$$

- The *Hurwicz criterion* [100] convexly combines the maximum and the minimum utilities of each action so as to find a compromise between both. Thus, $f_i \succcurlyeq f_j$ if and only if

$$\alpha \min_{k=1\ldots K} u_{ik} + (1-\alpha) \max_{k=1\ldots K} u_{ik} \geq \alpha \min_{k=1\ldots K} u_{jk} + (1-\alpha) \max_{k=1\ldots K} u_{jk}, \tag{1.3}$$

  where $\alpha \in [0,1]$ is called the pessimism index.

- The *Laplace criterion* regards each state of nature as having the same importance and calculates the average utility across all states of nature for each action: $f_i \succcurlyeq f_j$ if and only if

$$\frac{1}{K}\sum_{k=1}^{K} u_{ik} \geq \frac{1}{K}\sum_{k=1}^{K} u_{jk}. \tag{1.4}$$

- The *minimax regret rule* [168] considers action f_i to be at least as desirable as f_j if it has a smaller or equal maximum regret compared with f_j. Thus, $f_i \succcurlyeq f_j$ if and only if

$$\max_{k=1\ldots K} r_{ik} \leq \max_{k=1\ldots K} r_{jk}, \tag{1.5}$$

  where $r_{ik} = \max_{l=1\ldots N}(u_{lk} - u_{ik})$ is the maximum regret of f_i if c_k occurs, i.e., the difference between the utility of the best action and the utility of f_i when the state of nature c_k occurs.

As noted in [58], the Laplace, maximax, maximin, and Hurwicz criteria are special cases of the *Ordered Weighted Average* (OWA) operator, which proceeds, for each action, by defining a distribution of weights for states of nature so as to compute a weighted sum [223]. Suppose states of nature are sorted by descending

order of utility for a certain action f, the above criteria may be retrieved as follows:

- Laplace, by assigning equal weights to all states of nature: $(\frac{1}{K}, \frac{1}{K}, \ldots, \frac{1}{K})$;
- Maximin, by considering the lowest utility: $(0, 0, \ldots, 1)$;
- Maximax, by considering the highest utility: $(1, 0, \ldots, 0)$;
- Hurwicz, by computing a convex sum of the highest and lowest utilities: $(1-\alpha, 0, \ldots, \alpha)$.

The weight $w_k,\ k = 1, \ldots, K$, can be interpreted as the probability that the state with the k-th best outcome will happen. Besides the aforementioned weight assignments, Yager proposed in [223] to determine the weight assignment $\boldsymbol{w}^\star$ that maximizes the entropy under the constraint of the given degree of optimism β:

$$\boldsymbol{w}^\star = \arg\max_{\boldsymbol{w}\in\mathbb{R}_+^K} \sum_{k=1}^{K} -\omega_k \log \omega_k \quad \text{s.t.} \quad \sum_{k=1}^{K} \frac{K-k}{K-1}\omega_k = \beta,\ \sum_{k=1}^{K} \omega_k = 1. \tag{1.6}$$

Example 1.1 (Decision-making under complete ignorance)**.** *In a football match, there are three different outcomes for the home team: win (W), draw (D), and loss (L). Assume that there are three different gambles (actions): different rewards for different outcomes. As a decision-maker, without any other information (under complete ignorance), we need to choose the most desirable gamble according to different decision criteria.*

Table 1.1: Payoff matrix for the football match and calculation for the maximax, maximin, Hurwicz with $\alpha = 0.5$, Laplace and minimax regret criteria. The symbol $\uparrow$ indicates that higher values correspond to more desirable actions. The symbol $\downarrow$ represents the opposite.

f_i	W	D	L	$\max(u_i.) \uparrow$	$\min(u_i.) \uparrow$	$0.5(\max(u_i.) + \min(u_i.)) \uparrow$	$\text{ave}(u_i.) \uparrow$	$\max(r_i.) \downarrow$
f_1	6	-2	-2	6	-2	**2**	**2/3**	9
f_2	-4	7	-4	**7**	-4	1.5	-1/3	10
f_3	-1	-1	3	3	**-1**	1	1/3	**8**

We can find that different decision-making criteria may yield different preference relations: Maximax: $f_2 \succ f_1 \succ f_3$, Maximin: $f_3 \succ f_1 \succ f_2$, Hurwicz with $\alpha = 0.5$: $f_1 \succ f_2 \succ f_3$, Laplace: $f_1 \succ f_3 \succ f_2$, Minimax regret: $f_3 \succ f_1 \succ f_2$.

1.1.3 Decision-making under probabilistic uncertainty

Assume now that the probabilities $\boldsymbol{p} = \{p_1, \ldots, p_K\}$ over Ω and the utility matrix $\mathbf{U}$ are known: it is therefore possible to compute the expected utility of any action $f_i \in \mathcal{A}$ as

$$\mathbb{E}_{\boldsymbol{p}}(f_i, \mathbf{U}) = \sum_{k=1}^{K} u_{ik} p_k. \tag{1.7}$$

This expected utility can be interpreted as the average of utilities weighted by the probabilities of the possible states when action f_i is taken. The *maximum expected utility principle* leads to preferring the action with the highest expected utilities: $f_i \succcurlyeq_{\boldsymbol{p}} f_j$ if and only if

$$\mathbb{E}_{\boldsymbol{p}}(f_i, \mathbf{U}) \geq \mathbb{E}_{\boldsymbol{p}}(f_j, \mathbf{U}). \tag{1.8}$$

Example 1.2 (Decision-making under precise probabilities)**.** *Considering again the payoff matrix in Example 1.1, and historical information about these two teams is used to calculate the probability of each outcome of the match:* $p(W) = 0.45$, $p(D) = 0.3$, $p(L) = 0.25$. *Then, according to Eq. (1.7), the expected utilities for these three actions are* $\mathbb{E}_{\boldsymbol{p}}(f_1, \mathbf{U}) = 1.6$, $\mathbb{E}_{\boldsymbol{p}}(f_2, \mathbf{U}) = -0.7$, $\mathbb{E}_{\boldsymbol{p}}(f_3, \mathbf{U}) = 0$. *In this case, the action* f_1 *is the most desirable.*

1.1.4 Decision-making in classification problems

Classification in machine learning is a special kind of decision-making problem. Considering the set of finite possible classes $\Omega = \{c_1, \ldots, c_K\}$ (states of nature), the action of a classifier (decision-maker) consists in assigning a singleton class or more generally a subset of classes to a given test instance.

In a *precise classification problem*, each action f_i generally corresponds to assigning a single class of Ω to a test instance. Therefore, there are only K possible actions, and the utility matrix $\mathbf{U}$ is of dimension $K \times K$. The general term u_{ik} represents the utility of assigning $c_i \in \Omega$ to an instance when its real class is $c_k \in \Omega$. A commonly used utility function is the 0/1 utility (the utility of misclassification is zero, and that of a correct decision is one), which forms an identity matrix of

dimensions $K \times K$ as utility matrix $\mathbf{U}$.

However, if assigning a subset of classes to a test instance is allowed, then we consider a *cautious classification problem*, also called imprecise or partial classification [164]. In this case, the number of possible actions is 2^K and the dimension of the corresponding utility matrix is $2^K \times K$. Chapter 2 will review several strategies to fix the utility values in such a case.

Hereafter, we use f_i and f_A to denote the action of assigning a singleton $c_i \in \Omega$ and a non-empty subset $A \subseteq \Omega$ to a given instance, respectively. Following [129], the former is called *precise assignment* and the latter *partial assignment*.

There is only one way to achieve a precise classification, which amounts to determining a complete preorder over all precise assignments. Nevertheless, there are two strategies to perform an imprecise classification: one consists in building a partial preorder over precise assignments and the other in building a complete preorder over all possible partial assignments. In the context of imprecise probabilities, the former strategy is commonly employed, whereas in the framework of belief functions, both strategies are often used. In the following sections, we will thoroughly review these two frameworks.

1.2 Imprecise probability theory

Imprecise probability theory provides a framework for expressing uncertainty in a more nuanced and realistic way than classical probability theory. Imprecise probabilities define sets of possible probability values to an event, capturing the inherent vagueness and ambiguity that often accompanies real-world uncertainty [14].

The motivation behind specifying imprecise probabilities lies in the observation that in many practical situations, complete knowledge about the underlying probabilities is unattainable. Complex systems, limited data, subjective judgments, and various sources of uncertainty can make it challenging to quantify probabilities precisely. Imprecise probabilities offer a way to handle such situations by explicitly

acknowledging and incorporating the inherent uncertainty [212].

There are many frameworks of imprecise probabilities, among which lower previsions [201], credal sets [119], probability intervals [47], belief functions [50, 180] offer different perspectives and tools for dealing with uncertainty. In this section, we detail credal sets and probability intervals, before proceeding with belief functions in the next section.

1.2.1 Credal sets and probability intervals

Let $\Omega = \{c_1, \ldots, c_K\}$ be a finite set of possible alternatives. A *credal set* $\mathcal{P}(\Omega)$ is a closed set of probability distributions on Ω. It is often assumed to be convex. In a credal set, there are some probability distributions $\boldsymbol{p} \in \mathcal{P}$ that can not be represented as a strictly convex combination of other probability distributions in the same credal set. They are called *extreme points* of the credal set.

Example 1.3 (Probability simplex)**.** *A probability distribution* $\boldsymbol{p}$ *on* $\Omega = \{c_1, \ldots, c_K\}$ *is a vector* $(p(c_1), \ldots, p(c_k), \ldots, p(c_K))$ *such that* $\forall c_k \in \Omega,\ p(c_k) \geq 0$ *and* $\sum_{c_k \in \Omega} p(c_k) = 1$*, which can be regarded as a point in the space of a simplex of* $K - 1$ *dimension.*

If $\Omega = \{c_1, c_2, c_3\}$*, each probability distribution* $\boldsymbol{p}$ *can be represented in an equilateral triangle. The probability for* c_k *is defined as the distance between* $\boldsymbol{p}$ *and the edge that excludes* c_k*. Fig.1.1 provides an illustration.*

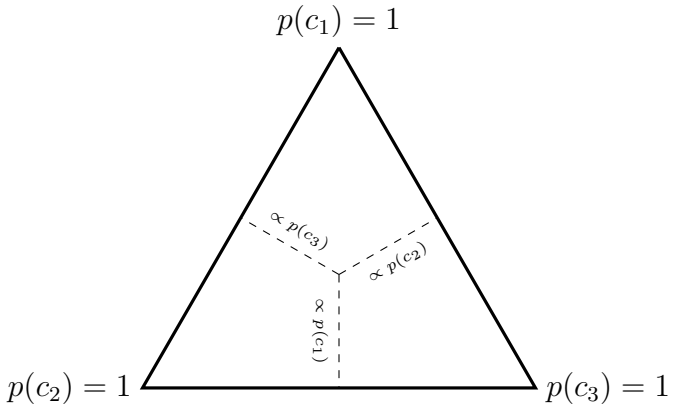


Figure 1.1: Representation of a probability distribution for Ω with three states.

Example 1.4 (Credal set)**.** *Suppose a credal set* $\mathcal{P}$ *on* $\Omega = \{c_1, c_2, c_3\}$ *defined by the following two constraints:*

$$p(c_1) \geq \frac{1}{3}, \quad p(c_2) - 2p(c_3) \geq 0,$$

which corresponds to the region in green represented in Fig. 1.2. Correspondingly, the union of the green and the blue regions is the credal set constrained by $p(c_1) \geq \frac{1}{3}$*, and the union of the green and the red regions is the one constrained by* $p(c_2) - 2p(c_3) \geq 0$*.*

A convex credal set can be equivalently represented by its extreme points. In this example, there are only three extreme points:

$$\boldsymbol{p}_1 = (1, 0, 0), \quad \boldsymbol{p}_2 = (\frac{1}{3}, \frac{2}{3}, 0), \quad \boldsymbol{p}_3 = (\frac{1}{3}, \frac{4}{9}, \frac{2}{9}).$$

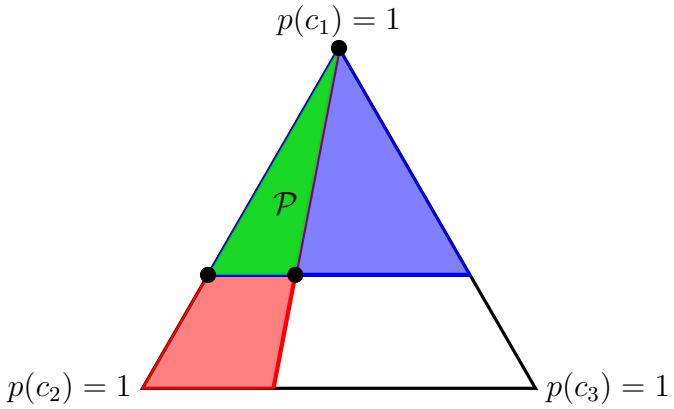


Figure 1.2: Example of credal set.

As a special case of credal sets, *probability intervals* represent a piece of probabilistic knowledge on Ω via a set of probability intervals on singletons:

$$\mathcal{I}(\Omega) = \{\mathcal{I}_k = [l_k, u_k], \ k = 1, \ldots, K\}, \tag{1.9}$$

where l_k and u_k are, respectively, the lower and upper probability bounds of alternative $c_k \in \Omega$, such that $0 \leq l_k \leq u_k \leq 1$. Obviously, the credal set associated with $\mathcal{I}$ is

$$\mathcal{P}(\mathcal{I}) = \{\boldsymbol{p} \mid l_k \leq p(c_k) \leq u_k, \ k = 1, \ldots, K; \sum_{k=1}^{K} p(c_k) = 1\}. \tag{1.10}$$

A set of probability intervals $\mathcal{I}(\Omega)$ is said to be *proper* if the sum of the lower probabilities bounds is less than or equal to 1 and the sum of upper probability bounds is larger than or equal to 1:

$$\sum_{k=1}^{K} l_k \leq 1 \leq \sum_{k=1}^{K} u_k. \tag{1.11}$$

If a set of probability intervals is proper, its associated credal set is guaranteed to be nonempty. A proper set of probability intervals $\mathcal{I}(\Omega)$ is said to be *reachable* if and only if

$$\begin{aligned} \sum_{j=1, j\neq k}^{K} l_j + u_k \leq 1, \ \forall k = 1, \ldots, K, \\ \sum_{j=1, j\neq k}^{K} u_j + l_k \geq 1, \ \forall k = 1, \ldots, K. \end{aligned} \tag{1.12}$$

If a set of probability intervals $\mathcal{I}(\Omega)$ is proper and reachable, the lower and upper coherent probabilities of each alternative are defined as

$$\underline{p}(c_k) = l_k, \quad \overline{p}(c_k) = u_k, \quad \forall k = 1, \ldots, K. \tag{1.13}$$

Example 1.5 (Set of probability intervals)**.** *We consider a set of probability intervals on* $\Omega = \{c_1, c_2, c_3\}$ *defined as follows:*

$$p(c_1) \in [0.2, 0.5], \quad p(c_2) \in [0.3, 0.6], \quad p(c_3) \in [0.1, 0.4].$$

It is easy to check that the given set of probability intervals is proper and reachable. The corresponding credal set $\mathcal{P}$ *is illustrated as the green region in Fig. 1.3.*

Since each probability interval can be seen as two constraints on a single state, the edges of the credal set formed by a set of probability intervals are always parallel to one edge of the equilateral triangle (simplex). In this example, there are six extreme points for the credal set:

$$\begin{aligned} &\boldsymbol{p}_1 = (0.5, 0.4, 0.1), \quad \boldsymbol{p}_2 = (0.3, 0.6, 0.1), \quad \boldsymbol{p}_3 = (0.2, 0.6, 0.2), \\ &\boldsymbol{p}_4 = (0.2, 0.4, 0.4), \quad \boldsymbol{p}_5 = (0.3, 0.3, 0.4), \quad \boldsymbol{p}_6 = (0.5, 0.3, 0.2). \end{aligned}$$

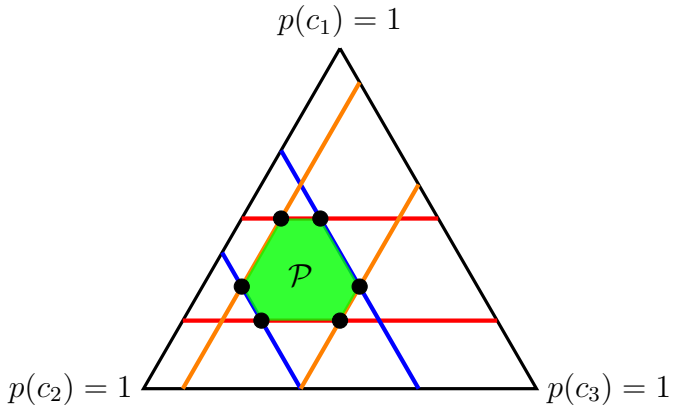


Figure 1.3: Example of credal set induced by probability intervals.

1.2.2 Imprecise Dirichlet model

Proposed by Walley [211], the *imprecise Dirichlet model (IDM)* is a tool to construct proper and reachable sets of probability intervals based on observations.

Let $\Omega = \{c_1, \ldots, c_K\}$ be the aforementioned set of $K \geq 2$ mutually exclusive alternatives or classes, and let $\pi_k = p(c_k)$, with $\pi_k \geq 0$ and $\sum_{k=1}^{K} \pi_k = 1$, for $k = 1, \ldots, K$. Assume that N independent and identically distributed (iid) observations have been sampled from an unknown multinomial distribution $\mathcal{M}(N; \pi_1, \ldots, \pi_K)$: let n_k denote the corresponding number of occurrences of c_k, with $\sum_{k=1}^{K} n_k = N$. For the sake of simplicity, we write $\boldsymbol{n} = \{n_1, \ldots, n_K\}$ and $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_K\}$. The likelihood of the parameter vector writes as

$$L(\boldsymbol{\pi}|\boldsymbol{n}) \propto \prod_{k=1}^{K} \pi_k^{n_k}. \tag{1.14}$$

In a standard Bayesian setting, prior knowledge over the probabilities π_k can be specified using the conjugate Dirichlet distribution $Dir(s, \boldsymbol{\alpha})$, with $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$ and $\sum_{k=1}^{K} \alpha_k = s$:

$$p(\boldsymbol{\pi}|\boldsymbol{\alpha}) \propto \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}. \tag{1.15}$$

Note that each parameter can be decomposed into $\alpha_k = s\, t_k$, with $s \geq 0$, $0 \leq t_k \leq 1$, and $\sum_{k=1}^{K} t_k = 1$: then, the parameters t_k, $k = 1, \ldots, K$, are the prior frequencies with $\mathbb{E}(\pi_k) = t_k$, whereas s corresponds to the prior's global strength. The posterior

distribution then writes as

$$p(\boldsymbol{\pi}|\boldsymbol{n},\boldsymbol{\alpha}) \propto \prod_{k=1}^{K} \pi_k^{n_k+st_k-1}, \tag{1.16}$$

which is also a Dirichlet distribution $Dir(N+s,\boldsymbol{t}^*)$ due to conjugacy, where $t_k^* = (n_k+st_k)/(N+s)$.

In standard Bayesian inference, the parameters s and $\boldsymbol{t} = \{t_1,\ldots,t_K\}$ are determined in advance, which results in point estimates for the frequencies π_k. However, in the IDM, a set of Dirichlet distributions is defined by considering all vectors $\boldsymbol{t}$ satisfying the constraints $0 \leq t_k \leq 1$ and $\sum_{k=1}^{K} t_k = 1$. Taking this set as a prior amounts to making as few assumptions as possible regarding $\boldsymbol{\pi}$, i.e., the prior is near-ignorant [131]. As a result, the posterior information is no longer a single distribution, but a set of distributions, from which it is possible to deduce posterior lower and upper bounds on the probabilities of alternatives, reached respectively when $t_k \to 0$ and $t_k \to 1$:

$$\underline{\mathbb{E}}(\pi_k|\boldsymbol{n},s) = \frac{n_k}{N+s}, \quad \overline{\mathbb{E}}(\pi_k|\boldsymbol{n},s) = \frac{n_k+s}{N+s}, \quad i = 1,\ldots,K. \tag{1.17}$$

The set of probability intervals is denoted as

$$\mathcal{I}(\Omega|\boldsymbol{n},s) = \left\{\mathcal{I}_k = \left[\underline{p}_k,\overline{p}_k\right] = \left[\frac{n_k}{N+s},\frac{n_k+s}{N+s}\right],\ k = 1,\ldots,K\right\}. \tag{1.18}$$

Note that the parameter s remains to be chosen in advance: it can be interpreted as a number of virtual instances with unknown class information. Although several studies have been conducted with regard to choosing an appropriate value for s [4], this problem remains open. In practice, values of $s = 1$ or $s = 2$ are often picked [211].

1.2.3 Decision-making with imprecise probabilities

Let $\mathcal{P}$ be a credal set on $\Omega = \{c_1,\ldots,c_K\}$ and f_i be a precise assignment associated with a utility matrix $\mathbf{U}$. The expected utility defined in Eq. (1.7) can be extended

to the lower and upper expected utilities [14], defined as

$$\underline{\mathbb{E}}_{\mathcal{P}}(f_i, \mathbf{U}) = \min_{\boldsymbol{p}\in\mathcal{P}} \mathbb{E}_{\boldsymbol{p}}(f_i, \mathbf{U}) = \min_{\boldsymbol{p}\in\mathcal{P}} \sum_{k=1}^{K} u_{ik} p_k, \tag{1.19}$$

and

$$\overline{\mathbb{E}}_{\mathcal{P}}(f_i, \mathbf{U}) = \max_{\boldsymbol{p}\in\mathcal{P}} \mathbb{E}_{\boldsymbol{p}}(f_i, \mathbf{U}) = \max_{\boldsymbol{p}\in\mathcal{P}} \sum_{k=1}^{K} u_{ik} p_k. \tag{1.20}$$

The lower and upper expected utilities can be used to build a complete preorder among all singleton alternatives in Ω. Then, the *maximin* and the *maximax* criteria can be applied, respectively, on $\underline{\mathbb{E}}_{\mathcal{P}}$ and $\overline{\mathbb{E}}_{\mathcal{P}}$ to select the most desirable action.

The lower and upper expected utilities can also be used to build a partial preorder among all singleton alternatives of Ω. For instance, the *interval dominance rule* states that

$$f_i \succcurlyeq_{id} f_j, \text{ if } \underline{\mathbb{E}}_{\mathcal{P}}(f_i, \mathbf{U}) \geq \overline{\mathbb{E}}_{\mathcal{P}}(f_j, \mathbf{U}). \tag{1.21}$$

With this criterion, all pairs of actions have to be compared, and non-dominated actions form the final choice. It is a very cautious criterion, which often leads to considering many pairs of actions as incomparable.

We mention here two other decision criteria that are less cautious than the interval dominance rule. For the *maximality criterion* [212], the preference between two actions is defined as

$$f_i \succcurlyeq_{max} f_j, \text{ if } \underline{\mathbb{E}}_{\mathcal{P}}(f_i - f_j, \mathbf{U}) \geq 0, \tag{1.22}$$

which means that in order to be more desirable, action f_i must have a higher or equal utility compared with action f_j for any distribution in $\mathcal{P}$.

The *E-admissibility criterion* considers an action as more desirable if there exists a probability distribution $\boldsymbol{p} \in \mathcal{P}$ such that all other actions have a smaller expected utility than f_i [119]. In other terms, f_i is E-admissible if

$$\exists \boldsymbol{p} \in \mathcal{P}, \text{ s.t. } \forall f_j \in \mathcal{A}, \ \mathbb{E}_{\boldsymbol{p}}(f_i) \geq \mathbb{E}_{\boldsymbol{p}}(f_j). \tag{1.23}$$

In the framework of imprecise probabilities, we can find that the three aforementioned decision-making criteria induce cautious (set-valued) predictions by constructing partial preorders among precise assignments. Let $\mathcal{F}^*_{e\text{-}ad}$, $\mathcal{F}^*_{max}$, $\mathcal{F}^*_{id}$ be the sets of finally selected maximal actions according to the E-admissibility, maximality, and interval dominance criteria, respectively; we have

$$\mathcal{F}^*_{e\text{-}ad} \subseteq \mathcal{F}^*_{max} \subseteq \mathcal{F}^*_{id}, \tag{1.24}$$

which means that the interval dominance criterion is the most cautious, followed by the maximality criterion, and the E-admissibility criterion is the least cautious [55].

1.3 Theory of belief functions

The theory of belief functions, also referred to as Dempster-Shafer theory (DST) or the theory of evidence, is a mathematical framework for dealing with uncertainty and reasoning with incomplete or conflicting information [50, 180]. It provides a formal way to combine and reason with uncertain information from multiple sources, allowing for a more robust and cautious approach to decision-making. DST has found applications in various fields, such as information fusion [64, 121, 221], pattern recognition [55, 56, 97, 197], semantic segmentation [198], fault diagnosis [220, 222, 235], etc.

In this section, some basic concepts of the theory of belief functions will be reviewed, including different representations of evidence, approaches to combining pieces of evidence, and decision-making strategies based on belief functions.

1.3.1 Representation of evidence

Let $\Omega = \{c_1, \ldots, c_K\}$, $K \geq 2$, be a finite set that contains all the possible, mutually exclusive states of nature for a question, referred to as the *frame of discernment*. Given a piece of evidence, the information is represented by a *mass function*, also referred to as a *basic probability assignment* (BPA) or a *basic belief assignment*

(BBA), which is a mapping $m : 2^{\Omega} \rightarrow [0, 1]$, such that

$$m(\emptyset) = 0 \text{ and } \sum_{A \subseteq \Omega} m(A) = 1. \tag{1.25}$$

The value $m(A)$ measures the degree of evidence supporting the fact that the true state is in A, but no more specific proposition (any subset of A). A subset $A \subseteq \Omega$ is called a *focal set* or *focal element* if $m(A) > 0$. If there is only one focal element, then m is said to be *logical*; and if furthermore, the unique focal element is Ω, m is said to be *vacuous* (it represents a total ignorance). A mass function is Bayesian if all of its focal elements are singletons, in which case it is reduced to a precise probability distribution. This framework can therefore be seen as an extension of both set theory and classical probability theory.

For any subset $A \subseteq \Omega$, the uncertainty of the proposition that the true state lies in A can be quantified by the degrees of *belief* $Bel(A)$ and *plausibility* $Pl(A)$, which are defined, respectively, as:

$$Bel(A) = \sum_{B \subseteq A} m(B), \quad \forall A \in \Omega, \tag{1.26}$$

and

$$Pl(A) = \sum_{B \cap A \neq \emptyset} m(B), \quad \forall A \in \Omega. \tag{1.27}$$

$Bel(A)$ and $Pl(A)$ measure the support (belief) and compatibility (plausibility), respectively, associated with the proposition that the truth lies in A. For a normal mass function, i.e., $m(\emptyset) = 0$, it is obvious that $Bel(\emptyset) = Pl(\emptyset) = 0$, $Bel(\Omega) = Pl(\Omega) = 1$, and $Bel(A) \leq Pl(A)$. The belief and plausibility measures are also dual since for $\forall A \subseteq \Omega$, $Bel(A) = 1 - Pl(\overline{A})$, and $Pl(A) = 1 - Bel(\overline{A})$, with $\overline{A}$ the complement of A. Belief and plausibility functions can also be transferred to a mass function through the *inverseMöbius transform*:

$$m(A) = \sum_{B \subseteq A} (-1)^{|A|-|B|} Bel(B), \quad \forall A \subseteq \Omega, \tag{1.28}$$

or equivalently

$$m(A) = \sum_{B \subseteq A} (-1)^{|A|-|B|+1} Pl(\overline{B}), \quad \forall A \subseteq \Omega. \tag{1.29}$$

Therefore, there is a one-to-one correspondence between mass, belief, and plausibility functions. Besides, a mass function can be transformed into a probability distribution through the *pignistic transformation* [185, 186]:

$$BetP(c_k) = \sum_{A \subseteq \Omega, c_k \in A} \frac{m(A)}{|A|}, \quad \forall c_k \in \Omega, \tag{1.30}$$

in which the mass of focal sets is equally assigned to their elements.

1.3.2 Combination of evidence

Sometimes, different pieces of evidence may be provided about the same variable of interest. They have to be combined into a single mass function which will then be exploited for reasoning. There are several available combination methods, among which *Dempster's rule* is the fundamental one [50]. Given two independent mass functions m_1 and m_2 defined on the same frame of discernment Ω, Dempster's combination rule combines them into one mass function m via

$$m(A) = (m_1 \oplus m_2)(A) = \frac{1}{1-\mathcal{K}} \sum_{B \cap C = A} m_1(B) m_2(C), \quad \forall A \subseteq \Omega, \tag{1.31}$$

where $\mathcal{K}$ is the degree of conflict between the two mass functions, defined as:

$$\mathcal{K} = \sum_{B \cap C = \emptyset} m_1(B) m_2(C)). \tag{1.32}$$

Dempster's combination rule is sometimes called the *orthogonal sum* of m_1 and m_2, and requires that the mass functions to be combined are independent and their conflict is smaller than one. This operation is commutative and associative, which makes it possible to sequentially combine a series of evidence in any order.

However, due to the fact that Dempster's combination rule discards conflict, it may produce counter-intuitive results when facing highly conflicting mass functions.

Therefore, several alternative combination rules have been proposed by considering different strategies to deal with conflict [65, 118, 184, 187, 224]. Each method has its strengths and limitations, and selecting the most appropriate one involves considering the context and characteristics of the problem at hand.

1.3.3 Decision-making with belief functions

Assume now the knowledge of the class of the test instance is represented by a mass function m; there are several criteria to make decisions under the framework of belief functions [58].

If only singleton assignments are considered, a convenient way to build a complete preference order is to transform the mass function m into a probability distribution according to Eq. (1.30), then calculate the expected utility of each action that assigns a singleton class. This expected utility is called the *pignistic expected utility*, which is defined as:

$$\mathbb{E}_{BetP}(f_i, \mathbf{U}) = \sum_{k=1}^{K} BetP(c_k) u_{ik}. \tag{1.33}$$

However, actions f_i may not be restricted to assigning a single class: we may consider subsets $A \subseteq \Omega$ of classes. Under this setting, the expected utility criterion may be extended to the *lower and upper expected utilities*, respectively defined as the weighted averages of the minimum and maximum utility within each focal set:

$$\underline{\mathbb{E}}_m(f_A, \mathbf{U}) = \sum_{B \subseteq \Omega} m(B) \min_{c_k \in B} u_{Ak}, \tag{1.34}$$

and

$$\overline{\mathbb{E}}_m(f_A, \mathbf{U}) = \sum_{B \subseteq \Omega} m(B) \max_{c_k \in B} u_{Ak}. \tag{1.35}$$

It is obvious that $\underline{\mathbb{E}}(m, f_i, \mathbf{U}) \leq \overline{\mathbb{E}}(m, f_i, \mathbf{U})$ and only when m is Bayesian, the equality applies, as we retrieve the probabilistic case. The *Hurwicz expected utility* is a convex combination of $\underline{\mathbb{E}}_m(f_A, \mathbf{U})$ and $\overline{\mathbb{E}}_m(f_A, \mathbf{U})$, defined as:

$$\mathbb{E}_{m,\alpha}(f_A, \mathbf{U}) = \alpha \underline{\mathbb{E}}(m, f_i, \mathbf{U}) + (1 - \alpha) \overline{\mathbb{E}}(m, f_i, \mathbf{U}), \tag{1.36}$$

where $\alpha \in [0, 1]$ is called the pessimism index.

The minimax regret criterion can also be extended to belief functions. The regret that action f_A is chosen whereas state c_k occurs is defined as $r_{Ak} = \max_B u_{Bk}$, $\forall B \subseteq \Omega$. The *expected maximal regret* of action A is defined as

$$\overline{R}_m(f_A, \mathbf{U}) = \sum_{B \subseteq \Omega} m(B) \max_{c_k \in B} r_{Ak}. \tag{1.37}$$

It should be noted that if the class assignments for Eq. (1.34) to Eq. (1.37) are restricted to singletons, i.e., $|A| = 1$, then, all these four expected measures lead to computing complete preorders among all possible precise assignments and the one that reaches the highest expected utility or the lowest expected maximal regret will be selected, which results in precise predictions [57].

Otherwise, if all possible partial assignments are considered, i.e., any subset $A \subseteq \Omega$, the lower, upper, Hurwicz expected utilities, and the expected maximal regret establish complete preorders among partial assignments, and the selection of the subset that reaches the highest expected utility or the lowest expected maximal regret leads to set-valued cautious predictions [129].

1.4 Conclusion

In this chapter, we reviewed the decision-making problem within different frameworks. Starting with the basic definition of the decision problem and the concept of preference relationship, we explored decision-making in the absence of uncertainty, under a probabilistic framework, and for classification tasks. Furthermore, we presented the important framework of imprecise probabilities, focusing on the notion of credal set, and a convenient way of obtaining credal sets: the imprecise Dirichlet model. We also provided decision-making criteria based on imprecise probabilities. Finally, we introduced fundamental concepts of the theory of belief functions and associated decision-making strategies.

Chapter 2

Traditional and cautious classification

Machine learning algorithms have been applied to various fields with remarkable success, such as loan approval [16, 10], medical diagnosis [74], recommendation systems [102], and autonomous driving [135]. In this chapter, we will review machine learning algorithms from the perspective of precise and imprecise (cautious) classification problems. For each kind of classification problem, we provide reminders about the problem, performance evaluations, as well as several commonly used classifiers. Section 2.1 deals with traditional precise classification and Section 2.2 is devoted to cautious classification.

2.1 Precise classification

Traditional classification algorithms aim to learn a model to predict the class of new observations based on training data. With a large amount of training data and complex model design, this paradigm has achieved great success in classification tasks and has been deployed in many fields [130].

2.1.1 Problem statement

A traditional classification problem consists in assigning a single class to a given input instance based on its feature values. To do so, a classifier should be learned from training data, for which both the input vectors and class values have been observed. Assume the problem is related to an input random vector $\mathbf{X} = \{\mathrm{X}^1, \ldots, \mathrm{X}^M\}$ of M dimensions and an output random variable Y, whose possible values (classes) belong to $\Omega = \{c_1, \ldots, c_K\}$, $K \geq 2$. More formally, let us write the input space as $\boldsymbol{\mathcal{X}} = \{\mathcal{X}^1, \ldots, \mathcal{X}^M\}$, and the output space as $\mathcal{Y} = \Omega$. The objective of the classification problem is to learn a function (classifier) $\boldsymbol{h} : \mathcal{X} \to \mathcal{Y}$ by minimizing a predefined cost (risk) measurement function on the observed training dataset $\mathcal{D}_{train} = \{(\boldsymbol{x}_i, y_i),\ i = 1, \ldots, N_{train}\}$. In this section, we only consider the 0/1 cost, i.e., for $(\boldsymbol{x}, y)$ and $\hat{y} = h(\boldsymbol{x})$, $\boldsymbol{c}(\hat{y}, y) = \mathbb{1}(\hat{y} \neq y)$, where $\mathbb{1}(.)$ is the indicator function.

2.1.2 Evaluation metrics

A classifier is commonly evaluated on a separate test set $\mathcal{D}_{test} = \{(\boldsymbol{x}_i, y_i),\ i = 1, \ldots, N_{test}\}$. Considering a binary classification problem, i.e., $\Omega = \{c_1, c_2\}$, where c_1 and c_2 are the positive and negative classes, respectively, the results obtained on the test set can be presented via a *confusion matrix*, as presented in Table 2.1.

Table 2.1: Confusion matrix.

	Actual $y = c_1$	Actual $y = c_2$
Predicted $\hat{y} = c_1$	True Positive (TP)	False Positive (FP)
Predicted $\hat{y} = c_2$	False Negative (FN)	True Negative (TN)

Based on the confusion matrix, several evaluation metrics can be defined as follows:

- the *accuracy* counts the proportion of test instances correctly classified,

$$Accuracy = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \mathbb{1}(\boldsymbol{h}(\boldsymbol{x}_i) = y_i) = \frac{TP + TN}{TP + FP + FN + TN}; \tag{2.1}$$

- the *precision* indicates, among the instances predicted as the positive class, the proportion of them that are actually positive,

$$Precision = \frac{TP}{TP + FP}; \tag{2.2}$$

- the *recall* reveals the proportion of actually positive instances that are correctly classified,

$$Recall = \frac{TP}{TP + FN}; \tag{2.3}$$

- the F_1 score is the harmonic mean of precision and recall, which is appropriate when the positive and negative classes are unbalanced,

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}. \tag{2.4}$$

In binary classification problems, the *receiver operating characteristic* (ROC) curve is also commonly used to evaluate the performance of binary classifiers [139, 240]. The ROC curve illustrates the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings that determine the assignment of class in a binary classification problem. The *area under the ROC curve* (AUC) measures the classifier's ability to distinguish between positive and negative instances, and it serves as an effective measure of classification performance, particularly in unbalanced binary classification scenarios where the consequences of false negatives or false positives are different [91].

In order to evaluate the aforementioned metrics, *cross-validation* is often applied. The idea is to repeatedly divide the dataset into different disjoint batches of training and test instances to train and test the classifier successively, so as to calculate average performance metrics to evaluate the generalization ability of the classifier. Common cross-validation techniques include K-fold cross-validation and leave-one-out cross-validation. Based on cross-validation, hypothesis testing is a usual way to determine whether there are significant differences between classifiers [51]. When confronted with a single dataset, the *paired t-test*, *McNemar's test* [136], and the *Wilcoxon signed-rank test* [215] emerge as applicable tools for studying whether the observed differences in performance are significant. On the other hand, if multiple datasets are used in the comparison procedure, *ANOVA* [189], the *Friedman test* [79], and the *Nemenyi test* [147] can be employed.

2.1.3 Single classifiers

We can distinguish between discriminative and generative models. Given an input instance $\boldsymbol{x}$, a discriminative model directly undertakes the task of estimating the posterior probabilities $p(c_k|\boldsymbol{x})$, $\forall c_k \in \Omega$. Conversely, a generative model proceeds by estimating the joint distribution of the feature vector and the class label $p(c_k, \boldsymbol{x})$, $\forall c_k \in \Omega$. Subsequently, it can also provide the class-conditional posterior probabilities using Bayes theorem. We review in the following some of the most popular classifiers.

Discriminative classifiers

The *K-nearest neighbors* (KNN) [38] method is introduced as a simple yet effective non-parametric algorithm, which identifies the K-nearest neighbors based on a chosen distance metric and chooses the majority class among these neighbors as the predicted class. The number of nearest neighbors determines the level of local generalization: smaller values tend to capture fine-grained details, while larger values provide a smoother decision boundary. In contrast, *logistic regression* [113] postulates a parametric model of the posterior probabilities, which are obtained by applying successively a linear transform and the logistic (or sigmoid) function to an instance; the model is estimated by maximizing the likelihood of the training data. A *decision tree* [24] recursively partitions the input data (and thus the input space) based on an impurity criterion until the input data in each region are pure (or almost pure); a prediction is made based on the decision associated with the region in which a test instance falls. *Support vector machines* (SVM) [93] seek to find an optimal hyperplane that separates the training data into different classes with the maximum margin. In cases where linear separation is unattainable, SVM employs kernel functions to map the input space into a higher-dimensional feature space, enabling nonlinear decision boundaries.

Generative classifiers

The *naive Bayesian classifier* [175] is a widely adopted generative model in the domain of machine learning. Its core assumption, referred to as the "naive" assumption, assumes the conditional independence of features given the class label. This assumption allows naive Bayes models to calculate the conditional probabilities of features given each class and subsequently employs Bayes' theorem to derive the posterior probabilities of classes for the given input. The classification process entails assigning the input to the class with the highest posterior probability.

Linear and quadratic discriminant analysis (LDA and QDA) [194], are other examples of generative probabilistic techniques. LDA operates under the assump-

tion that the conditional probability distributions of features given each class are multivariate Gaussian distributions, characterized by equal covariance matrices. It endeavors to learn a linear separation between the classes, aiming to identify a linear decision boundary that maximizes the ratio of between-class variance to within-class variance. In contrast, QDA relaxes the assumption of equal covariance, enabling each class to be characterized by a specific covariance matrix. Consequently, QDA estimates quadratic decision boundaries between the classes.

Beyond classification problems, generative models find utility in diverse domains, including clustering and density estimation, e.g., Gaussian mixture models [172], as well as time series processing, with hidden Markov models [66] being frequently employed.

2.1.4 Ensemble learning

Ensemble learning refers to algorithms that combine the predictions of several classifiers so as to improve classification accuracy. Ensemble learning can be divided into two categories, based on the classifiers being trained independently from each other or not. *Dependent methods* include boosting [75] and stacking [219] algorithms. *Independent approaches* notably include bagging [23] and random forest [25].

Boosting is a family of ensemble algorithms that can leverage "weak classifiers" to build strong classifiers through a sequential concatenation approach, e.g., AdaBoost [76], XGBoost [33], etc. The underlying mechanism of boosting involves training a base classifier on the initial dataset and adjusting the distribution for the training instances based on the performance of the base classifier. This adjustment aims to provide more attention to the samples that are previously misclassified by the base classifier in the subsequent classifiers. The adjusted training set is then used to train the next base classifier. This process is repeated until reaching the predetermined number of base classifiers. In the prediction phase, an aggregation of the predictions from all base classifiers is performed by applying appropriate weights for base classifiers.

Stacking employs meta-learning algorithms to learn how to combine the outputs of base classifiers. In the stacking approach, the classifiers trained on the training set are referred to as base classifiers, while the classifier used to fuse the predictions of the base classifiers is known as the meta-classifier. Stacking begins by training the base classifiers on the original training set. Subsequently, the outputs of the base classifiers on the training set are utilized as input features to train the meta-classifier. However, this implementation method carries a higher risk of over-fitting. To mitigate this, a common practice is to employ K-fold cross-validation on base classifiers. This involves using samples that were not used to train the base classifiers as training samples for the meta-classifier, thus reducing the risk of over-fitting.

Bagging employs bootstrap sampling, a method of sampling with replacement, to construct multiple datasets of the same size as the original training sets. In each of these datasets, some samples may appear multiple times, while others may never appear. The idea behind bagging is to train base classifiers on each of these sampled datasets and subsequently aggregate their predictions to make decisions. The introduction of sample randomness by bootstrap sampling aims to enhance diversity among the base classifiers, thereby reducing the variance of final predictions and improving its generalization capability.

Random forests

A *random forest* [25], a variant of bagging, is an ensemble learning technique based on the combination of decision trees. This approach is very popular due to its ability to reach excellent generalization performances and avoid over-fitting issues, compared to a single decision tree. Each decision tree in a random forest is trained without pruning on a bootstrap replicate of the original training set. Training samples that are not selected for training a specific tree are called "out-of-bag samples" for that tree. Trees in a random forest are classically grown, i.e., by determining the split which achieves the highest homogeneity (using, e.g., information gain for ID3 [166], information gain ratio for C4.5 [165], or the Gini index for CART [24]). The main difference between a tree in a random forest, with respect to a single tree, is

that the candidate features for each split are randomly selected among all features. If a node cannot be split (homogeneity cannot be improved, the maximum depth has been reached, or the minimum node size is attained), it will be regarded as a terminal node or a leaf that is used to classify test samples.

The number of candidate features for each split thus directly impacts the diversity of trees in the ensemble. Besides, the minimum size of terminal nodes, or alternatively the maximum depth of the tree, makes it possible to control the tree complexity and therefore its ability to fit the training data (low bias). In a random forest, trees are constructed so as to have very low bias and are consequently generally not pruned. The total number T of trees in the forest influences the variance of predictions (the larger the forest, the more stable the predictions). Combining a large number of decision trees makes it possible to exploit the diversity granted by both feature and sample randomness, and helps to limit the detrimental influence of outliers [85], ultimately improving generalization performances.

Let $\mathbf{H}$ be a random forest of T decision trees $\boldsymbol{h}_t$: $\mathbf{H} = \{\boldsymbol{h}_t,\ t = 1, \ldots, T\}$. For a given test instance $\boldsymbol{x} \in \mathcal{X}$, let $n_{tk}(\boldsymbol{x})$ denote the number of training samples of class c_k falling into the same leaf as $\boldsymbol{x}$ for tree $\boldsymbol{h}_t$. The probability of each class $c_k \in \Omega = \{c_1, \ldots, c_K\}$ estimated by each tree is then defined as

$$p_{tk} = p(c_k|\boldsymbol{x}, \boldsymbol{h}_t) = \frac{n_{tk}(\boldsymbol{x})}{\sum_{k=1}^{K} n_{tk}(\boldsymbol{x})}. \tag{2.5}$$

In order to aggregate the trees, the *averaging* strategy computes the mean class probabilities across all trees:

$$p_k^{ave} = \sum_{t=1}^{T} w_t \cdot p_{tk}, \tag{2.6}$$

where w_t denotes the weight of tree $\boldsymbol{h}_t$ such that $w_t > 0$ and $\sum_{t=1}^{T}\ w_t = 1$. Alternatively, the *voting* strategy requires trees to make their own decisions first and then

counts the proportion of votes for each class:

$$p_k^{vote} = \sum_{t=1}^{T} w_t \cdot \mathbb{1}(p_{tk} > p_{tk'}, \forall k' \neq k). \tag{2.7}$$

One natural choice is to assume all weights to be equal across trees, i.e., weights are set to $w_t = 1/T$ for $t = 1, \ldots, T$. We stress that both approaches give an estimate of the posterior probability distribution over the classes. Thus, a decision can be made for $\boldsymbol{x}$ by picking the most probable class that maximizes the expected utility. Fig. 2.1 represents the decision-making process for a given instance $\boldsymbol{x}$ and a trained random forest $\mathbf{H}$.

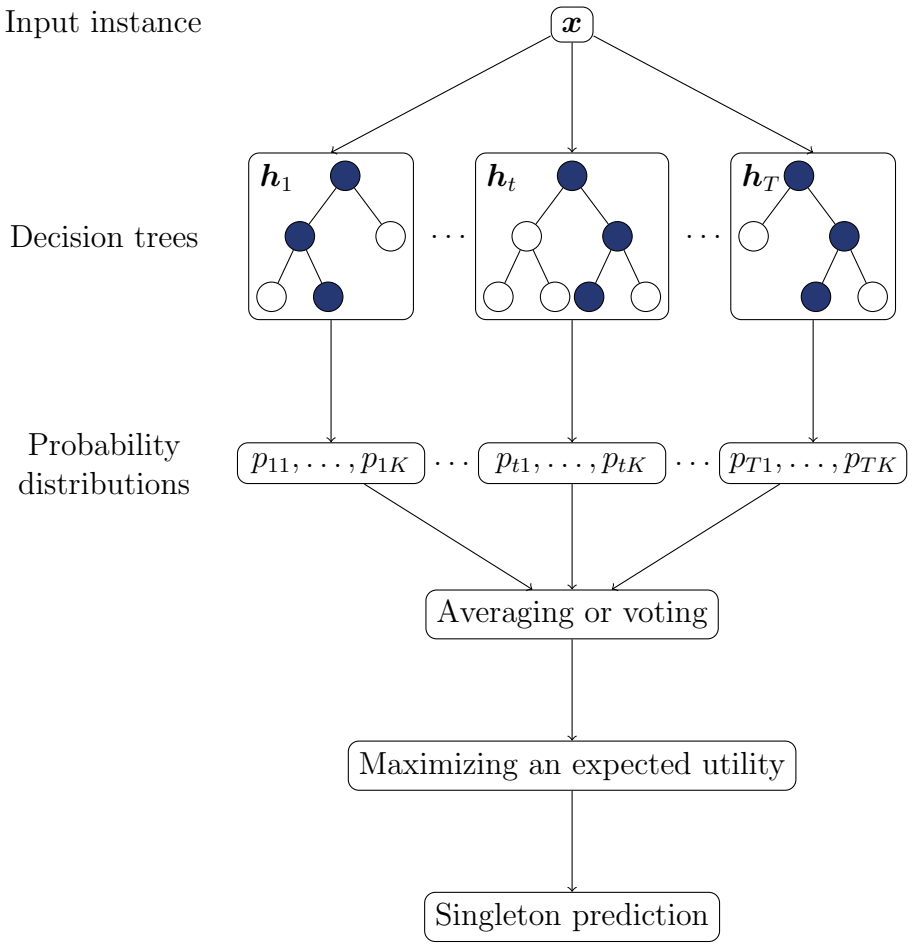


Figure 2.1: Decision-making process of random forests.

2.2 Cautious classification

Traditionally, classification models make precise (determinate) decisions, in the form of a single class (or a point prediction in regression). However, enforcing the assignment of the instance to a single class is questionable when the available information (i.e., from which a decision is made) is scarce. As well, in ensemble learning, a large conflict between the outputs of individual classifiers should lead to avoiding reaching a definitive conclusion. Therefore, in some critical systems where wrong decisions may have serious consequences, an alternative is to produce imprecise (indeterminate) predictions such as sets of plausible classes (or intervals in regression) when excessive uncertainty occurs. Following [164], when imprecise predictions are allowed to be made for a classifier, we will refer to the corresponding model as a *cautious classifier*.

2.2.1 Problem statement

A cautious classification problem, also referred to as an imprecise classification or partial classification problem, aims to assign a set-valued prediction consisting of several probable classes to input instances based on their feature values, when the uncertainty from the data or the model is too high. The main objective of cautious classification is to reduce the risk of making wrong decisions. Assume an input random variable $\mathbf{X} = \{\mathrm{X}^1, \ldots, \mathrm{X}^M\}$ of M dimensions and an output random vector Y, whose possible values (classes) are in $\Omega = \{c_1, \ldots, c_K\}$ with $K \geq 2$. As before, the input space is written as $\boldsymbol{\mathcal{X}} = \{\mathcal{X}^1, \ldots, \mathcal{X}^M\}$, whereas the output space is now $\mathcal{Y} = \mathscr{P}(\Omega)$, where $\mathscr{P}(\Omega)$ is the power set of Ω. The goal of cautious classification is to learn a function (classifier) $\boldsymbol{h} : \boldsymbol{\mathcal{X}} \rightarrow \mathscr{P}(\Omega)$ by minimizing a predefined cost (risk) measurement function on observed training data $\mathcal{D}_{train} = \{(\boldsymbol{x}_i, y_i),\ i = 1, \ldots, N_{train}\}$. Different from the output of precise classifiers, for an instance $\boldsymbol{x}$, the prediction of a cautious classifier is denoted as $\hat{Y} = \boldsymbol{h}(\boldsymbol{x})$.

2.2.2 Evaluation metrics

In contrast to traditional classifiers, cautious classifiers can generate indeterminate (set-valued) predictions. Obviously, a cautious classifier which always produces Ω as prediction is never wrong but also of no practical use. Thus, classical evaluation criteria are inadequate in this context. In light of this, given a test set $\mathcal{D}_{test} = \{(\boldsymbol{x}_i, y_i),\ i = 1, \ldots, N_{test}\}$, we mention here several evaluation criteria proposed to evaluate the quality of such set-valued predictions:

- the *determinacy* (*det*) counts the proportion of samples that are determinately classified (i.e., the classifier outputs a single class):

$$det = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \mathbb{1}(|\boldsymbol{h}(\boldsymbol{x}_i)| = 1), \tag{2.8}$$

  where $|\boldsymbol{h}(\boldsymbol{x}_i)|$ is the cardinality of the prediction, and where *Cautiousness* can be defined as $cau = 1 - Determinacy$;

- *single-set accuracy* (*ssa*) measures the proportion of correct determinate decisions:

$$ssa = \frac{1}{N_{pre}} \sum_{i=1}^{N_{test}} \mathbb{1}(\boldsymbol{h}(\boldsymbol{x}_i) = y_i), \tag{2.9}$$

  where $N_{pre} = \sum_{i=1}^{N_{test}} \mathbb{1}(|\boldsymbol{h}(\boldsymbol{x}_i)| = 1)$, indicating the number of instance determinately classified;

- *set accuracy* (*sa*) measures the proportion of indeterminate predictions containing the actual class:

$$sa = \frac{1}{N_{impre}} \sum_{i=1}^{N_{test}} \mathbb{1}(|\boldsymbol{h}(\boldsymbol{x}_i)| > 1) \cdot \mathbb{1}(y_i \in \boldsymbol{h}(\boldsymbol{x}_i)); \tag{2.10}$$

  where $N_{impre} = \sum_{i=1}^{N_{test}} \mathbb{1}(|\boldsymbol{h}(\boldsymbol{x}_i)| > 1)$, indicating the number of instance been indeterminately classified;

- *set size* (ss) gives the average size of indeterminate predictions:

$$ss = \frac{1}{N_{impre}} \sum_{i=1}^{N_{test}} |\boldsymbol{h}(\boldsymbol{x}_i)| \cdot \mathbb{1}(|\boldsymbol{h}(\boldsymbol{x}_i)| > 1); \tag{2.11}$$

- *discounted utility* (du) calculates the expected utility of making a correct (not wrong) decision, discounted by the size of the predicted set:

$$du = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} dr(|\boldsymbol{h}(\boldsymbol{x}_i)|)) \cdot \mathbb{1}(y_i \in \boldsymbol{h}(\boldsymbol{x}_i)), \tag{2.12}$$

  where $dr(\cdot)$ is the discount ratio.

Let $\hat{Y} = \boldsymbol{h}(\boldsymbol{x})$ be the set-valued prediction made by $\boldsymbol{h}$ for a given test sample $\boldsymbol{x}$, a natural choice of discount ratio is defined as

$$dr_{acc}(|\hat{Y}|) = \frac{1}{|\hat{Y}|}. \tag{2.13}$$

The discounted utility based on $dr_{acc}(\cdot)$ is called the *discounted accuracy*. However, Zaffalon et al. [227] argued that abstaining from making a decision is preferable to random guessing, i.e., the reward of an indeterminate prediction $\hat{Y}$ should be greater than $1/|\hat{Y}|$. Therefore, they proposed the following discounted utility functions:

$$dr_{u65}(|\hat{Y}|) = \frac{1.6}{|\hat{Y}|} - \frac{0.6}{|\hat{Y}|^2}, \tag{2.14}$$

and

$$dr_{u80}(|\hat{Y}|) = \frac{2.2}{|\hat{Y}|} - \frac{1.2}{|\hat{Y}|^2}. \tag{2.15}$$

The discounted utilities based on $dr_{u65}(\cdot)$ and $dr_{u80}(\cdot)$ are called the u_{65} and u_{80} scores, respectively. Finally, the F_β score can also be used as a discounting ratio:

$$dr_{F_\beta}(\hat{Y}) = \frac{\beta^2 + 1}{\beta^2 + |\hat{Y}|}. \tag{2.16}$$

The discounted utilities that use $dr_{F_1}(\cdot)$ and $dr_{F_2}(\cdot)$ are called F_1-*measure* and F_2-*measure*, respectively [48].

Table 2.2: Example of different discounted utility functions on three classes.

$\hat{Y}$	$1/\lvert\hat{Y}\rvert$			$dr_{u65}(\lvert\hat{Y}\rvert)$			$dr_{u80}(\lvert\hat{Y}\rvert)$			$dr_{F_1}(\hat{Y})$		
	c_1	c_2	c_3	c_1	c_2	c_3	c_1	c_2	c_3	c_1	c_2	c_3
$\{c_1\}$	1	0	0	1	0	0	1	0	0	1	0	0
$\{c_2\}$	0	1	0	0	1	0	0	1	0	0	1	0
$\{c_3\}$	0	0	1	0	0	1	0	0	1	0	0	1
$\{c_1, c_2\}$	0.5	0.5	0	0.65	0.65	0	0.8	0.8	0	0.667	0.667	0
$\{c_1, c_3\}$	0.5	0	0.5	0.65	0	0.65	0.8	0	0.8	0.667	0	0.667
$\{c_2, c_3\}$	0	0.5	0.5	0	0.65	0.65	0	0.8	0.8	0	0.667	0.667
$\{c_1, c_2, c_3\}$	0.333	0.333	0.333	0.467	0.467	0.467	0.6	0.6	0.6	0.5	0.5	0.5

2.2.3 Cautious classifiers based on precise probabilities

As mentioned above, based on the maximum expected utility principle and using the 0/1 loss, traditional classifiers choose the class that achieves the highest posterior probability $p(c_k|\boldsymbol{x}),\ c_k \in \Omega$ to any given test instance $\boldsymbol{x}$. However, sometimes, the conditional probabilities are very close, which means that the aleatoric uncertainty in the estimated probability distribution is very high. For example, $\boldsymbol{x}$ is an instance from a binary dataset, i.e., $\Omega = \{c_1, c_2\}$, and its estimated conditional probabilities are $p(c_1|\boldsymbol{x}) = 0.49$ and $p(c_2|\boldsymbol{x}) = 0.51$. In this case, the assignment of c_2 to $\boldsymbol{x}$ is questionable because the two classes are almost equally likely: the instance can be called ambiguous. Therefore, in this section, we will introduce some strategies to address the problem of decision-making under high aleatoric uncertainty.

Reject option

A simple and direct way to deal with a high uncertainty is to abstain from making a decision. The reject option can be divided into ambiguity rejection and novelty (or distance) rejection. The former avoids making decisions in the overlapping regions in the input space where the class of a given instance is ambiguous (high aleatoric uncertainty); and the latter abstains from making decisions in low-density regions where the instance is very dissimilar to the observed training data [94] (high epistemic uncertainty). We will detail the ambiguity rejection because it is related to our proposed cautious decision-making framework.

In [36], Chow proposed to abstain from making a decision if the maximum estimated posterior probability across all classes is less than a given threshold $t > \frac{1}{K}$:

$$\boldsymbol{h}(\boldsymbol{x}) = \begin{cases} \arg\max\limits_{c_k \in \Omega} p(c_k|\boldsymbol{x}) & \text{if } \max\limits_{c_k \in \Omega} p(c_k|\boldsymbol{x}) \geq t, \\ \text{reject} & \text{else.} \end{cases} \tag{2.17}$$

The parameter t controls the cautiousness of the reject option. The larger t is, the more cautious the model will be, i.e., more instances will not be classified. In practical applications, the value of t should be carefully selected [81].

The main drawback of the reject option is that nothing is said about possible classes. An alternative is to only reject some specific classes that are not plausible enough and return the non-rejected classes as predictions [90].

In [89], Gupta proposed a strategy called "constant risk", which fixes an acceptable risk threshold r and selects the smallest number of best classes with cumulative probability exceeding $1 - r$. Formally, for instance $\boldsymbol{x}$, the prediction is defined as:

$$\boldsymbol{h}(\boldsymbol{x}) = \arg \min_{\ell \in \{1,\ldots,K\}} |A_\ell| \text{ s.t. } \Pr(A_\ell|\boldsymbol{x}) \geq 1 - r, \tag{2.18}$$

with $A_\ell = \{c_{(1)}, \ldots, c_{(\ell)}\}$ the subset of the ℓ most probable classes, i.e. the set of classes $\{c_{(k)}, k = 1, \ldots, K\}$ is ordered by decreasing probability: $p(c_{(1)}|\boldsymbol{x}) \geq \cdots \geq p(c_{(K)}|\boldsymbol{x})$.

Nondeterministic classifier

Different from the fixed risk threshold in the reject option, Del Coz et al. [48] proposed the *nondeterministic classifier* (NDC) to directly compute the set of classes that achieves the lowest expected risk (equivalent to the highest expected utility) based on the distribution of posterior probabilities of classes for a given instance. The NDC aims to maximize the discounted utility F_β-*measure* that is defined in Eq. (2.16). In the inference phase, given the posterior probability distribution of

classes for $\boldsymbol{x}$, the set-valued prediction returned by $\boldsymbol{h}$ is

$$\boldsymbol{h}(\boldsymbol{x}) = \arg\max_{A \subseteq \Omega} \frac{1+\beta^2}{\beta^2 + |A|} \cdot \sum_{c_k \in A} p(c_k|\boldsymbol{x}). \tag{2.19}$$

As demonstrated in [48, 143], if the posterior probability distribution of classes is known, this problem can also be solved in linear time complexity as a function of the number of classes. The method consists in sorting the classes in descending order according to posterior probabilities and adding one by one classes in the set-valued prediction until the addition of the next class decreases the expected utility.

Conformal prediction

Conformal prediction is a framework that constructs cautious predictions through reliable confidence measures [208]. For a given underlying classifier that can estimate the posterior probability distribution of classes for $\boldsymbol{x}$ and a fixed probability of error ε, the objective of conformal prediction is to produce a set-valued prediction $\hat{Y} \subseteq \Omega$, $\hat{Y} \neq \emptyset$ that contains the real class y with probability at least $1-\varepsilon$, i.e., $p(y \in \hat{Y}) \geq 1-\varepsilon$.

In the setting of inductive conformal prediction, a calibration data set $\mathcal{D}_{calib} = \{(\boldsymbol{x}_i, y_i),\ i = 1, \ldots, N_{calib}\}$ is required, in which none of its samples were present in the training set of the underlying classifier. Then, a *nonconformity measure* (NCM) is computed for each sample in $\mathcal{D}_{calib}$ (e.g., $nc_i = 1 - p(y_i|\boldsymbol{x}_i),\ i = 1, \ldots, N_{calib}$) and the $1-\varepsilon$ quantile $\hat{q}$ among these nonconformity scores is calculated, which means that a proportion of samples equal to $1-\varepsilon$ have a nonconformity score no larger than $\hat{q}$. It should be noted that there are other different strategies to construct the calibration dataset, different nonconformity measures and that the $1-\varepsilon$ quantile may be adjusted according to the number of samples in the calibration dataset [181].

In the inference phase, for a given instance $\boldsymbol{x}$, the estimated posterior probabilities of classes are provided by the underlying classifier. For each class $c_k \in \Omega$, the

p-value for the hypothesis $\hat{y} = c_k$ is defined as follows:

$$p^k_{value} = p(NCM > 1 - p(c_k|\boldsymbol{x})) \approx \frac{1}{N_{calib}} \sum_{i=1}^{N_{calib}} \mathbb{1}(nc_i > 1 - p(c_k|\boldsymbol{x})). \tag{2.20}$$

If c_k reaches a p-value $p^k_{value} > \varepsilon$, the hypothesis $\hat{y} = c_k$ should not be rejected at the significance level ε, and thus c_k has to be included in the set-valued prediction. Therefore, the prediction provided by the conformal prediction classifier is

$$\boldsymbol{h}(\boldsymbol{x}) = \{c_k : p^k_{value} > \varepsilon,\ \forall c_k \in \Omega\}. \tag{2.21}$$

An equivalent representation of this decision-making strategy is

$$\boldsymbol{h}(\boldsymbol{x}) = \{c_k : 1 - p(c_k|\boldsymbol{x}) < \hat{q},\ \forall c_k \in \Omega\}. \tag{2.22}$$

Example 2.1 (Cautious classification with precise probabilities)**.** *Assume* $\Omega = \{c_1, c_2, c_3, c_4\}$ *and a given instance* $\boldsymbol{x}$ *with actual class* $y = c_1$*, its estimated posterior probabilities* $p(c_1|\boldsymbol{x}) = 0.45,\ p(c_2|\boldsymbol{x}) = 0.05,\ p(c_3|\boldsymbol{x}) = 0.4$ *and* $p(c_4|\boldsymbol{x}) = 0.1$*. The predictions taken by different models are explained as follows:*

- *reject option: suppose the threshold is set to* $t = 0.6$*, then* $\max\limits_{c_k \in \Omega} p(c_k|\boldsymbol{x}) = 0.45 < t$ *and therefore* $\hat{Y} = \{c_1, c_2, c_3, c_4\}$*;*
- *constant risk: suppose the fixed risk is* $r = 0.1$*, then we have* $\hat{Y} = \{c_1, c_3, c_4\}$ *as prediction because it is the smallest set having a sum of posterior probabilities larger than* $1 - r = 0.9$*;*
- *nondeterministic classifier: we suppose* $\beta = 1$*. The order of the classes according to the posterior probabilities is* c_1, c_3, c_4, c_2*, then we add them one by one to the prediction and calculate the expected* F_1*-measure measurement:*

$$\mathbb{E}_{F_1}[\hat{Y} = \{c_1\}] = \frac{2}{1+1} \times 0.45 = 0.45,\ \textit{continue},$$
$$\mathbb{E}_{F_1}[\hat{Y} = \{c_1, c_3\}] = \frac{2}{1+2} \times (0.45 + 0.4) = 0.567 > 0.45,\ \textit{continue},$$
$$\mathbb{E}_{F_1}[\hat{Y} = \{c_1, c_3, c_4\}] = \frac{2}{1+3} \times (0.45 + 0.4 + 0.15) = 0.475 < 0.567,\ \textit{stop},$$

since it is sure that $\mathbb{E}_{F_1}[\hat{Y} = \{c_1, c_3, c_4, c_2\}] < 0.475$. *Therefore, the set of classes maximizing the expected* F_1*-measure is* $\hat{Y} = \{c_1, c_3\}$*;*

- *conformal prediction: suppose that* $\varepsilon = 0.1$ *and* $\hat{q} = 0.65$ *(calculated with the calibration set), then the prediction should be* $\hat{Y} = \{c_1, c_3\}$ *since we have only* $1 - p(c_1|\boldsymbol{x}) < \hat{q}$ *and* $1 - p(c_3|\boldsymbol{x}) < \hat{q}$.

2.2.4 Cautious classifiers based on imprecise probabilities

Several cautious classifiers have been explicitly developed in different frameworks. Rooted in the theoretical framework of belief functions, the *evidential KNN model* regards the nearest neighbours as pieces of evidence with respect to their corresponding classes [55]. These evidences are discounted by the distances to the test data point and combined via Dempster's rule. The resulting mass functions can be used to make either precise or imprecise predictions. A similar treatment (distance-based discounting) is applied in the *evidential neural network*, which uses an evidence layer to learn prototypes in the latent space and calculates a mass function on the set of possible classes [56]. Based on this, in [197], the Hurwicz criterion is used to select the best partial assignment to produce cautious predictions. However, the number of potential partial assignments grows exponentially with the number of classes. To address this issue, the authors proposed in [197] to reduce the number of potential partial assignments by clustering the set of classes based on the confusion matrix provided by a precise CNN classifier. In [101], training data are relabeled by assigning set-valued labels to instances in overlapping or isolated regions, based on which an evidential classifier is trained. Finally, the NDC decision rule is used on the pignistic probabilities to provide set-valued predictions.

In the imprecise-probabilistic setting, the construction of cautious classifiers is mainly based on replacing the point-valued estimation of a probability distribution with a set-valued one and building partial orders among classes. The *naive credal classifier* [226], *credal networks* [39, 46], *credal sum-product networks* [134], *credal decision trees* [3] and *imprecise credal decision trees* [2] make use of credal sets

to define cautious models which provide in turn imprecise predictions. The *lower prevision KNN* [59] leverages distance-based discounting to make a lower prevision of each class. The *imprecise Gaussian discriminant analysis* adopts a robust Bayesian analysis and near-ignorance priors to imprecisely estimate the centroid of each class, leading to imprecise posterior probability estimations [8]. For multi-class cautious classifications, nested dichotomies, a special binary decomposition technique, are extended to imprecise probabilities in [225], and in [60, 167], the lower and upper probabilities of each class are calculated by solving linear programs where the binary cautious classifier outputs are interpreted as constraints.

2.3 Conclusion

In this chapter, we have provided an overview of precise and cautious classification, recalling the problem statement, and providing evaluation metrics for classifiers as well as for model comparison. Additionally, we have also presented several commonly used traditional and cautious classifiers.

In the next chapter, we will present the proposed cautious random forest model with only two classes, which cooperates with random forests, the imprecise Dirichlet model and the theory of belief functions to make cautious predictions when the uncertainty is high.

Chapter 3

Binary cautious random forests

The main objective of cautious classifiers is to identify (test) instances that are prone to errors and mitigate the risk of wrong predictions by retaining sets of classes in the presence of uncertainty. However, this comes with a cost: set-valued predictions make the classifier less informative. Therefore, it is essential to achieve a balance between risk (proportion of wrong predictions) and cautiousness (proportion and cardinality of set-valued predictions).

To address this problem, we propose a strategy within the framework of belief functions where we combine imprecise decision trees induced by the imprecise Dirichlet model to construct a cautious classifier, called cautious random forest [230, 234]. This strategy aims to achieve a better compromise between the accuracy and the cautiousness of predictions than existing aggregation methods for imprecise trees. Additionally, we introduce a cost function specifically designed for cautious classifiers to assign weights to trees in the ensemble. We stress that this chapter deals with binary classification problems. The extension to multi-class problems will be considered in Chapter 4.

This chapter is organized as follows. Section 3.1 presents imprecise trees constructed using the imprecise Dirichlet model and the existing aggregation approaches. Then, we detail our proposed imprecise tree aggregation scheme and the cost function used to learn tree weights in Section 3.2. Section 3.3 reports experiments conducted on 25 datasets and discusses the corresponding results. Finally, we conclude the chapter in Section 3.4.

3.1 Imprecise random forests: state of the art

3.1.1 Imprecise trees via the imprecise Dirichlet model

Walley's imprecise Dirichlet model (IDM) [211] is a simple yet powerful approach to propagate epistemic uncertainty, i.e., arising from a data sample of small size. Assuming that we have a set of instances, the classical inference is based on the estimated (multinomial) posterior probabilities over the classes. In a Bayesian setting,

a specific prior may be considered, for which a typical choice would be the Dirichlet distribution, being conjugate to the multinomial posterior probabilities. The IDM rather makes use of a set of Dirichlet distributions as a prior, thus resulting in a set of distributions (in the form of probability intervals) after updating [19]. These posterior probability intervals across the classes are as wide as the limited amount of data.

In this chapter, we focus on binary classification problems, i.e., we consider a training set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i),\ i = 1, \ldots, N\}$, where $y_i \in \{c_1, c_2\}$. Let $\mathbf{H}$ be a random forest of T decision trees : $\mathbf{H} = \{\boldsymbol{h}_t,\ t = 1, \ldots, T\}$ trained on $\mathcal{D}$. For a given test instance $\boldsymbol{x}$, let $n_{t1}(\boldsymbol{x})$ and $n_{t2}(\boldsymbol{x})$ denote the number of training samples of classes c_1 and c_2 falling into the same leaf as $\boldsymbol{x}$ for tree $\boldsymbol{h}_t$. The posterior probabilities estimated by each tree $\boldsymbol{h}_t$ are then noted as $p_{t1}(\boldsymbol{x}) = p(c_1|\boldsymbol{x}, \boldsymbol{h}_t)$ and $p_{t2}(\boldsymbol{x}) = p(c_2|\boldsymbol{x}, \boldsymbol{h}_t)$, respectively.

Obviously, the reliability of an individual estimate (or decision) provided by a tree strongly depends on the sample size $N_t(\boldsymbol{x}) = n_{t1}(\boldsymbol{x}) + n_{t2}(\boldsymbol{x})$ in the leaf attained by $\boldsymbol{x}$, and might therefore differ from the actual probability for some small leaves (e.g., with only one or two samples). In order to reflect epistemic uncertainty (i.e., the lack of information at the tree leaf level), the IDM can be used to produce interval-valued probability estimates. According to Eq. (1.18), for $t = 1, \ldots, T$, the IDM intervals of the estimated posterior probabilities for $c_k \in \{c_1, c_2\}$ are defined as

$$\mathcal{I}_{tk}(\boldsymbol{x}) = \left[\underline{p}_{tk}(\boldsymbol{x}),\ \overline{p}_{tk}(\boldsymbol{x})\right] = \left[\frac{n_{tk}(\boldsymbol{x})}{N_t(\boldsymbol{x}) + s},\ \frac{n_{tk}(\boldsymbol{x}) + s}{N_t(\boldsymbol{x}) + s}\right], \tag{3.1}$$

where $\underline{p}_{tk}(\boldsymbol{x})$ and $\overline{p}_{tk}(\boldsymbol{x})$ are the lower and upper bounds of $p(c_k|\boldsymbol{x}, \boldsymbol{h}_t)$. By duality, we have $\underline{p}_{t1}(\boldsymbol{x}) = 1 - \overline{p}_{t2}(\boldsymbol{x})$ and $\overline{p}_{t1}(\boldsymbol{x}) = 1 - \underline{p}_{t2}(\boldsymbol{x})$.

3.1.2 Aggregation of imprecise trees

The joint use of the IDM and decision trees is not new: it has previously been explored in two directions. First, it has been used to improve the training of single trees or tree ensembles. Credal decision trees (CDT) [3, 132] and credal random forests

(CRF) [1] use a maximum entropy principle to select split features and values from the probability intervals obtained via the IDM, thus improving robustness to data noise. To enhance the generalization performance of tree ensembles trained on small datasets, data sampling, and augmentation based on the IDM probability intervals have been proposed to train deep forests [202] and to learn weights associated with trees in the ensemble in order to further optimize their combination [203].

Second, the probability intervals given by the IDM can also be used to make cautious decisions, thereby reducing the risk of prediction error [19, 164], which is the focus of our study. A cautious decision is a set-valued one, i.e., a cautious classifier may return a set of classes instead of a single one when the uncertainty is too high. An imprecise credal decision tree (ICDT) [2] is a single tree where set-valued predictions are returned by applying the interval dominance principle [200] to the probability intervals obtained via the IDM.

In tree ensembles, applying cautious decision-making strategies becomes more complex. One approach consists in aggregating the probability intervals given by the trees, for example using conjunction, disjunction, or averaging, before making cautious decisions via computing a partial order among classes [47, 72]. Another approach consists in allowing each tree to make a cautious decision first, before pooling them. The Minimum Vote Against (MVA) is such an approach, where the classes with minimal opposition are retained [142]. It should be noted that MVA generally results in precise predictions, whereas disjunction and averaging often turn out to be inconclusive. Even worse, using conjunction very frequently results in empty predictions due to conflict.

To address the shortcomings of the aforementioned imprecise tree aggregation methods, in Chapter 3 and Chapter 4, we propose new aggregation strategies under the framework of belief functions for imprecise decision trees and cautious decision-making processes based on the aggregated information. The main objective is to achieve a better compromise between the accuracy and the cautiousness of the model. Fig. 3.1 provides an overview of the proposed cautious random forests.

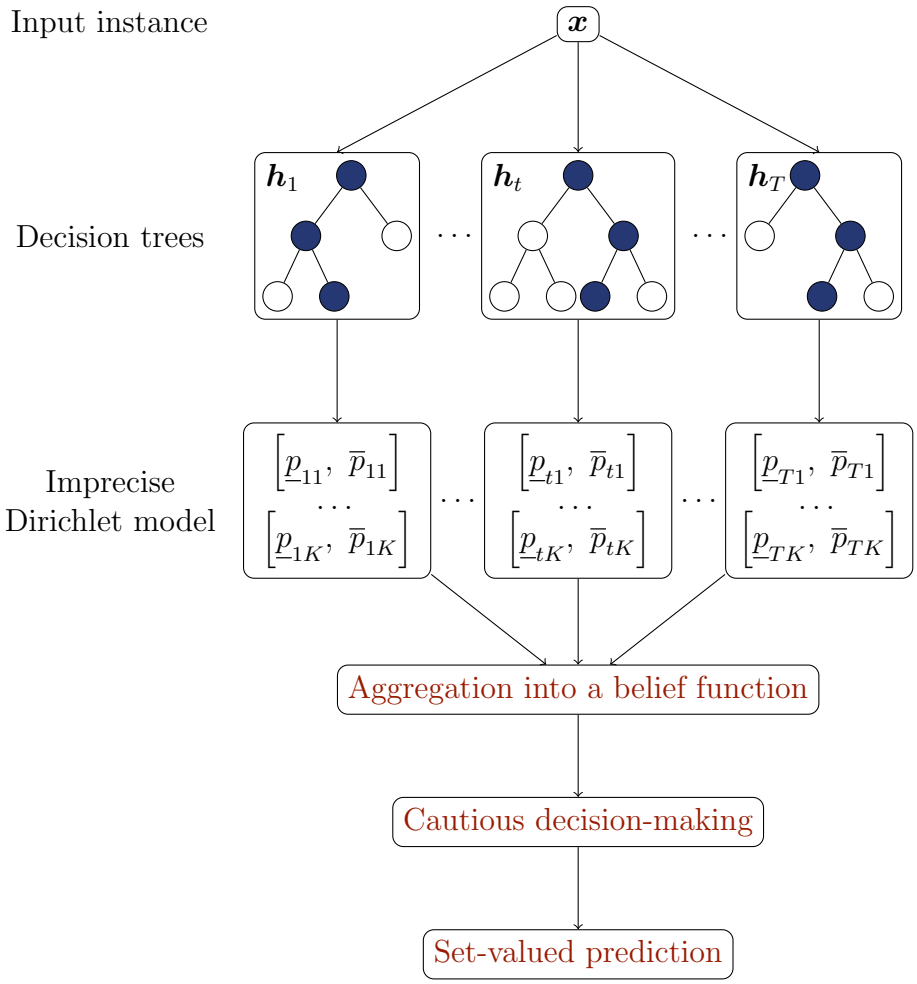


Figure 3.1: Decision-making process of cautious random forests.

3.2 New aggregation scheme

3.2.1 Imprecise tree aggregation strategy

In this section, we adopt the theoretical framework of belief functions induced by random intervals [49] to aggregate imprecise decision trees.

Let U and V be two random variables such that $U \leq V$; they may be regarded as determining a random interval $[U, V]$ and defining a belief and plausibility function on $\mathbb{R}$:

$$Bel(A) = \mathbb{P}([U, V] \subseteq A), \tag{3.2}$$

$$Pl(A) = \mathbb{P}([U, V] \cap A \neq \emptyset), \tag{3.3}$$

for any element A of the Borel sigma-algebra $\mathcal{B}(\mathbb{R})$ of the real line [49]. Let $\mathcal{I}_i = [u_i, v_i]$ with $i = 1, \ldots, N$, and let m be the mass function from the set $\mathcal{I}$ of closed real intervals on $[0, 1]$ such that $m(\mathcal{I}_i) = m_i$ with $i = 1, \ldots, N$ and $\sum_{i=1}^{N} m_i = 1$. Under this setting, the belief and plausibility functions are

$$Bel(A) = \sum_{\mathcal{I}_i \subseteq A} m_i, \quad Pl(A) = \sum_{\mathcal{I}_i \cap A \neq \emptyset} m_i, \quad \forall i = 1, \ldots, n. \tag{3.4}$$

The intervals $\mathcal{I}_i$ are called focal intervals of m [54]. This definition provides a basis for pooling pieces of information provided by the trees with respect to the class probabilities.

Bssed on the definition above, we propose to interpret the tree outputs as pieces of evidence about the actual class of a test instance in the form of closed random intervals defined on $[0, 1]$ [230, 234]. These posterior probability intervals can be aggregated into belief and plausibility degrees that can then be used in a cautious decision-making process to indicate whether one of the two classes is strictly preferable to the other or not.

More precisely, the proposed aggregation strategy consists in computing the belief and plausibility of the event "$p_1(\boldsymbol{x}) \in [0.5, 1]$". We regard each probability interval provided by a tree as a focal element on the unit interval $[0, 1]$, which provides evidence regarding the proposition that instance $\boldsymbol{x}$ belongs to class c_1. The belief and plausibility degrees across the classes are defined as

$$bel_1(\boldsymbol{x}) = Bel(\{c_1\}|\boldsymbol{x}) = Bel(p_1(\boldsymbol{x}) \in [0.5, 1]) = \sum_{t=1}^{T} w_t \cdot \mathbb{1}(\underline{p}_{t1} \geq 0.5), \tag{3.5}$$

and

$$pl_1(\boldsymbol{x}) = Pl(\{c_1\}|\boldsymbol{x}) = Pl(p_1(\boldsymbol{x}) \in]0.5, 1]) = \sum_{t=1}^{T} w_t \cdot \mathbb{1}(\overline{p}_{t1} > 0.5), \tag{3.6}$$

where w_t is the weight for tree $\boldsymbol{h}_t$, and can actually be interpreted as the degree of support $m(\mathcal{I}_{t1}(\boldsymbol{x}))$ of each interval $\mathcal{I}_{t1}(\boldsymbol{x})$ provided by the tree $\boldsymbol{h}_t$. It should be remarked that by duality, $bel_2(\boldsymbol{x}) = 1 - pl_1(\boldsymbol{x})$ and $pl_2(\boldsymbol{x}) = 1 - bel_1(\boldsymbol{x})$.

Based on the final interval $[bel_1(\boldsymbol{x}), pl_1(\boldsymbol{x})]$, the interval dominance decision rule can be applied to make a decision:

$$\hat{Y} = \begin{cases} \{c_1\}, & \text{if } bel_1(\boldsymbol{x}) \geq 0.5, \\ \{c_2\}, & \text{if } pl_1(\boldsymbol{x}) < 0.5, \\ \{c_1, c_2\}, & \text{otherwise.} \end{cases} \tag{3.7}$$

Algorithm 1 describes the inference process of our cautious random forest strategy. This aggregation strategy can be seen as a generalized voting scheme of binary classification, i.e., each tree can vote on $\{c_1\}$, $\{c_2\}$, or $\{c_1, c_2\}$.

Algorithm 1: Binary cautious random forest inference procedure.

Input: random forest $\mathbf{H}$, tree weights $w_1, \ldots, w_T$, IDM parameter s, test instance $\boldsymbol{x}$

Output: prediction $\hat{Y}$ for the given test instance

```
Ŷ ← {}
for h_t ∈ H do
    Compute I_t1(x, s) via Eq. (3.1)
Calculate bel_1(x) via Eq. (3.5)
Calculate pl_1(x) via Eq. (3.6)
if bel_1(x) ≥ 0.5 then
    Ŷ ← {c_1}
else if pl_1(x) < 0.5 then
    Ŷ ← {c_2}
else
    Ŷ ← {c_1, c_2}
Return Ŷ
```

3.2.2 Learning the tree weights

In this section, we investigate assigning weights to trees in our combination scheme. As in [202, 203], we propose to automatically learn the tree weights w_t so as to optimize the tree ensemble performances. However, to the best of our knowledge, all existing approaches are based on tree accuracy [31, 112, 120, 202, 203], and are therefore not well-suited to our imprecise classification setting, since they would

amount to give indeterminate predictions the same status as faults. We propose here to make use of a cautious criterion, which rewards both the cautiousness (associated with indeterminate predictions) and the correctness (associated with accurate determinate predictions) of the cautious classifier. For this end, we propose to learn tree weights by replacing the classically optimized accuracy measure with a utility-discounted accuracy metric [230].

Let us define

$$\boldsymbol{w} = [w_1, \ldots, w_T]^\top, \tag{3.8a}$$

$$\underline{\boldsymbol{\delta}}(\boldsymbol{x}) = [\mathbb{1}(\underline{p}_{11}(\boldsymbol{x}) \geq 0.5), \ldots, \mathbb{1}(\underline{p}_{T1}(\boldsymbol{x}) \geq 0.5)]^\top, \tag{3.8b}$$

and

$$\overline{\boldsymbol{\delta}}(\boldsymbol{x}) = [\mathbb{1}(\overline{p}_{11}(\boldsymbol{x}) > 0.5), \ldots, \mathbb{1}(\overline{p}_{T1}(\boldsymbol{x}) > 0.5)]^\top. \tag{3.8c}$$

Here, $\boldsymbol{w}$, $\underline{\boldsymbol{\delta}}(\boldsymbol{x})$ and $\overline{\boldsymbol{\delta}}(\boldsymbol{x})$ are all column vectors of T elements, with $\boldsymbol{w}$ the vector of variables to be identified. Using these notations, Equations (3.5) and (3.6) can be rewritten as

$$bel_1(\boldsymbol{x}) = \boldsymbol{w}^\top \underline{\boldsymbol{\delta}}(\boldsymbol{x}), \tag{3.9a}$$

$$pl_1(\boldsymbol{x}) = \boldsymbol{w}^\top \overline{\boldsymbol{\delta}}(\boldsymbol{x}). \tag{3.9b}$$

Note that the vectors $\underline{\boldsymbol{\delta}}(\boldsymbol{x})$ and $\overline{\boldsymbol{\delta}}(\boldsymbol{x})$ of binary values are constant once the random forest has been trained and the value of parameter s is given. Remark also that the duality property holds: $bel_2(\boldsymbol{x}) = 1 - pl_1(\boldsymbol{x})$, and $pl_2(\boldsymbol{x}) = 1 - bel_1(\boldsymbol{x})$. In the following, for the sake of simplicity, we will write $bel_{i1} = bel_1(\boldsymbol{x}_i)$, $pl_{i1} = pl_1(\boldsymbol{x}_i)$, $\underline{\boldsymbol{\delta}}_i = \underline{\boldsymbol{\delta}}(\boldsymbol{x}_i)$ and $\overline{\boldsymbol{\delta}}_i = \overline{\boldsymbol{\delta}}(\boldsymbol{x}_i)$, for any training instance $\boldsymbol{x}_i$ in the training set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i),\ i = 1, \ldots, T\}$. For the sake of simplicity, we introduce new class labels:

$$z_i = \begin{cases} 1, & \text{if } y_i = c_1, \\ 0, & \text{if } y_i = c_2. \end{cases} \tag{3.10}$$

We may naturally define an optimization criterion based on the log-loss:

$$\begin{aligned} \mathcal{L}(\boldsymbol{w}) = & -\frac{1}{N}\sum_{i=1}^{N}\{z_i \ln(bel_{i1}) + (1-z_i)\ln(bel_{i2})\} + \lambda \parallel \boldsymbol{w} \parallel_2^2 , \\ & s.t. \sum_{t=1}^{T} w_t = 1, \quad w_t \geq 0, \quad \forall t = 1, \ldots, T. \end{aligned} \tag{3.11}$$

A similar cost function was introduced in [202] as

$$\begin{aligned} \mathcal{L}(\boldsymbol{w}) = & -\frac{1}{N}\sum_{i=1}^{N}\{z_i\, bel_{i1} + (1-z_i)\, bel_{i2}\} + \lambda \parallel \boldsymbol{w} \parallel_2^2 \\ & s.t. \sum_{t=1}^{T} w_t = 1, \ \frac{1-\epsilon}{T} \leq w_t \leq \frac{1-\epsilon}{T} + \epsilon, \ \forall t = 1, \ldots, T. \end{aligned} \tag{3.12}$$

While Eq. (3.11) is akin to a cross-entropy loss, Eq. (3.12) can be regarded as a kind of hinge loss; both are convex. However, both methods tend to produce determinate predictions, since they prefer that the belief of actual class tends to one, and indeterminate predictions are penalized as errors. In a cautious setting, the cost of an indeterminate prediction should be lower than that of a determinate but erroneous one.

Therefore, we propose to optimize a cost function that considers both determinate and indeterminate predictions:

$$\begin{aligned} \mathcal{L}(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N}\{ & z_i H(0.5 - bel_{i1}) + (1-z_i)H(pl_{i1} - 0.5) \\ & -\gamma H((0.5 - bl_{i1})(pl_{i1} - 0.5))\}, \end{aligned} \tag{3.13}$$

where $H(\cdot)$ is the Heaviside function. In this cost function, determinate predictions cost nothing if they are correct, and are penalized (cost 1) if they are wrong. All indeterminate predictions cost $1-\gamma$. Optimizing this cost function amounts to looking for a compromise between making precise predictions and avoiding making mistakes. To this extent, the criterion in Eq. (3.13) can be seen as a utility-discounted accuracy measure [227]. The parameter γ can be considered as the utility of being indeterminate, which can be tuned to adjust the cautiousness of the model (the larger the

value of gamma, the more cautious the model).

Example 3.1. *Consider an instance $\boldsymbol{x}_i$ with actual label $y_i = c_1$, thus $z_i = 1$: if the model return $bel_{i1} = 0.1$ and $pl_{i1} = 0.2$, the prediction would be $\hat{Y}_i = \{c_2\}$ (wrong prediction) with a cost equal to 1, since $z_i H(0.5 - bel_{i1}) = 1$ and all other components in the cost function are zero. Conversely, if $bel_{i1} = 0.8$ and $pl_{i1} = 0.9$, the prediction would be $\{c_1\}$ (correct) and costs 0. Eventually, if $bel_{i1} = 0.4$ and $pl_{i1} = 0.6$, the indeterminate prediction $\hat{Y}_i = \{c_1, c_2\}$ would cost $1 - \gamma$, since $z_i H(0.5 - bel_{i1}) = 1$, $(1 - z_i)H(pl_{i1} - 0.5)$ and $\gamma H((0.5 - bl_{i1})(pl_{i1} - 0.5)) = \gamma$.*

It is known that the Heaviside function is neither continuous nor differentiable. Therefore, we propose to use the sigmoid function as an approximation to it:

$$H(x) \approx \sigma(x) = \frac{1}{1 + \exp(-\alpha x)}. \tag{3.14}$$

This approximation is reasonable if α is large enough. However, the sigmoid function is non-convex, and this cost function is prone to local minima. A solution to this issue consists in minimizing a surrogate (upper bound) $\mathcal{L}_{\sup}(\boldsymbol{w})$ of $\mathcal{L}(\boldsymbol{w})$ [61]. Using the inequality $x \leq -\ln(1 - x),\ \forall x < 1$, the equality $\sigma(-x) = 1 - \sigma(x)$ and $\sigma(x) < 1,\ \forall x \in \mathbb{R}$, we have

$$\sigma(0.5 - bel_{i1}) \leq -\ln(\sigma(bel_{i1} - 0.5)), \tag{3.15a}$$

$$\sigma(pl_{i1} - 0.5) \leq -\ln(1 - \sigma(pl_{i1} - 0.5)), \tag{3.15b}$$

and

$$-\sigma\left((0.5 - bl_{i1})(pl_{i1} - 0.5)\right) \leq -\ln\left(1 - \sigma\left((bel_{i1} - 0.5)(pl_{i1} - 0.5)\right)\right) - 1. \tag{3.15c}$$

We remark that a regularization term should be taken into account in the cost function, so as to avoid over-fitting. We finally obtain the following regularized

upper bound cost function:

$$\mathcal{L}_{\sup}(w) = -\frac{1}{N}\sum_{i=1}^{N}\Big\{z_i \ln\big(\sigma(\boldsymbol{w}^\top \underline{\boldsymbol{\delta}}_i - 0.5)\big) + (1-z_i)\ln\big(1-\sigma(\boldsymbol{w}^\top \overline{\boldsymbol{\delta}}_i - 0.5)\big) + \gamma \ln\big(1-\sigma((\boldsymbol{w}^\top \underline{\boldsymbol{\delta}}_i - 0.5)(\boldsymbol{w}^\top \overline{\boldsymbol{\delta}}_i - 0.5))\big)\Big\} + \frac{1}{2}\lambda \parallel \boldsymbol{w} \parallel_2^2$$
$$s.t. \sum_{t=1}^{T} w_t = 1, \quad w_t \geq 0, \ \forall t = 1, \ldots, T. \tag{3.16}$$

In Eq. (3.16), the first and the second terms within the summation correspond to the penalty incurred for not assigning an instance to the right class. However, in case of an indeterminate decision, this penalty would be partially compensated (depending on the γ value) by the third term in the summation. The last term out of the summation is a regularization term to avoid over-fitting. The gradient, the Hessian matrix, and the proof of the convexity of Eq. (3.16) are given in Appendices A and B. It can therefore be easily minimized using any convex optimization solver.

3.3 Experiments and results

In this section, we detail the experiments conducted on 25 public datasets from the UCI repository [15] to show the interest of our approach. All datasets are collected for binary classification problems and cover a large range of sample sizes and number of features (see Table 3.1). Experiments are reported in two steps:

- in Section 3.3.1, the different tree aggregation strategies providing cautious predictions are compared on normal data, noisy data, and small training data;
- in Section 3.3.2, it illustrates the advantage of our proposed strategy for learning tree weights compared to other weight assignment methods and studies the influence of the hyper-parameter tuning the compromise between the risk and the informativeness of the model.

We adopt evaluation metrics defined in Section 2.2.2, including cautiousness, single-set accuracy, u_{65} score, and u_{80} score to evaluate different cautious classifiers.

Table 3.1: Datasets used in the experiments, with abbreviation ABB, numbers of instances (N) and of features (nominal/numerical).

Dataset	**ABB**	**N**	**Feat**	**Nom**	**Num**
adult	ADT	45222	11	0	11
banknote	BKT	1372	4	0	4
biodeg	BID	1053	41	0	41
breast-cancer	BRC	568	30	0	30
cardiac	CAD	889	12	0	12
compas	COP	2652	6	0	6
credit	CRD	690	15	9	6
diabetes	DIB	768	8	0	8
german	GER	1000	24	0	24
heart	HRT	303	13	0	13
heloc	HLC	10459	23	0	23
ionosphere	INS	351	34	0	34
liver	LIV	345	6	0	6
magic	MGC	2300	57	0	57
mammographic	MMG	830	5	0	5
occupancy	OCP	2665	6	1	5
phishing	PHS	11054	30	0	30
pima	PMA	768	8	0	8
post-operative	POP	88	8	7	1
ringnorm	RNO	7400	20	0	20
seismic	SSC	2584	18	4	14
sonar	SNR	208	60	0	60
spam	SPM	4594	57	0	57
vote	VTE	435	16	16	0
wine	WNE	1599	11	0	11

The set accuracy is not considered here because it always equals to one for binary cautious classification.

In order to compare multiple models over multiple datasets, we followed the recommendation of [51]. First, the Friedman test [79], which is a non-parametric test that scores the algorithms independently for each data set, is performed to determine whether the classifiers are significantly different or not. The top-performing algorithm receives a rank of 1, the second-best receives a rank of 2, and so on. If the null hypothesis (all algorithms are equivalent) is rejected, a Nemenyi test [147] can be applied in a second step to identify the significant differences between models.

3.3.1 Comparison of tree aggregation strategies

Compared models

In this first phase of experiments, we benchmark different tree aggregation strategies in random forests, all tree weights being considered to be equal. The methods compared are:

- AVE: AVErage, where, following [146] and [72], we average the lower and upper probabilities provided by the trees at hand, i.e., $bel_1(\boldsymbol{x}) = \frac{1}{T}\sum_{t=1}^{T} \underline{p}_{t1}(\boldsymbol{x})$ and $pl_1(\boldsymbol{x}) = \frac{1}{T}\sum_{t=1}^{T} \overline{p}_{t1}(\boldsymbol{x})$, before applying interval dominance defined in Eq. (1.21);
- MV: Majority Voting is adapted to our imprecise classification setting, by applying interval dominance to each tree, and considering indeterminate predictions $\{c_1, c_2\}$ as a possible outcome when counting the votes [72];
- RO: in Reject Option, we first estimate the probability $p_1(\boldsymbol{x})$ of class c_1 as the number of trees providing a probability $p_{t1}(\boldsymbol{x}) \geq 0.5$, and we predict class c_1 whenever $p_1(\boldsymbol{x}) > 0.5 + \theta$, class c_2 whenever $p_1(\boldsymbol{x}) < 0.5 - \theta$, and $\{c_1, c_2\}$ otherwise (with θ a hyper-parameter to be set) [36];
- MVA: Minimum Vote Against counts the number of classifiers that predict a class as dominated (vote against), the final non-dominated set of classes being made of the classes with the lowest amount of votes against [142];
- CRF: our proposed cautious random forest strategy, where we first pool the trees by computing the belief and plausibility degrees according to Eq. (3.5) and Eq. (3.6) (with equal tree weights), before applying interval dominance.

Experimental setting and procedure

The experiments were realized using the random forest classifier from the scikit-learn [157] python library. Each tree in the ensemble is trained to its full depth, i.e., the minimum number of training samples allowed in a leaf is one. Since the library made

it possible to handle numeric features only, all categorical features were converted by one-hot encoding. The forest consists of $T = 100$ trees.

We implemented the following protocol to compare the aggregation strategies. For each dataset, for our method (CRF) we selected by cross-validation the value of the IDM parameter s which maximizes u_{65} score; we used the same s for the MV strategy. For AVE, the value was fixed to $s = 1$, following the recommendations in [211]. For RO, the threshold was set to $\theta = 0.05$ for all datasets, which was found to give good results across all datasets.

Tests have been carried out in three directions. First, we applied our protocol to the standard UCI datasets. In the second step, we introduced noise in the training data by flipping a fixed proportion of labels drawn at random. In the experiments, we considered various levels of label noise (0%, 5%, 10%, 15%, 20%). Average cautiousness, single-set accuracy, u_{65}, and u_{80} scores were computed by averaging the measures made on ten repetitions of 10-fold cross-validation according to the selected parameters. Last, we studied the effect of the training set size on the results. For different sizes of the training set ($N \in \{20, 30, 50, 100, 150, 200\}$), each metric was computed by averaging 100 independent repetitions according to the selected parameters. The training samples were randomly selected from the whole dataset and the remaining ones were used as the test set.

Results and discussion

First, we discuss the results obtained on standard datasets, which are reported in Tables 3.2 to 3.5. As it can be seen from Table 3.2, CRF appears to be the most cautious of all models and yields very similar results to AVE. MVA is the least cautious on all datasets, reaching a level of cautiousness of less than 1%.

All cautious classifiers outperform the precise random forest (RF) — often by a significant amount — in terms of single-set accuracy, thanks to their ability to classify some difficult samples as indeterminate. However, according to the results in Table 3.3, CRF is able to achieve the highest single-set accuracy, which indicates that it is the most reliable model when determinate predictions are made. Tables 3.4

Table 3.2: Cautiousness (%) of different aggregation strategies on each dataset (without label noise) where the best result is printed in bold.

Data	**AVE**	**MV**	**RO**	**MVA**	**CRF**
ADT	13.43	15.79	4.82	0.26	**18.66**
BKT	**0.40**	0.03	0.37	0.02	0.26
BID	8.69	0.93	5.64	0.25	**10.32**
BRC	**2.36**	0.05	1.88	0.07	1.51
CAD	3.98	**8.73**	1.62	0.18	8.05
COP	35.58	31.03	8.91	0.72	**37.40**
CRD	10.67	8.15	4.48	0.17	**13.32**
DIB	18.30	2.59	9.74	0.50	**20.88**
GER	27.32	15.47	11.82	0.56	**33.13**
HRT	16.31	4.98	7.60	0.56	**20.07**
HLC	19.61	1.41	11.45	0.46	**21.90**
INS	2.45	0.23	1.77	0.00	**3.42**
LIV	**27.36**	0.46	13.50	0.58	17.56
MGC	**3.98**	0.28	2.39	0.13	2.95
MMG	14.65	**27.28**	3.40	0.24	25.03
OCP	0.58	0.67	0.30	0.02	**0.94**
PHS	**6.18**	1.86	2.42	0.14	5.63
PMA	18.59	2.40	10.29	0.48	**21.08**
POP	29.53	**39.14**	6.25	0.35	29.04
RNO	4.99	0.32	4.70	0.22	**5.41**
SSC	**1.98**	0.06	1.13	0.01	1.29
SNR	**18.48**	0.77	14.47	0.67	11.64
SPM	**3.71**	0.20	2.48	0.05	2.61
VTE	**3.95**	1.20	1.42	0.09	3.70
WNE	**14.42**	0.61	8.30	0.37	10.01
Average	12.30	6.59	5.65	0.28	**13.03**
#Highest	10	3	0	0	**12**
#Lowest	0	2	0	**23**	0

Table 3.3: Single-set accuracy (%) of different aggregation strategies on each dataset (without label noise) where the best result is printed in bold.

Data	**AVE**	**MV**	**RO**	**MVA**	**RF**	**CRF**
ADT	87.79	88.93	84.63	84.18	83.60	**89.73**
BKT	99.42	99.29	99.46	99.36	**99.63**	99.37
BID	90.10	87.36	89.19	87.02	87.04	**90.94**
BRC	**97.03**	95.90	96.87	96.04	96.02	96.93
CAD	78.98	79.75	78.42	77.90	77.82	**79.94**
COP	64.57	61.77	60.61	59.66	60.11	**64.64**
CRD	91.18	90.01	89.22	87.18	87.51	**92.15**
DIB	81.24	77.25	79.14	76.77	76.36	**82.26**
GER	83.51	79.47	79.63	76.26	77.28	**84.78**
HRT	87.30	84.11	84.67	82.75	82.66	**88.47**
HLC	74.59	71.01	72.75	70.87	70.75	**75.15**
INS	94.47	93.56	94.27	93.53	93.34	**94.81**
LIV	**78.65**	73.74	76.37	74.22	73.68	77.05
MGC	**95.92**	94.48	95.57	93.35	94.62	95.89
MMG	84.94	**88.00**	80.24	81.30	79.89	87.49
OCP	98.78	98.88	98.61	98.62	**99.09**	98.97
PHS	**96.39**	94.59	95.19	94.28	94.84	96.11
PMA	81.02	76.94	79.06	76.48	76.24	**81.79**
POP	67.24	62.98	65.14	65.00	65.05	**67.91**
RNO	**95.16**	93.13	95.09	93.27	93.72	95.03
SSC	**93.87**	93.25	93.64	93.25	93.76	93.73
SNR	**89.00**	83.27	88.26	83.40	84.69	87.58
SPM	**95.82**	94.49	95.43	94.42	95.00	95.37
VTE	97.53	96.40	96.40	96.27	95.86	**97.68**
WNE	**86.05**	82.32	84.88	82.25	82.43	85.17
Average	87.62	85.64	86.11	84.71	84.84	**87.96**
#Highest	9	1	0	0	2	**13**
#Lowest	0	6	1	9	**10**	0

Table 3.4: u_{65} score (%) of different aggregation strategies on each dataset (without label noise) where the best result is printed in bold.

Data	**AVE**	**MV**	**RO**	**MVA**	**CRF**
ADT	84.72	**85.13**	83.67	84.13	85.09
BKT	99.29	99.28	99.33	**99.35**	99.29
BID	87.93	87.14	87.82	86.96	**88.26**
BRC	96.27	95.88	96.26	96.02	**96.44**
CAD	78.38	78.43	78.18	77.88	**78.72**
COP	64.70	62.72	61.00	59.70	**64.74**
CRD	88.43	88.00	88.16	87.14	**88.57**
DIB	78.23	76.91	77.73	76.71	**78.62**
GER	**78.38**	77.18	77.86	76.19	78.14
HRT	83.66	83.16	83.21	82.64	**83.75**
HLC	72.69	70.93	71.85	70.85	**72.92**
INS	93.73	93.48	93.74	93.53	**93.77**
LIV	**74.89**	73.70	74.73	74.14	74.87
MGC	94.68	94.40	94.83	93.31	**94.98**
MMG	**82.00**	81.72	79.71	81.27	81.86
OCP	98.58	**98.65**	98.51	98.62	**98.65**
PHS	94.44	94.03	**94.46**	94.23	94.34
PMA	78.00	76.66	77.60	76.42	**78.22**
POP	65.86	63.02	65.11	65.02	**66.17**
RNO	93.65	93.04	**93.67**	93.21	93.40
SSC	93.30	93.24	93.31	93.25	**93.36**
SNR	84.53	83.12	84.86	83.28	**84.90**
SPM	**94.67**	94.43	**94.67**	94.41	94.57
VTE	96.25	96.02	95.96	96.24	**96.48**
WNE	83.01	82.21	**83.22**	82.19	83.15
Average	85.61	84.90	85.18	84.67	**85.73**
#Highest	4	2	4	1	**16**
#Lowest	0	9	4	**12**	0

Table 3.5: u_{80} score (%) of different aggregation strategies on each dataset (without label noise) where the best result is printed in bold.

Data	**AVE**	**MV**	**RO**	**MVA**	**CRF**
DT	86.74	87.50	84.40	84.17	**87.89**
BKT	99.35	99.29	**99.38**	99.35	99.32
BID	89.23	87.28	88.66	87.00	**89.81**
BRC	96.62	95.89	96.54	96.03	**96.67**
CAD	78.98	79.74	78.43	77.91	**79.93**
COP	70.04	67.38	62.34	59.81	**70.35**
CRD	90.03	89.23	88.83	87.17	**90.57**
DIB	80.97	77.30	79.19	76.78	**81.76**
GER	82.48	79.50	79.64	76.28	**83.11**
HRT	86.11	83.91	84.35	82.73	**86.76**
HLC	75.63	71.15	73.57	70.92	**76.21**
INS	94.10	93.52	94.01	93.53	**94.28**
LIV	**78.99**	73.77	76.75	74.23	77.51
MGC	95.28	94.44	95.19	93.33	**95.42**
MMG	84.20	**85.81**	80.22	81.30	85.61
OCP	98.67	98.75	98.56	98.62	**98.79**
PHS	**95.37**	94.31	94.83	94.26	95.19
PMA	80.79	77.02	79.14	76.49	**81.38**
POP	70.29	68.89	66.04	65.07	**70.52**
RNO	**94.40**	93.09	94.38	93.24	94.21
SSC	**93.60**	93.25	93.48	93.25	93.55
SNR	**87.30**	83.23	87.03	83.38	86.64
SPM	**95.23**	94.46	95.05	94.42	94.96
VTE	96.84	96.20	96.18	96.25	**97.03**
WNE	**85.17**	82.30	84.47	82.24	84.65
Average	87.46	85.89	86.03	84.71	**87.68**
#Highest	7	1	1	0	**16**
#Lowest	0	7	3	**16**	0

Table 3.6: Friedman statistic and p-value (left), Nemenyi p-values for pairwise model comparison on noise-free data (right). The best result for Friedman rank is printed in bold.

(a) Friedman rank and test

	AVE	**MV**	**RO**	**MVA**	**CRF**	**p-value**
cau	1.96	3.08	3.48	4.48	**1.64**	5.21×10^{-8}
ssa	2.08	3.08	3.84	4.44	**1.56**	7.99×10^{-9}
u65	2.36	2.96	3.80	4.12	**1.76**	1.74×10^{-7}
u80	2.16	3.28	3.68	4.40	**1.48**	9.01×10^{-9}

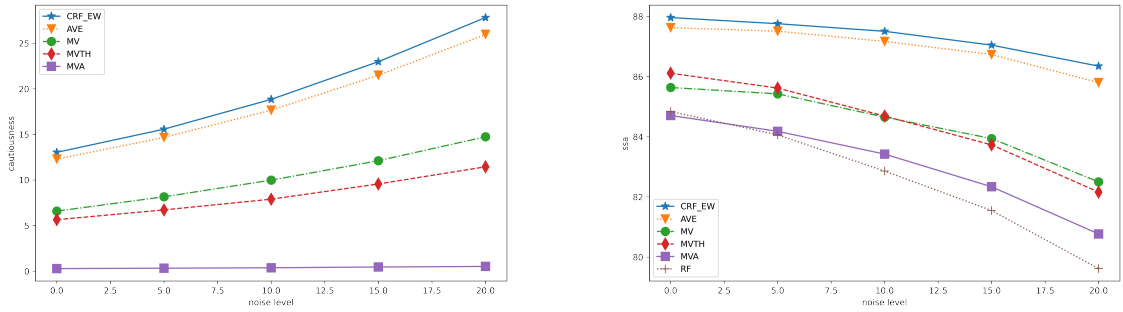
(b) Nemenyi test

CRF	**vs. AVE**	**vs. MV**	**vs. RO**	**vs. MVA**
cau	0.90	0.001	0.007	0.001
ssa	0.90	0.001	0.005	0.001
u65	0.49	0.001	0.020	0.001
u80	0.90	0.001	0.002	0.001

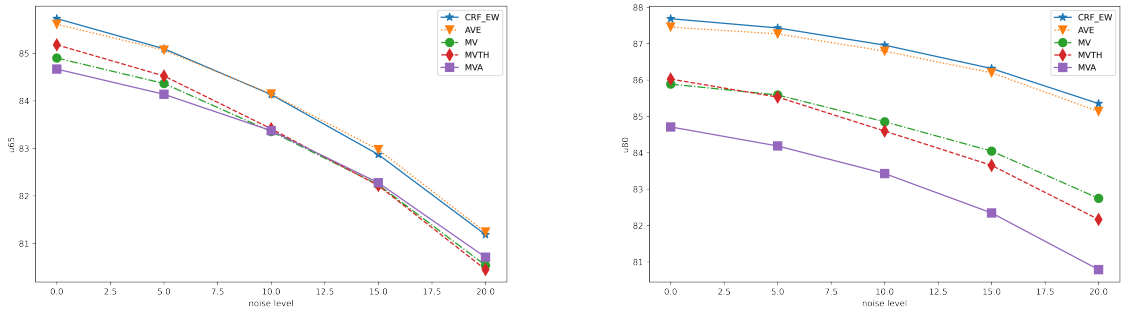
and 3.5 show that in terms of utility-discounted accuracy (both u_{65} and u_{80}), which measures a trade-off between cautiousness and single-set accuracy, CRF outperforms all other baselines in the great majority of cases. This is confirmed by the Friedman test and Nemenyi test in Table 3.6(a) and 3.6(b). CRF outperforms significantly all other models (with a p-value less than 0.05) except AVE, for which the differences are not significant. This first round of experiments thus shows that our combination and decision strategy based on the theory of belief functions provides an interesting way of making cautious and reliable decisions.

We now move on to the second part of this first phase of experiments, designed to study the robustness of CRF against noisy data. The ability to perform well in the presence of noisy data is an important feature of a good classifier. In our case, the classifier is expected to become more cautious when facing low-quality data. In these experiments, we investigate the impact of label noise on model performance, by introducing a given percentage of erroneous labels in the training samples. Figures 3.2(a) to 3.2(d) display the behavior of the four evaluation metrics for the compared models, averaged over all datasets, as a function of label noise.

As expected, the cautiousness of all models increases as the level of noise in-

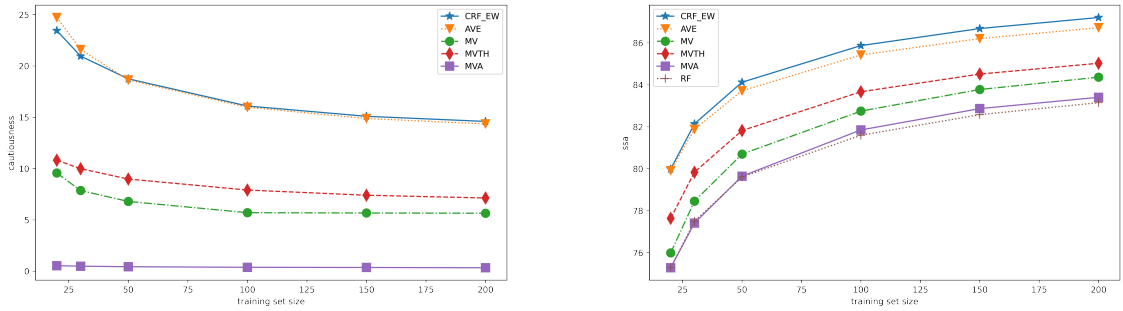


(a) cautiousness (b) single-set accuracy

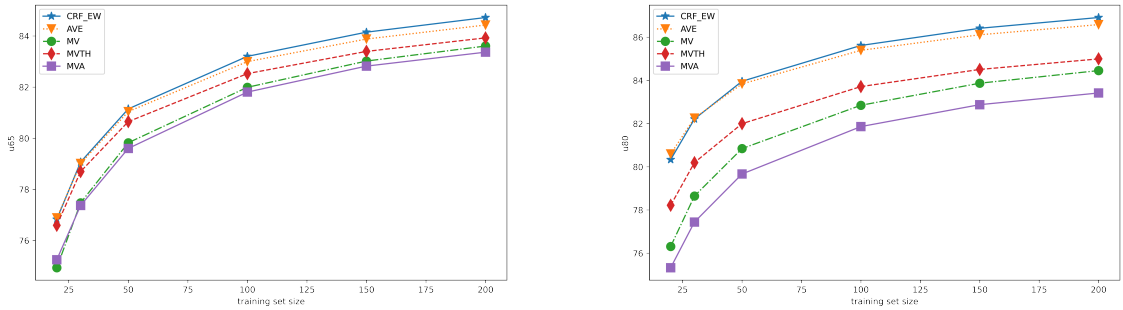


(c) u_{65} score (d) u_{80} score

Figure 3.2: Average cautiousness, single-set accuracy, u_{65}, and u_{80} scores computed over all datasets, as a function of label noise.



(a) cautiousness (b) single-set accuracy



(c) u_{65} score (d) u_{80} score

Figure 3.3: Average cautiousness, single-set accuracy, u_{65} and u_{80} scores computed over all datasets, as a function of training set size.

creases. However, the effect is strongest for CRF and AVE: with 20% of noisy labels, cautiousness increases by about 15%, which indicates that CRF and AVE perform better in the presence of noise compared to MV, RO, and MVA. For MV and RO, cautiousness only increased by about 5%. Even worse, MVA seems to be insensitive to noise and always maintains cautiousness around 0.5%. It should be noted that CRF is even more cautious than AVE for very high levels of label noise. From 0% to 20%, the single-set accuracy of the traditional random forest drops by 5%, and by 3% for MV, RO, and MVA, whereas the results of CRF and AVE suffer a decrease of about 1% only. Note also that CRF always keeps a slight advantage over AVE. The same can also be noticed with the u_{65} and u_{80} metrics.

These results show that CRF performs well in cases of high aleatoric uncertainty in the data. Another crucial type of uncertainty is epistemic uncertainty, which is mainly caused by a lack of training data [99]. In general, a cautious classifier facing a high epistemic uncertainty should maintain a high degree of cautiousness to reduce the risk of making incorrect decisions. As the training set size increases (i.e., more data are collected), cautiousness should decrease and the uncertainty in the outputs should mainly be of an aleatoric nature. Thus we carried out some experiments to study this point. Figure 3.3(a) presents the average cautiousness computed over all datasets for the four compared methods when varying the size of the training set. It can be seen that all models tend to be more cautious as the size of the training set gets smaller, but CRF and AVE are far more sensitive to this parameter. This makes it possible for CRF and AVE to reach a higher single-set accuracy, and therefore higher u_{65} and u_{80} scores, regardless of the amount of training data, as shown in Figures 3.3(b) to 3.3(d).

3.3.2 Comparison of weight assignment strategies

Compared models

The second phase of experiments evaluates the interest of learning tree weights by optimizing the proposed cost function (3.16). For this purpose, the aggregation

strategy used for all models is the one defined by Eq. (3.7) and Eq. (1.21). Different weighting strategies are compared to each other:

- EW: the Equal Weight strategy assigns a weight $1/T$ to each tree;
- OOBACC: the Out-Of-Bag ACCuracy approach assigns a weight to each tree according to its accuracy, estimated using out-of-bag samples;
- OOBU65: this approach is similar to OOBACC, except that the performance of each tree is determined using the u_{65} criterion;
- IRF: tree weights are learned using the cost function proposed by [203], which corresponds to Eq. (3.12);
- AW: our proposed tree weight allocation strategy, where weights are obtained so as to minimize Equation (3.16).

Experimental setting and procedure

In order to evaluate the various tree-weighting strategies, we used the following procedure. For all weight assignment strategies, we used the same values of s as in the first phase of experiments.

For CRF with AW, and for each dataset, we selected the value for the parameter γ in Eq. (3.16) that maximizes the u_{65} score via cross-validation and fix the corresponding parameter λ to 10 for all datasets. Regarding the IRF approach, it is proposed to avoid over-fitting by grouping the trees, and computing a weight for each group instead of each tree. We followed this procedure and performed grid search cross-validation to select the best combination of the two hyper-parameters $\epsilon \in \{0.25, 0.5, 0.75\}$ and $G \in \{5, 10, 20, 25, 100\}$; however, we maximized the u_{65} score instead of accuracy, since we compare here cautious classification strategies. The parameter λ in Eq. (3.12) was set to 0.5 for all datasets in the experiments.

Cautiousness, single-set accuracy, u_{65} and u_{80} were evaluated by averaging the results obtained on 10 repetitions for each of the weight assignment methods compared after the parameters were selected (in each repetition) using 10-fold cross-validation.

Results and discussion

In this subsection, the results obtained for various tree weight assignments in a cautious random forest are presented and analyzed. The influence of the parameter γ in the learning process is also discussed.

Tables 3.7 to 3.10 report the performances of CRF with different weight assignment methods. Thanks to the introduction of a specific utility value for indeterminate predictions, CRF with automatically-learned weights (AW) always makes it possible to reach a good compromise between single-set accuracy and cautiousness: for all datasets, it yields the highest cautiousness degree, and at the same time the highest single-set accuracy, u_{65} and u_{80} values. The differences are significant (all p-values being less than 0.05), which is confirmed by the Friedman and Nemenyi tests reported in Tables 3.11(a) and 3.11(b).

It is worth noting that the three weight assignment methods EW, OOBACC, and OOBU65 achieve almost identical performances. This may be due to the fact that the differences between the trees are not significant enough to result in different decisions being made after normalization, especially since a voting mechanism is used. By contrast, the proposed weight assignment strategy better fits the decision trees in the forest, which results in higher accuracy scores. Remember that as illustrated by [203], the cost function in IRF is advantageous for precise classification problems: in an imprecise classification setting, considering only accuracy leads to designing classifiers that are not cautious enough, hence resulting in lower single-set accuracy, u_{65} and u_{80} values.

Table 3.7: Cautiousness (%) of different weight assignment methods on each dataset (without label noise) where the best result is printed in bold.

Data	**EW**	**OOBACC**	**OOBU65**	**IRF**	**AW**
ADT	18.66	18.84	18.90	17.75	**20.27**
BKT	0.26	0.18	0.17	0.14	**0.27**
BID	10.32	10.13	10.20	9.97	**10.54**
BRC	1.51	1.65	1.63	1.69	**1.83**
CAD	8.05	8.03	8.02	7.94	**10.00**
COP	37.40	37.07	37.18	35.61	**38.77**
CRD	**13.32**	12.91	13.01	12.78	13.17
DIB	20.88	20.37	20.50	19.20	**20.90**
GER	33.13	33.20	33.28	32.95	**33.49**
HRT	**20.07**	20.00	**20.07**	18.65	19.84
HLC	21.90	21.72	21.79	21.33	**22.06**
INS	3.42	3.65	3.68	3.33	**3.73**
LIV	17.56	16.93	17.10	16.55	**18.76**
MGC	2.95	3.01	3.01	2.88	**3.17**
MMG	25.03	25.21	25.21	22.93	**25.25**
OCP	0.94	0.95	0.95	0.95	**1.26**
PHS	**5.63**	5.37	5.38	5.37	5.63
PMA	21.08	21.08	21.12	20.30	**21.84**
POP	29.04	28.25	28.37	23.36	**30.03**
RNO	5.41	5.51	5.51	5.43	**5.69**
SSC	1.29	1.17	1.15	1.26	**1.37**
SNR	11.64	11.70	11.65	10.67	**13.11**
SPM	2.61	2.53	2.51	2.41	**2.78**
VTE	3.70	3.54	3.54	3.13	**3.95**
WNE	**10.01**	9.49	9.55	9.58	9.85
Average	13.03	12.90	12.94	12.25	**13.50**
#Highest	4	0	1	0	**22**
#Lowest	3	2	1	**20**	0

Table 3.8: Single-set accuracy (%) of different weight assignment methods on each dataset (without label noise) where the best result is printed in bold.

Data	**EW**	**OOBACC**	**OOBU65**	**IRF**	**AW**
ADT	89.73	90.28	90.27	89.60	**91.69**
BKT	**99.37**	99.35	99.34	99.21	99.34
BID	90.94	90.88	90.89	90.50	**92.08**
BRC	96.93	97.20	97.16	96.91	**98.34**
CAD	79.94	80.03	80.05	79.68	**81.95**
COP	64.64	65.01	64.93	64.46	**66.67**
CRD	92.16	92.19	92.18	91.89	**93.45**
DIB	82.26	81.74	81.89	81.17	**83.08**
GER	84.78	84.81	84.83	84.37	**86.04**
HRT	88.47	88.64	88.65	87.80	**89.64**
HLC	75.15	75.14	75.21	74.77	**76.27**
INS	94.81	94.92	94.93	94.71	**96.00**
LIV	77.05	76.90	76.89	76.38	**78.71**
MGC	95.89	94.74	94.74	94.39	**95.90**
MMG	87.49	87.71	87.72	87.30	**88.84**
OCP	98.97	99.18	99.16	98.95	**99.31**
PHS	96.11	96.46	96.44	96.17	**97.67**
PMA	81.79	81.97	81.97	81.58	**83.35**
POP	67.91	65.84	66.23	65.39	**68.30**
RNO	95.03	95.41	95.46	95.14	**96.71**
SSC	93.73	94.05	94.07	93.77	**95.23**
SNR	87.58	86.95	86.82	86.26	**88.61**
SPM	95.37	95.53	95.56	95.21	**96.75**
VTE	97.68	97.76	97.75	97.37	**99.00**
WNE	85.17	85.02	85.05	84.75	**86.26**
Average	87.96	87.91	87.93	87.51	**89.17**
#Highest	1	0	0	0	**24**
#Lowest	3	0	0	**22**	0

Table 3.9: u_{65} score (%) of different weight assignment methods on each dataset (without label noise) where the best result is printed in bold.

Data	**EW**	**OOBACC**	**OOBU65**	**IRF**	**AW**
ADT	85.09	85.50	85.48	85.22	**86.26**
BKT	**99.29**	**99.29**	99.28	99.16	99.25
BID	88.26	88.25	88.25	87.94	**89.22**
BRC	96.44	96.67	96.63	96.37	**97.73**
CAD	78.72	78.81	78.84	78.51	**80.26**
COP	64.74	64.96	64.91	64.62	**65.99**
CRD	88.57	88.68	88.65	88.45	**89.71**
DIB	78.62	78.32	78.40	78.02	**79.28**
GER	78.14	78.20	78.20	77.96	**78.96**
HRT	83.75	83.92	83.89	83.54	**84.77**
HLC	72.92	72.92	72.97	72.67	**73.76**
INS	93.77	93.82	93.82	93.71	**94.82**
LIV	74.87	74.79	74.75	74.42	**76.08**
MGC	**94.98**	93.85	93.85	93.54	94.92
MMG	81.86	81.95	81.96	82.16	**82.79**
OCP	98.65	98.92	98.89	98.63	**99.07**
PHS	94.34	94.76	94.74	94.49	**95.82**
PMA	78.22	78.36	78.36	78.20	**79.31**
POP	66.17	65.62	65.81	65.44	**67.26**
RNO	93.40	93.74	93.78	93.51	**94.91**
SSC	93.36	93.70	93.73	93.40	**94.81**
SNR	84.90	84.35	84.27	84.07	**85.51**
SPM	94.57	94.76	94.79	94.47	**95.87**
VTE	96.48	96.60	96.59	96.36	**97.66**
WNE	83.15	83.11	83.13	82.86	**84.16**
Average	85.73	85.75	85.76	85.51	**86.73**
#Highest	2	1	0	0	**23**
#Lowest	5	0	0	**20**	0

Table 3.10: u_{80} score (%) of different weight assignment methods on each dataset (without label noise) where the best result is printed in bold.

Data	**EW**	**OOBACC**	**OOBU65**	**IRF**	**AW**
ADT	87.89	88.32	88.32	87.88	**89.30**
BKT	**99.32**	99.31	99.31	99.18	99.29
BID	89.81	89.77	89.78	89.44	**90.80**
BRC	96.67	96.92	96.88	96.62	**98.00**
CAD	79.93	80.02	80.04	79.70	**81.76**
COP	70.35	70.52	70.49	69.96	**71.81**
CRD	90.57	90.62	90.60	90.37	**91.68**
DIB	81.76	81.37	81.48	80.90	**82.42**
GER	83.11	83.18	83.19	82.90	**83.98**
HRT	86.76	86.92	86.90	86.33	**87.75**
HLC	76.21	76.18	76.23	75.87	**77.07**
INS	94.28	94.36	94.37	94.21	**95.39**
LIV	77.51	77.33	77.32	76.90	**78.89**
MGC	**95.42**	94.30	94.30	93.98	95.40
MMG	85.61	85.73	85.74	85.60	**86.58**
OCP	98.79	99.04	99.02	98.77	**99.07**
PHS	95.19	95.56	95.54	95.29	**96.66**
PMA	81.38	81.52	81.53	81.24	**82.58**
POP	70.52	69.86	70.06	68.94	**71.76**
RNO	94.21	94.56	94.60	94.32	**95.76**
SSC	93.55	93.88	93.90	93.59	**95.02**
SNR	86.64	86.11	86.02	85.67	**87.48**
SPM	94.96	95.13	95.17	94.84	**96.29**
VTE	97.03	97.13	97.12	96.83	**98.25**
WNE	84.65	84.53	84.56	84.29	**85.64**
Average	87.68	87.69	87.70	87.34	**88.75**
#Highest	2	0	0	0	**23**
#Lowest	3	0	0	**22**	0

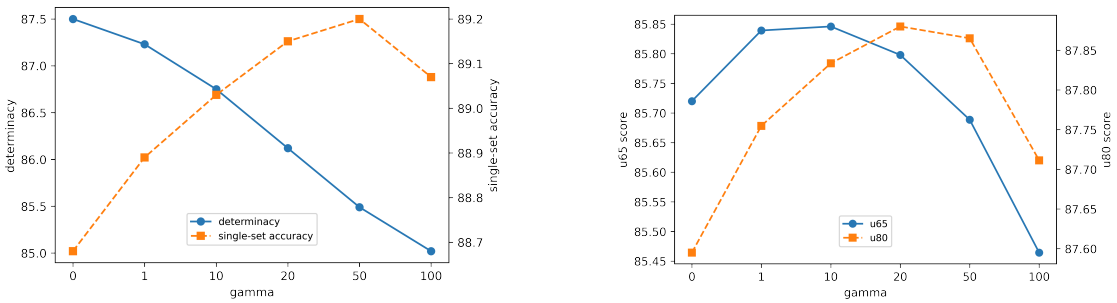
Table 3.11: Phase 2: Friedman statistic and p-value (left), Nemenyi p-values for pairwise model comparison. The best Friedman rank is printed in bold.

(a) Friedman rank and test

	EW	**OOBACC**	**OOBU65**	**IRF**	**AW**	**p-value**
cau	2.64	3.64	3.08	4.52	**1.12**	5.85×10^{-8}
ssa	3.08	3.36	2.88	4.64	**1.04**	3.89×10^{-9}
u65	2.96	3.60	2.96	4.40	**1.08**	1.32×10^{-8}
u80	3.04	3.48	2.76	4.64	**1.08**	2.20×10^{-8}

(b) Nemenyi test

AW	**vs. EW**	**vs. OOBACC**	**vs. OOBU65**	**vs. IRF**
cau	0.004	0.001	0.001	0.001
ssa	0.001	0.001	0.002	0.001
u65	0.001	0.003	0.001	0.001
u80	0.001	0.001	0.005	0.001



(a) Determinacy and single-set accuracy

(b) u_{65} and u_{80}

Figure 3.4: Average metrics computed over the 25 datasets, as a function of γ.

Our last experiment focuses on the influence of parameter γ, which was fixed using cross-validation in previous experiments. As explained above, this parameter has been introduced in the cost function to adjust the level of cautiousness in the model, so as to choose a specific behavior according to the user's needs: in general, the larger the value of γ, the more cautious the model, i.e., the lower the determinacy. Ideally, for each dataset, picking an appropriate value of γ would make it possible to reach the best compromise between determinacy and single-set accuracy.

Figures 3.4(a) and 3.4(b) illustrate the influence of γ on average determinacy, single-set accuracy, u_{65} and u_{80}, computed over all datasets. These metrics behave as expected: determinacy appears to be a decreasing function of γ, whereas single-set

accuracy is increasing. When the value of γ is too large (for example for $\gamma = 100$), single-set accuracy slightly decreases; an explanation for this behavior would be that the cost function then highly favors indeterminate predictions: turning determinate, correct predictions into indeterminate ones then leads to a decrease in accuracy. The u_{65} and u_{80} both present an optimum, obviously attained for different values of γ, which could be determined for instance by cross-validation.

3.4 Conclusion

In this chapter, we have proposed a new aggregation method of imprecise trees using belief functions to construct a cautious random forest for binary imprecise classification. Each tree in the forest provides intervals of probabilities obtained via the imprecise Dirichlet model, rather than point estimates. Our aggregation strategy can be regarded as an extension of the voting mechanism. We have also proposed a strategy for assigning weights to trees by optimizing a cost function that takes both determinacy and accuracy into account, which thus allows us to reach a better compromise between cautiousness and accuracy.

Our experiments showed that our aggregation method compares favorably to other aggregation operators leading to cautious decisions, such as averaging, majority voting (with indeterminate predictions), and reject option with a threshold. Experiments also show that our approach is robust to label noise and to the scarcity of training data. In a second series of experiments, we showed that our strategy for learning tree weights results in a more cautious model compared to the other four baselines, and achieves the best performances in terms of single-set accuracy, as well as u_{65} and u_{80} scores. In a nutshell, our strategy makes it possible to reach a good compromise between informativeness and cautiousness, by avoiding mistakes when the tree outputs appear to be too conflicting or too indeterminate.

The proposed imprecise tree aggregation strategy is only adapted to binary data. In the following chapter, we will introduce a more general aggregation scheme for multi-class data, of which the approach proposed in this chapter is a special case.

Chapter 4

Multi-class cautious random forests

In the previous chapter, we have proposed an aggregation strategy for imprecise trees in the context of binary cautious classification problems, within the framework of belief functions. In this chapter, we extend these previous works and address the multi-class case. For this purpose, we propose two cautious decision-making strategies (always in the belief functions framework) which generalize averaging and voting for tree ensembles. These strategies are axiomatically principled: they amount to maximizing the lower expected discounted utility rather than the expected utility as done in the conventional case. From the decision-making perspective, it makes cautious predictions by constructing partial preorders among partial assignments of classes for a given instance. Note that our approach can be applied to any kind of classifier ensemble where individual classifier outputs are probability intervals.

The structure of this chapter is as follows. In Section 4.1, we explain the discounted utility metric that we aim to maximize. Section 4.2 and 4.3 present the generalization of averaging and voting schemes in traditional random forests to the cautious random forests. We report experimental results and conduct an analysis in Section 4.4. Finally, a conclusion is drawn in Section 4.5.

4.1 Lower discounted utility maximization

Let m be a mass function defined on the frame of discernment $\Omega = \{c_1, \ldots, c_K\}$ with $K \geq 2$. Let us consider a given instance $\boldsymbol{x}$ and assume the associated prediction is set-valued, in the form of a non-empty subset $A \subseteq \Omega$. The lower expected utility of A associated with a utility matrix $\mathbf{U}$ is defined as follows:

$$\underline{\mathbb{E}}_m(A, \mathbf{U}) = \sum_{B \subseteq \Omega} m(B) \min_{c_k \in B} u_{Ak}. \tag{4.1}$$

The discounted utility of decision A when the real label of $\boldsymbol{x}$ is c_k is defined as

$$u_{Ak} = dr(|A|)\mathbb{1}(c_k \in A), \tag{4.2}$$

with $dr(\cdot)$ being a discounted utility function, such as dr_{acc}, dr_{u65}, dr_{u80} or dr_{F_β} defined by Eq. (2.13) to Eq. (2.16). In this chapter, we only consider dr_{u65} and dr_{u80} since dr_{acc} does not reward cautious decisions and dr_{F_β} is an intermediate between dr_{u65} and dr_{u80}. Hereafter, we will use the simplified notation dr_{u_α} as the discount ratio, where α is equal to 65 or 80 (examples of which are provided in Table 2.2).

Using such a utility matrix, the calculation of the lower expected utilities of any $A \subseteq \Omega$ is equivalent to calculating the product of its belief degree $Bel(A)$ and the corresponding discounted utility $dr_{u_\alpha}(|A|)$, as demonstrated in Theorem 4.1.

Theorem 4.1. *Given the utility matrix* $\mathbf{U}$ *of general term* $u_{Ak} = dr_{u_\alpha}(|A|)\mathbb{1}(c_k \in A)$ *where* c_k *refers to the actual class and* $A \subseteq \Omega$ *to an imprecise decision, the lower expected utility* $\underline{\mathbb{E}}_m(A, \mathbf{U})$ *is equal to* $dr_{u_\alpha}(|A|)Bel(A)$.

Proof. Following Eq. (4.1), and taking any $A \subseteq \Omega$ as action, we have

$$\begin{aligned}
\underline{\mathbb{E}}_m(A, \mathbf{U}) &= \sum_{B \subseteq \Omega} m(B) \min_{c_k \in B} u_{Ak} \\
&= \sum_{B \subseteq \Omega} m(B) \min_{c_k \in B} [dr_{u_\alpha}(|A|)\mathbb{1}(c_k \in A)] \\
&= dr_{u_\alpha}(|A|) \sum_{B \subseteq \Omega} m(B) \min_{c_k \in B} \mathbb{1}(c_k \in A) \\
&= dr_{u_\alpha}(|A|) \sum_{B \subseteq A} m(B) \\
&= dr_{u_\alpha}(|A|)Bel(A).
\end{aligned}$$

Indeed, for any $B \cap A \neq \emptyset$ such that $B \not\subseteq A$, there obviously exists $c_k \in B$ such that $c_k \notin A$: thus, $\min_{c_k \in B} \mathbb{1}(c_k \in A) = 1$ if, and only if, $B \subseteq A$. □

4.2 Generalization of averaging

We assume that the output of each decision tree $\boldsymbol{h}_t$ is a set of probability intervals as defined by Eq. (3.1), written $\mathcal{I}_{tk}(\boldsymbol{x}) = \left[\underline{p}_{tk}(\boldsymbol{x}), \overline{p}_{tk}(\boldsymbol{x})\right]$, $t = 1, \ldots, T$, $k = 1, \ldots, K$. According to [53], the corresponding quasi-Bayesian mass function associated with

$\mathcal{I}_{tk}(\boldsymbol{x})$ is

$$m_t(\{c_k\}) = \underline{p}_{tk},\ k = 1, \ldots, K; \quad m_t(\Omega) = 1 - \sum_{k=1}^{K} m_t(\{c_k\}). \tag{4.3}$$

These masses can then be averaged across all trees:

$$m(\{c_k\}) = \frac{\sum_{t=1}^{T} m_t(\{c_k\})}{T},\ k = 1, \ldots, K; \quad m(\Omega) = \frac{\sum_{t=1}^{T} m_t(\Omega)}{T}. \tag{4.4}$$

To make a decision based on this mass function, we build a sequence of nested subsets $A \subseteq \Omega$ by repeatedly aggregating the class with the highest mass, and we choose the subset $A^\star$ which maximizes $\underline{\mathbb{E}}(A) := \underline{\mathbb{E}}_m(A, \mathbf{U})$ over all $A \subseteq \Omega$. This procedure makes it possible to determine $A^\star$ in linear complexity, as shown by Theorem 4.2.

Theorem 4.2. *Consider the mass function in Eq. (4.4) with classes sorted by decreasing mass: $m(\{c_{(k)}\}) \geq m(\{c_{(k+1)}\})$, for $k = 1, \ldots, K-1$. Scanning the sequence of nested subsets $\{c_{(1)}\} \subset \{c_{(1)}, c_{(2)}\} \subset \cdots \subseteq \Omega$ makes it possible to identify the subset $A^\star = \arg\max \underline{\mathbb{E}}(A)$ in complexity $O(K)$.*

Proof. Since the masses $m(\{c_{(k)}\})$ are sorted in decreasing order, the focal element with the highest belief among those of cardinality i is $A_i^\star = \{c_{(k)}, k = 1, \ldots, i\}$, i.e., $Bel(A_i^\star) = \sum_{k=1}^{i} m(\{c_{(k)}\}) \geq Bel(B)$, for all $B \subseteq \Omega$ such that $|B| = i$. Since $dr_{u_\alpha}(|A|)$ only depends on $|A|$, $A_i^\star$ maximizes the lower expected utility over all subsets of size i. As a consequence, keeping the subset with maximal lower expected utility in the sequence of nested subsets defined above computes the maximizer $A^\star$ in time complexity $O(K)$. □

The overall procedure, hereafter referred to as CDM_Ave, extends classical averaging for precise probabilities to averaging mass functions across imprecise trees. It is summarized in Algorithm 2.

Note that a theorem similar to Theorem 4.2 was independently proven in [143], which addressed set-valued prediction in a probabilistic framework for a wide range

Algorithm 2: Cautious Decision Making by Averaging

Input: Tree outputs $\left\{(\underline{p}_{tk}(\boldsymbol{x}), \overline{p}_{tk}(\boldsymbol{x})), t = 1, \ldots, T,\ k = 1, \ldots, K\right\}$, discount ratio dr_{u_α}

Output: Decision A

```
for k = 1, ..., K do
    m({c_k}) = 1/T × Σ_{t=1}^{T} p_tk
m(Ω) = 1 − Σ_{k=1}^{K} m({c_k})
Sort classes by decreasing mass: m({c_(1)}) ≥ m({c_(2)}) ≥ ··· ≥ m({c_(K)})
A = ∅
bel = 0
mleu = 0                                   // maximum lower expected utility
for i = 1, ..., K do
    bel = bel + m({c_(i)})
    leu = dr_{u_α}(i) × bel                // lower expected utility of {c_(1), ..., c_(i)}
    if leu > mleu then
        mleu = leu
        A = A ∪ {c_(i)}
Return A
```

of utility functions. Since the masses considered here are quasi-Bayesian, the procedure described in Algorithm 2 is close to that described in [143]. The overall complexity of Algorithm 2 is $O(K \log K)$ due to sorting the classes by decreasing mass.

Example 4.1 (Cautious decision-making via generalised averaging)**.** *Assume the averaged mass function on* $\Omega = \{c_1, c_2, c_3, c_4\}$ *is given as follows:*

$$m(\{c_1\}) = 0.32,\ m(\{c_2\}) = 0.48,\ m(\{c_3\}) = 0.04,\ m(\{c_4\}) = 0.06,\ m(\Omega) = 0.05.$$

The classes ordered by decreasing mass are thus $\{c_2, c_1, c_4, c_3\}$*. These classes are added to the prediction one by one and the corresponding expected lower discounted utilities (using* $dr_{u_{65}}$*) are calculated:*

- $\underline{\mathbb{E}}(\{c_2\}) = dr_{u_{65}}(1)Bel(\{c_2\}) = 1 \times 0.48 = 0.48$*;*
- $\underline{\mathbb{E}}(\{c_2, c_1\}) = dr_{u_{65}}(2)Bel(\{c_2, c_1\}) = 0.65 \times 0.8 = 0.52$*;*
- $\underline{\mathbb{E}}(\{c_2, c_1, c_4\}) = dr_{u_{65}}(3)Bel(\{c_2, c_1, c_4\}) = 0.4667 \times 0.86 = 0.401$*;*

- $\underline{\mathbb{E}}(\{c_2, c_1, c_4, c_3\}) = dr_{u_{65}}(4)Bel(\Omega) = 0.3625 \times 1 = 0.3625.$

We can find that $\{c_2, c_1\}$ *reaches the maximum expected lower discounted utility: thus, the cautious prediction made is* $A^\star = \{c_2, c_1\}$.

4.3 Generalization of voting

We now address the combination of probability intervals via voting. Our approach consists of identifying first, for each tree, the set of non-dominated classes according to the interval dominance criterion, which amounts to letting each tree vote for the corresponding subset of classes. Then, we combine the trees, again by computing the subset $A^\star$ maximizing $\underline{\mathbb{E}}(A)$ over all $A \subseteq \Omega$. Algorithm 3 describes how interval dominance criterion can be used to aggregate all tree outputs into a single mass function m, in time complexity $O(TK^2)$.

In this approach, the focal elements of m obtained can be any subset of Ω. Since m is not quasi-Bayesian anymore, maximizing the lower expected utility requires in principle checking all subsets of Ω in the decision step: this check has a worst-case complexity of $O(2^K)$, which prohibits using this strategy for datasets with large numbers of classes. In order to reduce the complexity, we introduce three tricks:

(i) we arbitrarily restrict the decision to subsets $A \subseteq \Omega$ with cardinality $|A| \leq \overline{K} < K$, which reduces the complexity to $O(\sum_{k=1}^{\overline{K}} \binom{K}{k})$;

(ii) when searching for a maximizer of the lower expected utility by scanning subsets of classes of increasing cardinality, we show that the procedure can be stopped when larger subsets are known not to further improve the lower expected utility (see Proposition 4.1);

(iii) during this search, for a given cardinality i, only subsets A composed of classes appearing in focal elements B such that $|B| \leq i$ need to be considered (see Proposition 4.2).

Algorithm 3: Tree aggregation via interval dominance

Input: Tree outputs $\left\{(\underline{p}_{tk}(\boldsymbol{x}), \overline{p}_{tk}(\boldsymbol{x})), t = 1, \ldots, T,\ k = 1, \ldots, K\right\}$
Output: Mass function m

$m(A) = 0, \forall A \subseteq \Omega$
for $t = 1, \ldots, T$ **do**
  $DC = \emptyset$ `// set of dominated classes`
  for $k = 1, \ldots, K$ **do**
    for $j = 1, \ldots, K$ *and* $j \neq k$ **do**
      if $\overline{p}_{tk} < \underline{p}_{tj}$ **then**
        $DC = DC \cup c_k$
        break
  $NDC = \Omega \setminus DC$ `// non-dominated classes`
  $m(NDC) = m(NDC) + \frac{1}{T}$
Return m

Proposition 4.1. *If the lower expected utility of a subset $A \subseteq \Omega$ is (strictly) greater than $dr_{u_\alpha}(i)$ for some $i > |A|$, then it is (strictly) greater than the lower expected utility of any subset $B \subseteq \Omega$ with cardinality $|B| \geq i$.*

Proof. Let $A \subseteq \Omega$ be a subset of classes (typically, the current maximizer of the lower expected utility in the procedure described in Algorithm 4). Assume that $\underline{\mathbb{E}}(A) > dr_{u_\alpha}(i)$ for some $i > |A|$. Since $Bel(B) \leq 1$ for all $B \subseteq \Omega$, then $\underline{\mathbb{E}}(A) > \underline{\mathbb{E}}(B)$ for all subsets B such that $|B| = i$. The generalization to all subsets B such that $|B| > i$ comes from $dr_{u_\alpha}(i)$ being monotone decreasing in i. □

Proposition 4.2. *The subset $A_i^\star \subseteq \Omega$ maximizing the lower expected utility among all A such that $|A| = i$ is a subset of Ω_i that consists of classes appearing in focal elements B such that $|B| \leq i$.*

Proof. Let Ω_i be the set of classes appearing in focal elements of cardinality less than or equal to i, for some $i \in \{1, \ldots, K\}$. Assume a subset A of cardinality i is such that $A = A_1 \cup A_2$, with $A \cap \Omega_i = A_1$, then, $Bel(A) = Bel(A_1)$. If $A_2 \neq \emptyset$, then $\underline{\mathbb{E}}(A) < \underline{\mathbb{E}}(A_1)$ since $|A_1| < |A|$: classes $c_k \notin \Omega_i$ necessarily decrease $\underline{\mathbb{E}}(A)$. Moreover, since $Bel(A)$ sums masses $m(B)$ of subsets $B \subseteq A$, any focal element B such that $|B| > i$ does not contribute to $Bel(A)$. □

Algorithm 4: Cautious Decision Making by Voting

Input: Tree outputs $\mathcal{I}_{tk} = \left\{(\underline{p}_{tk}(\boldsymbol{x}), \overline{p}_{tk}(\boldsymbol{x})), t = 1, \ldots, T,\ k = 1, \ldots, K\right\}$, cardinality bound $\overline{K}$, discount ratio dr_{u_α}

Output: Decision A

Obtain m via Alg 3(I_{tk}, $t = 1, \ldots T$, and $k = 1, \ldots, K$)
$FE = \emptyset$ // focal elements
$\Omega_i = \emptyset$ // considering classes
$A = \emptyset$
$mleu = 0$ // maximum lower expected utility
for $i = 1, \ldots, M$ **do**
  // trick 1
  $dr = dr_{u_\alpha}(i)$
  if $mleu > dr$ **then**
    Return A // trick 2, see Proposition 4.1
  else
    $FE = FE \cup \{B : m(B) > 0, |B| = i, B \subseteq \Omega\}$
    $\Omega_i = \Omega_i \cup \{c : c \in B, B \in FE\}$ // trick 3, see Proposition 4.2
    for *all* $B \subseteq \Omega_i$ *and* $|B| = i$ **do**
      $bel = \sum_{C \in FE,\ C \subseteq B} m(C)$
      $leu = dr \times bel$ // lower expected utility for B
      if $leu > mleu$ **then**
        $mleu = leu$
        $A = B$
Return A

The procedure described in Algorithm 4, hereafter referred to as CDM_Vote, extends voting when votes are expressed as subsets of classes and returns the subset $A^\star = \arg\max \underline{\mathbb{E}}(A)$ among all subsets $A \subseteq \Omega$ such that $|A| \leq \overline{K} \leq K$. It generalizes the method proposed in Chapter 3 for binary cautious classification, which amounts to maximizing the discounted accuracy dr_{acc} for binary cases. CDM_Vote is computationally less efficient than CDM_Ave by design, even if the time complexity can be controlled. However, as it will be shown in the experimental part, this approach remains able to deal with cautious classification problems of a large number of classes.

Example 4.2 (Cautious decision-making via generalised voting)**.** *Assume the mass function on* $\Omega = \{c_1, c_2, c_3, c_4\}$ *obtained via Algorithm 3 is as follows:*

$$m(\{c_1\}) = 0.15,\ m(\{c_2\}) = 0.25, m(\{c_1, c_2\}) = 0.35,\ m(\{c_1, c_3\}) = 0.05,$$
$$m(\{c_2, c_3\}) = 0.1,\ m(\{c_2, c_3, c_4\}) = 0.05,\ m(\Omega) = 0.05.$$

Let us apply Algorithm 4 to make a decision (using $dr_{u_{65}}$*):*

- *for subsets of* Ω *with cardinality 1, we only consider classes that appear in focal elements with cardinality 1, i.e.,* $\Omega_1 = \{c_1, c_2\}$*:*

$$\underline{\mathbb{E}}(\{c_1\}) = dr_{u_{65}}(1)Bel(\{c_1\}) = 1 \times 0.15 = 0.15,$$

$$\underline{\mathbb{E}}(\{c_2\}) = dr_{u_{65}}(1)Bel(\{c_2\}) = 1 \times 0.25 = 0.25,$$

  the current maximum lower expected utility is reached by $\{c_2\}$ *with value* $0.25 < dr_{u_{65}}(2) = 0.65$*, continue;*

- *for subsets of* Ω *with cardinality 2, we only consider classes that appear in focal elements with cardinality smaller than or equal to 2, i.e.,* $\Omega_2 = \{c_1, c_2, c_3\}$*:*

$$\underline{\mathbb{E}}(\{c_1, c_2\}) = dr_{u_{65}}(2)Bel(\{c_1, c_2\}) = 0.65 \times 0.75 = 0.4875,$$

$$\underline{\mathbb{E}}(\{c_1, c_3\}) = dr_{u_{65}}(2)Bel(\{c_1, c_3\}) = 0.65 \times 0.2 = 0.13,$$

$$\underline{\mathbb{E}}(\{c_2, c_3\}) = dr_{u_{65}}(2)Bel(\{c_2, c_3\}) = 0.65 \times 0.35 = 0.2275,$$

  the current maximum lower expected utility is reached by $\{c_1, c_2\}$ *with value* $0.4875 > dr_{u_{65}}(3) = 0.4667$*, stop.*

The final set-valued prediction is then $A^\star = \{c_1, c_2\}$ *since any subset* $B \subseteq \Omega$ *with* $|B| > 2$ *has a smaller lower expected utility than* $A^\star$*. Here, class* c_4 *has never been considered because it first appears in focal element* $\{c_2, c_3, c_4\}$ *with a cardinality of three, which was known not to ameliorate the lower expected utility.*

4.4 Experiments and results

In order to illustrate the performance of our imprecise tree aggregation approaches, we compare them with two other existing aggregation methods on various datasets using different evaluation metrics for cautious classification problems. The two existing approaches used for comparison are Minimum Vote Against (MVA) and Averaging (AVE), which can respectively be considered as generalizations of voting and averaging for random forests. Evaluation metrics include determinacy, single-set accuracy, set accuracy, set size, u_{65} score, and u_{80} score. Details about the datasets used in experiments are presented in Table 4.1.

Table 4.1: Description of datasets, including the number of instances, features, and classes.

Datasets	**n_instance**	**n_feature**	**n_class**
Balance-scale	625	4	3
Ecoli	366	7	7
Letter	2000	16	26
Optdigits	5620	64	10
Page-blocks	5473	10	5
Pendigits	10992	16	10
Segment	2310	19	7
Spectrometer	531	100	48
Vehicle	946	18	4
Vowel	990	11	11
Waveform	5000	40	3
Wine	101	13	3

For all experiments, we used the scikit-learn implementation of random forests [157] with the default parameter setting. To construct imprecise decision trees, we chose $s = 2$ for the IDM. The CDM_Vote and CDM_Ave procedures use the d_{65} discounted ratio to make decisions. Each metric on each dataset and each aggregation approach is evaluated through 10 times 10-fold cross-validation and presented with mean and standard deviation values. It should be noted that the same imprecise trees are used to compare the different aggregation approaches.

4.4.1 Decision-Making efficiency

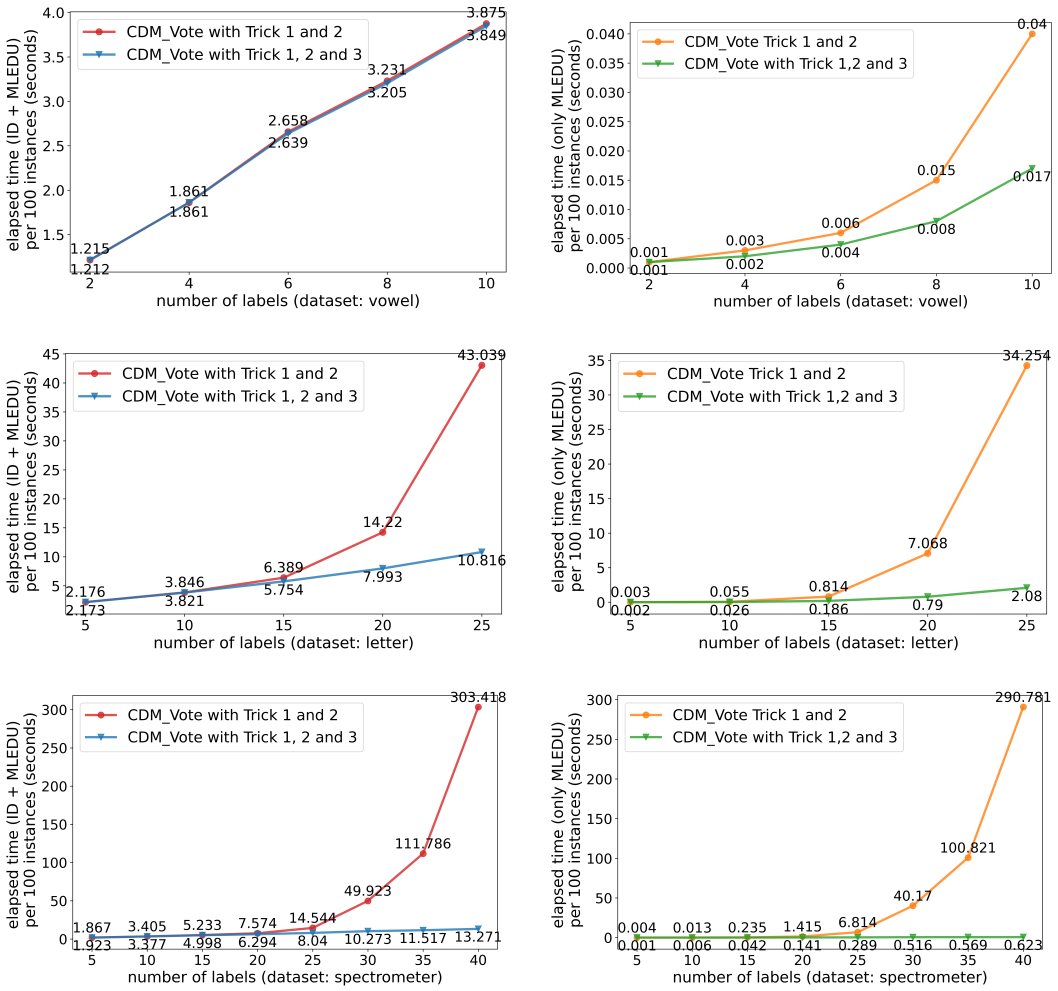


Figure 4.1: Decision-making time complexity of CDM_Vote according to the number of labels (for 100 samples). Up to down: *vowel*, *letter*, and *spectrometer*; left: ID+MLEDU, right: MLEDU only.

First, we studied the time complexity of CDM_Vote strategy as a function of the number of labels. For a given integer i, we first picked i labels at random and extracted the corresponding samples from original dataset. Then, we trained a random forest with the parameter s of the IDM set to one, and processed the test data using CDM_Vote. During the test phase, we recorded for each sample the elapsed time of the entire process (interval dominance plus maximizing lower expected discounted utility), and the elapsed time needed to maximize the lower expected utility after having applied interval dominance, respectively referred to as ID+MLEDU and MLEDU. For each i, we report average elapsed time per 100

inferences, computed over 10 repetitions of the above process. Since for high values of i, decision-making would be intractable without any control of the complexity, we compared the efficiency when using all tricks in Section 4.2 with that when using only the two first ones.

Fig. 4.1 shows that for a small number of labels (less than 15), trick 3 (filtering out subsets $A \not\subseteq \Omega_i$) does not significantly improve the efficiency, as the time required for applying interval dominance prevails. However, for a large number of labels, the time required for maximizing the lower expected utility dominates, and filtering out subsets $A \not\subseteq \Omega_i$ accelerates significantly the procedure. Apart from interval dominance, this filtering step accelerates the decision-making process regardless of the number of labels, as shown in the right column of Fig. 4.1. This experiment demonstrates that CDM_Vote remains applicable with a large number of labels.

4.4.2 Performance comparison on original data

In this section, the results obtained for the various aggregation approaches on original data are illustrated from Table 4.2 to Table 4.7 and analyzed. For each evaluation metric, the best average result is marked in bold.

Table 4.2: Comparison of aggregation approaches using the determinacy on each multi-class dataset (without label noise).

Dataset	**MVA**	**AVE**	**CDM_Vote**	**CDM_Ave**
Balance-scale	0.9934±0.0099	0.7201±0.0575	0.7472±0.0563	0.7390±0.0566
Ecoli	0.9958±0.0119	0.7756±0.0633	0.8889±0.0521	0.8788±0.0591
Letter	0.9882±0.0080	0.7726±0.0263	0.8163±0.0261	0.8114±0.0262
Optdigits	0.9977±0.0033	0.8679±0.0234	0.9405±0.0176	0.9374±0.0175
Page-blocks	0.9994±0.0018	0.9675±0.0124	0.9786±0.0101	0.9778±0.0101
Pendigits	0.9990±0.0022	0.9329±0.0160	0.9601±0.0113	0.9583±0.0110
Segment	0.9992±0.0021	0.9541±0.0138	0.9653±0.0118	0.9637±0.0127
Spectrometer	0.9786±0.0231	0.5443±0.0712	0.4796±0.0629	0.4989±0.0636
Vehicle	0.9910±0.0105	0.6303±0.0495	0.7238±0.0485	0.7069±0.0489
Vowel	0.9943±0.0076	0.6704±0.0494	0.8590±0.0380	0.8596±0.0355
Waveform	0.9950±0.0044	0.7690±0.0324	0.7338±0.0319	0.7261±0.0329
Wine	1.0±0.0	0.9746±0.0367	0.9652±0.0461	0.9629±0.0444
Average	**0.9943**	0.7983	0.8382	0.8351

Table 4.3: Comparison of aggregation approaches using the single-set accuracy on each multi-class dataset (without label noise).

Dataset	**MVA**	**AVE**	**CDM_Vote**	**CDM_Ave**
Balance-scale	0.8894±0.0390	0.9888±0.0167	0.9844±0.0209	0.9861±0.0182
Ecoli	0.8745±0.0517	0.9259±0.0465	0.8990±0.0505	0.9017±0.0504
Letter	0.8607±0.0259	0.9642±0.0163	0.9428±0.0176	0.9487±0.0164
Optdigits	0.9651±0.0136	0.9952±0.0051	0.9857±0.0083	0.9875±0.0076
Page-blocks	0.9690±0.0121	0.9824±0.0088	0.9787±0.0101	0.9796±0.0097
Pendigits	0.9716±0.0099	0.9894±0.0077	0.9870±0.0074	0.9879±0.0073
Segment	0.9742±0.0117	0.9912±0.0072	0.9879±0.0073	0.9890±0.0072
Spectrometer	0.5497±0.0681	0.6942±0.0737	0.6998±0.0759	0.6897±0.0769
Vehicle	0.7456±0.0413	0.9166±0.0396	0.8613±0.0419	0.8725±0.0444
Vowel	0.9436±0.0254	0.9942±0.0105	0.9831±0.0149	0.9872±0.0124
Waveform	0.8464±0.0268	0.9192±0.0238	0.9281±0.0225	0.9304±0.0225
Wine	0.9825±0.0273	0.9883±0.0234	0.9905±0.0218	0.9911±0.0213
Average	0.8810	**0.9458**	0.9357	0.9376

Table 4.4: Comparison of aggregation approaches using the set accuracy on each multi-class dataset (without label noise).

Dataset	**MVA**	**AVE**	**CDM_Vote**	**CDM_Ave**
Balance-scale	0.9286±0.1750	0.8844±0.0826	0.9547±0.0607	0.9910±0.0263
Ecoli	1.0±0.0	0.9691±0.0616	0.9625±0.0930	0.9559±0.0963
Letter	0.7173±0.2588	0.9491±0.0296	0.7098±0.0782	0.7284±0.0713
Optdigits	0.9459±0.1553	0.9983±0.0076	0.9332±0.0654	0.9431±0.0675
Page-blocks	1.0±0.0	0.9632±0.0941	0.9652±0.0967	0.9672±0.0800
Pendigits	1.0±0.0	0.9941±0.0210	0.9386±0.0866	0.9494±0.0805
Segment	1.0±0.0	0.9705±0.0569	0.9218±0.1040	0.9101±0.1178
Spectrometer	0.7410±0.2803	0.8167±0.0796	0.7221±0.0973	0.7124±0.0895
Vehicle	0.9362±0.1559	0.9873±0.0208	0.9636±0.0357	0.9795±0.0312
Vowel	0.9919±0.0514	0.9990±0.0055	0.9509±0.0612	0.9641±0.0529
Waveform	1.0±0.0	1.0±0.0	1.0±0.0	1.0±0.0
Wine	NA	1.0±0.0	1.0±0.0	1.0±0.0
Average	0.9328	**0.9610**	0.9185	0.9251

Table 4.5: Comparison of aggregation approaches using the set size on each multi-class dataset (without label noise).

Dataset	**MVA**	**AVE**	**CDM_Vote**	**CDM_Ave**
Balance-scale	2.0294±0.1690	2.7213±0.1199	2.9450±0.0666	2.9892±0.0316
Ecoli	2.0833±0.2764	4.7917±1.1838	2.1552±0.5295	2.0267±0.1590
Letter	2.0774±0.2081	12.1968±1.3898	2.1390±0.0584	2.1627±0.0623
Optdigits	2.0±0.0	6.6066±0.7170	2.0574±0.0747	2.0607±0.0681
Page-blocks	2.0±0.0	2.8837±0.5161	2.2613±0.4691	2.1976±0.4148
Pendigits	2.0±0.0	4.5456±0.9741	2.0273±0.0591	2.0383±0.0871
Segment	2.0±0.0	3.0340±0.6089	2.0231±0.1119	2.0082±0.0305
Spectrometer	2.0667±0.2222	9.5821±3.2134	2.1317±0.0722	2.1210±0.0653
Vehicle	2.0189±0.1361	2.7920±0.1511	2.5447±0.1975	2.7308±0.2051
Vowel	2.0233±0.1507	6.1601±0.7814	2.0488±0.0573	2.0428±0.0547
Waveform	2.0±0.0	2.0153±0.0200	2.0393±0.0271	2.0775±0.0340
Wine	NA	2.0270±0.1622	2.0426±0.1727	2.1599±0.3531
Average	**2.0272**	4.9464	2.2013	2.2180

Table 4.6: Comparison of aggregation approaches using the u_{65} score on each multi-class dataset (without label noise).

Dataset	**MVA**	**AVE**	**CDM_Vote**	**CDM_Ave**
Balance-scale	0.8859±0.0396	0.8363±0.0350	0.8487±0.0350	0.8496±0.0328
Ecoli	0.8736±0.0521	0.8097±0.0523	0.8672±0.0475	0.8672±0.0470
Letter	0.8548±0.0256	0.8094±0.0231	0.8518±0.0206	0.8562±0.0201
Optdigits	0.9641±0.0135	0.9071±0.0171	0.9626±0.0099	0.9634±0.0095
Page-blocks	0.9688±0.0120	0.9671±0.0111	0.9704±0.0110	0.9712±0.0107
Pendigits	0.9710±0.0098	0.9536±0.0129	0.9719±0.0084	0.9723±0.0087
Segment	0.9738±0.0116	0.9699±0.0097	0.9743±0.0088	0.9745±0.0087
Spectrometer	0.5453±0.0656	0.5384±0.0497	0.5706±0.0512	0.5683±0.0517
Vehicle	0.7438±0.0417	0.7709±0.0298	0.7744±0.0318	0.7721±0.0321
Vowel	0.9418±0.0255	0.7895±0.0357	0.9301±0.0219	0.9353±0.0197
Waveform	0.8454±0.0267	0.8562±0.0189	0.8521±0.0185	0.8496±0.0184
Wine	0.9825±0.0273	0.9797±0.0265	0.9785±0.0276	0.9776±0.0269
Average	0.8792	0.8490	0.8794	**0.8798**

Table 4.7: Comparison of aggregation approaches using the u_{80} score on each multi-class dataset (without label noise).

Dataset	**MVA**	**AVE**	**CDM_Vote**	**CDM_Ave**
Balance-scale	0.8864±0.0397	0.8701±0.0303	0.8809±0.0296	0.8841±0.0263
Ecoli	0.8742±0.0517	0.8336±0.0504	0.8829±0.0460	0.8844±0.0453
Letter	0.8562±0.0258	0.8263±0.0221	0.8709±0.0197	0.8763±0.0189
Optdigits	0.9645±0.0134	0.9192±0.0155	0.9709±0.0088	0.9722±0.0084
Page-blocks	0.9689±0.0120	0.9713±0.0103	0.9734±0.0105	0.9743±0.0102
Pendigits	0.9712±0.0098	0.9613±0.0120	0.9775±0.0078	0.9782±0.0080
Segment	0.9739±0.0116	0.9758±0.0087	0.9790±0.0083	0.9794±0.0081
Spectrometer	0.5463±0.0657	0.5802±0.0516	0.6264±0.0552	0.6211±0.0547
Vehicle	0.7449±0.0418	0.8203±0.0275	0.8115±0.0311	0.8112±0.0313
Vowel	0.9426±0.0255	0.8224±0.0325	0.9501±0.0193	0.9554±0.0170
Waveform	0.8462±0.0267	0.8908±0.0183	0.8919±0.0171	0.8903±0.0171
Wine	0.9825±0.0273	0.9835±0.0241	0.9837±0.0237	0.9831±0.0231
Average	0.8798	0.8712	0.8999	**0.9008**

From the perspective of determinate predictions, according to Table 4.2 and Table 4.3, we can conclude that MVA often results in determinate decisions, whereas AVE tends to be very cautious; our proposed CDM_Vote and CDM_Ave methods turn out to be in-between. The same can be observed for the single-set accuracy, which is negatively correlated to determinacy. Our proposed approaches have a much higher single-set accuracy than MVA but slightly lower than AVE. CDM_Ave is a little bit more cautious than CDM_Vote, but the difference is very small.

Table 4.4 and Table 4.5 make it possible to compare the aggregation methods based on indeterminate predictions. MVA only makes indeterminate predictions for a few samples, achieving a high set accuracy as expected. In addition, due to the design of the model, almost all indeterminate predictions contain only two class labels. AVE makes indeterminate predictions for more samples, but its set accuracy still remains very high because each indeterminate prediction contains much more labels than other methods. Our proposals do not perform as well as MVA and AVE regarding set accuracy but remain comparable. Remark that the indeterminate predictions made by CDM_Ave and CDM_Vote are overall more precise than those made by AVE, containing fewer labels, which is very significant over datasets of a large number of labels, e.g., *Letter*, *Spectrometer*, and *Vowel*.

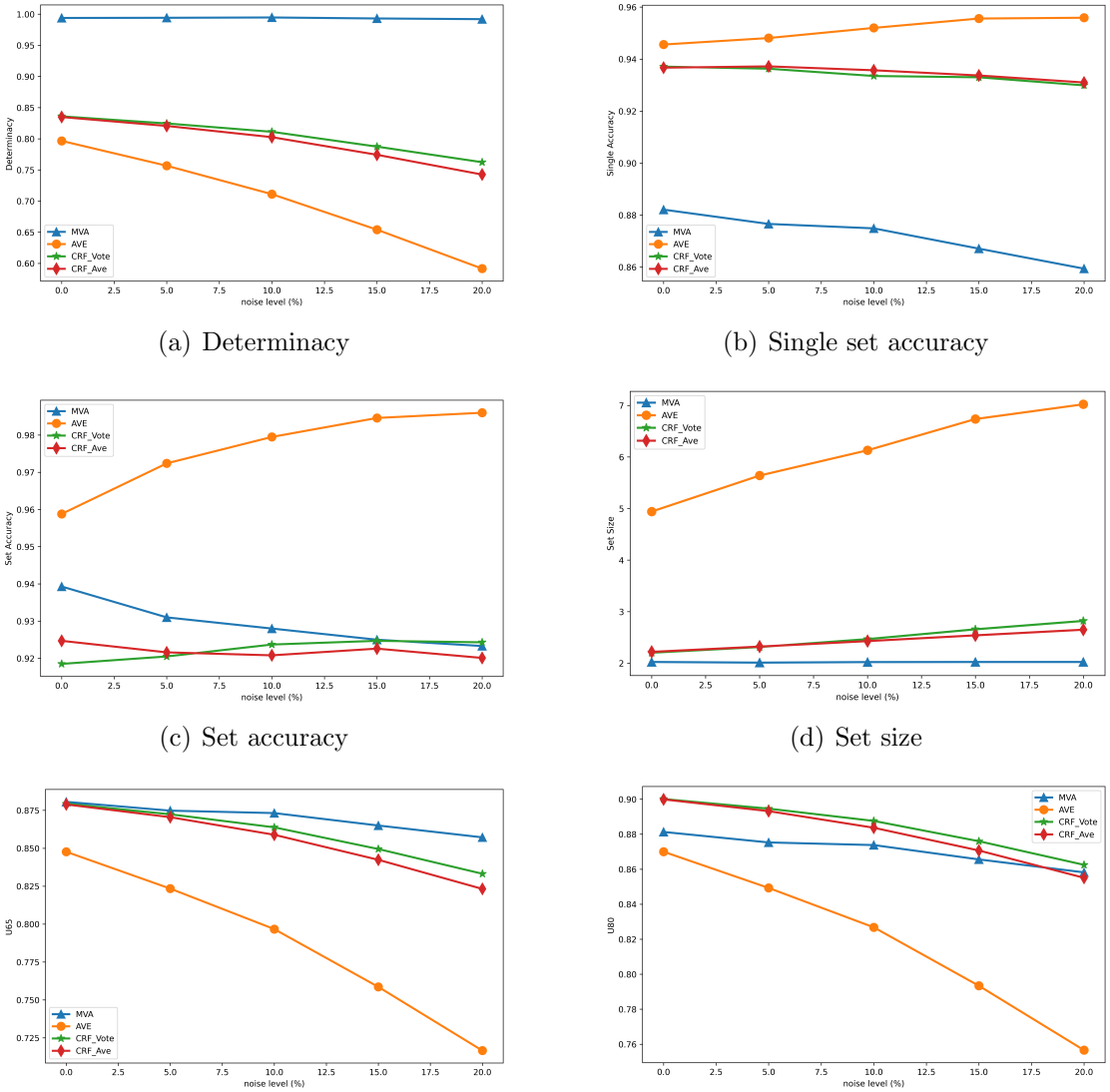
The results in Table 4.6 and Table 4.7 lead to the conclusion that CDM_Vote and CDM_Ave achieve a better compromise between accuracy and determinacy than MVA and AVE. It should be noted that the u_{65} score of either of our proposals is similar to that of MVA since the majority of datasets here are quite easy to classify: the single-set accuracy is therefore quite high, and u_{65} compensates less for indeterminate predictions. For relatively difficult datasets, such as *Spectrometer*, *Vehicle* and *Waveform*, CDM_vote and CDM_Ave significantly outperform MVA in terms of u_{65}. According to the u_{80} score, they outperform the other two methods, because u_{80} compensates more for indeterminate predictions and AVE turns out to be too cautious on these datasets.

Overall, there does not seem to be any significant difference between CDM_Vote and CDM_Ave. CDM_Ave seems to be slightly more cautious than CDM_Vote, this difference being more obvious on difficult datasets. Beyond performance, another huge advantage of CDM_Ave is that its time complexity for prediction-making is much lower than that of CDM_Vote. Therefore, CDM_Ave seems more adequate when facing datasets with a large number of classes.

4.4.3 Performance comparison on noisy data

Here, we study the behavior of the aggregation methods in the presence of label noise, by averaging the evaluation metrics computed for various noise levels over the 10 datasets considered. Label noise is introduced by randomly selecting a given proportion of training samples and replacing each corresponding label with a randomly selected label different from the actual one. According to Fig. 4.2(a) to Fig. 4.2(f), we can find that our proposed methods are more robust to noise in labels than both MVA and AVE.

Compared to MVA, even though the determinacy of CDM_Vote and CDM_Ave decreases when the level of noise gets high, their single-set accuracy and set accuracy remain at the same high level, meaning that whenever the decisions made by CDM_Vote and CDM_Ave are determinate, they are more reliable than those of MVA. Another serious shortcoming of MVA is that it does not decrease determinacy



(a) Determinacy

(b) Single set accuracy

(c) Set accuracy

(d) Set size

(e) u_{65} score

(f) u_{80} score

Figure 4.2: Evaluation metrics averaged across all datasets in the function of noise levels in training labels.

or increase the size of the set of predicted classes when facing a high level of noise or difficult data.

Compared to AVE, our model predictions decrease in terms of determinacy and increase in size, but less than AVE, which leads to higher u_{65} and u_{80} scores. AVE is very sensitive to label noise: as the noise level increases, determinacy significantly decreases, and the average set size dramatically increases. This disadvantage makes the informativeness of the model predictions decrease.

4.5 Conclusion

In this chapter, we have proposed two aggregation strategies to make cautious decisions from imprecise trees for multi-class cautious classification problems. In this setting, each tree provides probability intervals as outputs, which are typically obtained by using the imprecise Dirichlet model. The two strategies respectively generalize averaging and voting for classical tree ensembles. In both cases, they aim at making decisions by maximizing the lower expected discounted utility, thus providing set-valued predictions. These algorithms also generalize those presented in Chapter 3 for binary classification. The experiments conducted on different datasets confirm the interest of our proposals in order to achieve a good compromise between model accuracy and determinacy, especially for difficult datasets. Furthermore, by restricting the cardinality of the set-valued predictions, and by leveraging two tricks that avoid scanning all subsets of classes, our cautious decision-making procedure is able to process datasets with a large number of classes in a limited computational complexity.

Part II

Explanations in cautious random forests

Chapter 5

Explainable artificial intelligence

Artificial intelligence algorithms, including machine learning and deep learning algorithms, are nowadays widely deployed in many fields, such as recommendation systems, healthcare, and finance [130]. These deployments have achieved significant success, owing to their complex algorithmic design and the abundance of training data. However, evaluating models exclusively based on their accuracy is not sufficient, since in a wide range of applications, the decision-making process needs to be explainable [62]. For instance, in criminal justice, knowing why a person is guilty or not is more critical than the court judgment; as well, in medical diagnosis, giving the basis for the diagnosis is a key factor in choosing treatments [140]. Therefore, the research field of *eXplainable Artificial Intelligence* (XAI), which aims at making artificial intelligence algorithms more transparent, interpretable, and accountable to humans, has received increasing attention since its recent emergence [124].

In this chapter, we provide an introduction to XAI and a review of the main explanation methods. In Section 5.1, we start from the definition of explainability and its importance. Then, we discuss the prominent properties of good explanations, establish a taxonomy of explanation methods, and investigate different types of explanations for machine learning models. In Section 5.2, we focus on explanation methods for random forests and provide a mind map to choose the proper explanation method according to the context requirements. Finally, a conclusion is drawn in Section 5.3.

5.1 Introduction to XAI

5.1.1 Explainability and its necessity

Interpretability and explainability

Defining *interpretability* or *explainability* in a mathematical manner is challenging due to their subjective and context-dependent nature, as well as the absence of a uniform framework or consensus across different domains [145]. Nevertheless, several popular non-mathematical definitions have been proposed. According to Miller,

interpretability can be defined as the extent to which a human can understand the cause behind a decision [140]. Another definition proposed by Kim in [111] states that interpretability is the degree to which a human can consistently predict the results produced by a model.

In XAI, interpretability and explainability are often used interchangeably, but there are indeed subtle differences between them [13]. Interpretability refers to the degree to which a model or system's behavior can be understood by humans. It focuses on providing insights into how the system functions, which factors or features it considers, and how it produces its predictions or decisions. Interpretability allows humans to form mental models or mental representations of the system's behavior, enabling them to trust, validate, and potentially modify or improve its functioning. Explainability, on the other hand, aims at providing explanations or justifications for specific outputs or decisions made by the system. It aims to provide understandable reasons for why a particular prediction or outcome was produced. Explainability focuses on the "why" aspect rather than the overall understanding of the model operation. In our research, we use the terms explainability and explanation since we mainly focus on providing explanations for the model outputs rather than its inner functioning.

In the remainder of this section, we investigate the importance of explainability as well as the properties of good explanations. We also establish a taxonomy of explanation methods according to different perspectives and present different types of explanations for machine learning model outputs.

Importance of explainability

The importance and the necessity for explainability have been emphasized by many previous works [62, 88, 140, 141], which can be summarized as follows.

1. *Fairness*: by default, artificial intelligence models tend to reproduce biases that present in the training data. This fact can result in decision-making processes being influenced with respect to sensitive features, such as race, gender,

or religion. The COMPAS algorithm is a well-known example of a predictive model for "risk of criminal recidivism", which was found to exhibit significant ethnic bias against black people. XAI aims at facilitating the detection of such biases [163, 179, 190].

2. *Reliability and robustness*: the trained machine learning models may work well on large-scale test data, but they may also make errors in some situations, either facing corner cases or adversarial attacks. XAI technologies can improve robustness, i.e., avoiding that small changes in the input samples result in large changes in the predictions, for the sake of model reliability [158].

3. *Regulation requirements*: the regulations on artificial intelligence are rapidly evolving, as governments and organizations strive to ensure that these technologies are used ethically and responsibly [195]. The *General Data Protection Regulation* (GDPR), which went into effect in 2018, requires agencies to provide explanations for automated decisions that impact individuals, called "the right to explanation" [188, 209]. Then, in 2019, the *Algorithmic Accountability Act* was introduced in the United States Congress, requiring companies to conduct impact assessments for their algorithms to ensure they are fair, transparent, and accountable [138].

4. *Trust*: if the model can only provide the user with a prediction, but cannot explain how the result was derived, then it may be difficult for the user to trust the model, even though the model is highly accurate [45]. XAI enables users to access more information provided by the model, combined with their own perceptions to make the final decision. Compared to black-box models, it is easier for users to trust models with explainability [171, 239].

5. *Knowledge discovery*: artificial intelligence algorithms are designed to learn and extract implicit structures and patterns from the data. These structures, in turn, can be seen as a form of knowledge acquired by the algorithm [70, 103]. It is not efficient to make use of these models only to generate predictions while ignoring the implicit knowledge captured during the learning process.

Explanation allows it to extract and formalize this knowledge captured by models [107].

6. *Causality*: statistical machine learning is limited to learning associations between features and outputs based on a large amount of data, and cannot easily discover cause-to-effect relationships that are essential in certain critical applications [156]. However, explainable AI techniques can be used to discover some cause-to-effect relationships by leveraging the data and the model [169]. Conversely, the results obtained from causal inference can also be used to validate the model with the aid of XAI [193].

5.1.2 Explanations and explanation methods

Desirable properties of explanations

As mentioned before, explanations are subjective and context-dependent, making it difficult to define what constitutes a good explanation. Here, from the perspective of users and from a social psychological standpoint, we explore some properties that seem to be desirable for explanations to be of practical use. The importance of each property may vary according to the use case.

1. *Ease of understanding*: the most critical property of explanation is arguably comprehensibility. An explanation that cannot be understood is meaningless to its receiver. Different kinds of explanations may have different levels of comprehensibility [141]. For example, for ordinary users, IF-THEN rules, sample-based explanations, and feature importance are easier to understand than mode-internals-based explanations. On another hand, due to the limitations of their cognitive abilities, people often do not desire to obtain complex, thorough explanations of an event but tend to select a few main causes [199]. As a consequence, explanations provided for a prediction should refer to several features, rather than all of them [109, 216, 217].

2. *Actionability*: explanations should provide actionable insights or recommen-

dations based on which users can act [17]. They should guide users in taking appropriate actions based on the explanation. For example, when a counterfactual explanation suggests modifications of the feature vector, the selected features should be actionable (and the corresponding modifications should be reasonable).

3. *Fidelity*: it refers to how accurately and faithfully the provided explanations reflect the actual behavior and decision-making process of a machine learning model being explained. If the explanations are faithful to the model's internal workings, users can better understand why a certain prediction was made and can judge whether the reasoning of the model is sound and aligned with their expectations, thus, helping users build trust in the outputs of the model [176].

4. *Compatibility*: due to the effect of "confirmation bias", humans may ignore explanations that are inconsistent with their prior beliefs and knowledge [151]. Therefore, explanations should be compatible with such prior beliefs. In machine learning, it is extremely important, to avoid users perceiving the model as behaving strangely and thus do not trust it. However, integrating such prior beliefs into the generation of explanations is very difficult due to the diversity of prior beliefs among users [105].

5. *Social interaction*: the explainer providing explanations to the explainee is an interactive process of a social nature, in which knowledge is transferred [140]. This social character requires that explanations of different extents be provided to different people. In machine learning, attention needs to be paid to the characteristics of the audience group and the social context [32].

Taxonomy of explanation methods

The classification of explanation methods can vary depending on the criteria used, as outlined in [141]. We can nevertheless point out three general perspectives:

1. *Intrinsic vs. Post-hoc*: intrinsic explainable models are inherently interpretable by design, which means that their internal mechanisms can be easily under-

stood by a human. These models, such as decision trees and linear regression, can provide transparent explanations about how they arrived at a particular decision or prediction. Post-hoc explanation methods, on the other hand, are used to provide explanations for machine learning models that are not inherently interpretable. These latter methods analyze the input and output behavior of the model after training and try to understand its decision-making process.

2. *Model-agnostic vs. Model-specific*: explanation methods that are designed to provide insights into a particular AI model are referred to as model-specific. Intrinsic explanation methods, as discussed earlier, are all model-specific, since they are tailored to the unique structure and properties of the model under consideration. In contrast, model-agnostic explanation methods can be used for any model. These latter methods generally examine the association between input and output features but do not have access to the internal workings of the model.

3. *Global vs. Local*: global model explainability refers to the ability to explain the overall behavior of a model. This typically involves understanding the key factors or features that are considered by the model to make predictions. Global explanations provide a high-level understanding of the model decision-making process, but they do not necessarily provide insight into the specific reasons why a particular prediction was made for a given input. Local model explainability, on the other hand, focuses on providing explanations to individual model predictions. Such explanations are particularly useful when a model prediction differs from what a human would expect. Local explanations can help identifying the specific features that are driving the model prediction for a particular input, which can be helpful in identifying errors or biases in the model.

Review of explanation methods

Apart from these general characteristics, the nature of the explanation can vary from one method to another. We list the following different kinds of explanation methods.

1. *Model internals*: as mentioned above, for intrinsically explainable models, the model internals are often used as explanations, such as the weight coefficients learned by linear models, the decision paths of decision trees, the if-then statements of rule-based models, etc. For convolutional neural networks, the learned high-level internal features and concepts in the hidden layers can be displayed as explanations for their behaviors [148, 149, 152, 153].

2. *Feature importance*: this is the most common kind of explanation, where each feature of the input space is associated with a measure of its contribution, often referred to as feature importance. For inherently explainable models, such as linear regression and logistic regression, the learned weights can be seen as a direct measure of feature importance. For decision trees, the importance of a given feature can be computed as the average decrease in the impurity (such as the Gini index) of the nodes where the feature is selected as the split feature, weighted by the proportion of the sample number in the nodes to the total number of samples [24, 122]. For tree ensembles, feature importance is computed as the average or weighted average feature importance across trees. As a post-hoc explanation method, the *Permutation Feature Importance* (PFI), which can be seen as a sensitivity analysis approach, evaluates the feature importance by the difference between the baseline performance metric (often the accuracy) and the one obtained from the same data set with one permuted column [25, 73]. At the local level, LIME (*Local Interpretable Model-agnostic Explanations*) [173] and SHAP (*SHapley Additive exPlanations*) [128] are two popular feature importance evaluation methods for single query instances. LIME learns a sparse linear model based on instances sampled around the query instance to approximate the local classification boundary and uses the

learned weights as feature importance measures. SHAP starts from a completely different idea based on game theory and identifies the importance of a feature with its contribution to the predictions. Evaluating the interactions among features can also be interesting to explain model outputs [78, 127].

3. *Feature visualization*: the *Partial Dependence Plot* (PDP) shows the marginal effects of one (or two) features on the predictions of the model [77]. In the single-feature PDP, the x-axis corresponds to the variation of the feature X^j that needs to be explained. For a given x-value of X^j, the corresponding y value is the average across all predictions obtained by replacing the value of X^j of all samples with the given x-value. However, PDP depends on the assumption that features are independent. *Accumulated Local Effects plots* (ALE) are a fast and unbiased alternative to PDPs [12], which calculate the effect of a feature based on the conditional distribution of the features to eliminate the effect of correlated features. To the side of local explanations, *Individual Conditional Expectation plots* (ICE) display one curve for each instance, indicating how the prediction changes as a feature changes while the others are kept fixed, which is intuitive and easy to understand [83]. It is equivalent to the PDP calculated on a single instance. In deep learning, especially for image classification, some pixel-level importance visualization methods have also been proposed. *Saliency maps* [183, 228], *feature inversion* [63], and *class activation mapping* [178, 237] all evaluate the importance of a pixel by mapping the gradient information of a neural network to the input pixels through the back-propagation algorithm. These methods intuitively indicate to the user which part of the image determines the classification result of the image.

4. *Example-based explanation methods*: following the philosophy of "similar questions, similar answers", *prototypes* are used as a global model-agnostic explanation method that aims to approximate the entire data distribution by selecting a small number of representative samples from the training data [111, 192]. For a query instance, the nearest prototype is used as the explanation. A concept similar to prototypes is *influential instances*. If deleting one of the training

instances would have an impact on the model, the instance is said to be influential. It can be found through deletion diagnosis [37] or influence functions [114]. Another well-known local case-based explanation technique consists in producing *counterfactual explanations*. A counterfactual explanation describes a causal relationship (if some cause had not happened, then the result would have been different as well). In counterfactual explanations, the causes are the feature values and the result is the model output. The counterfactual explanation of a model output describes the minimum change in feature values so as to make the prediction change to a predefined one [209]. A large number of model-specific and model-agnostic methods for generating counterfactuals can be found in the survey of Guidotti [86]. A concept akin to counterfactual examples is that of *adversarial instances*, which differ only slightly (imperceptibly to humans) from the real ones [20]. In general, these differences from intentional perturbations are used to attack and fool the model. However, in model diagnosis, they are mainly used to improve robustness and increase the safety.

5. *Simplified models*: when the structure of the model is too complex, it is very difficult to understand the decision process as a whole. An effective way to solve this problem is to use low-complexity interpretable models, such as linear models, decision rules, decision trees, shallow neural networks, etc., to approximate the original model. For tree ensembles, a *rule extraction algorithm* is usually applied to select the rules with an appropriate length, a high coverage, and high accuracy from all the decision rules of the trees to form a decision rule model [18, 22, 52, 92, 133]. For neural networks, *model distillation* is often adopted, the core idea of which is to use the original model to label the data, using which a simple model (called surrogate model) can be trained [29, 40, 80, 95]. However, this methodology has an obvious pitfall: the simplified model may not inherit the properties and performance of the original model.

5.2 Explanations for random forests

A random forest is an efficient and highly accurate classifier. However, its integration of numerous unpruned decision trees renders it a "black box" model, thereby sacrificing explainability. In the past years, extensive research efforts have been devoted to enhancing the explainability of random forests and their outputs. In this section, we provide a review of commonly used explanation methods for random forests. Fig. 5.1 provides a visual representation of these methods that are categorized based on model dependencies, scope of explanation, and types of explanations.

5.2.1 Model-agnostic explanation methods

As a machine learning model, random forests can benefit from model-agnostic explanation methods. In terms of global explanations, we consider two categories: feature importance and model simplification. Section 5.1.2 already covered methods of the first kind, such as Permutation Feature Importance (PFI) [25], Partial Dependence Plots (PDP) [77], Individual Conditional Expectation (ICE) [83], and Accumulated Local Effects (ALE) [12], which will not be covered again here. Model simplification involves approximating a less interpretable model with interpretable ones, such as decision trees [18]. The main idea is to replace the original labels of the training set with the predictions given by the model that need to be explained and build a new training set used to train the simplified model.

Local explanation methods can be categorized into feature importance approaches, surrogate models, prototypes, and counterfactual explanations. SHAP [128], LIME [173], and prototypes [111] were reviewed in Section 5.1.2. In the case of surrogate models, LEAFAGE [6] provides explanations similar to LIME and considers the "most similar examples" as additional explanations to predictions. The Counterfactual Local Explanations via Regression (CLEAR) [214] method employs counterfactual samples to build a local surrogate model with increased faithfulness. Anchors [174] utilizes reinforcement learning and graph search techniques to find an if-then rule that "anchors" a particular prediction, i.e., changes in features that are not

involved in the rule have no influence on the prediction. The Local Rule-Based Explanation (LORE) [87] method employs genetic algorithms for sampling around the query instance and trains a decision tree based on the labels given by the original model on the sampled instances to proxy the local behavior of the model.

Overall, model-agnostic counterfactual generation methods can be classified into four categories: methods that search for counterfactual samples in the training data, methods that generate such samples, methods based on local surrogate models, and methods that involve solving optimization problems. Searching for counterfactuals in the training data is efficient and faithful. The Feasible and Actionable Counterfactual Explanations (FACE) [162] method searches for counterfactual samples in the training set that satisfy density and user constraints. DisCERN [218] searches for the nearest counterfactual in the training set and replaces the feature values of the query instance one by one with the values of this nearest counterfactual based on feature importance until the desired prediction is obtained. Instead of using the training set, the Growing Sphere method [116] proposes a data-agnostic sample generation approach and feature selection process to find sparse counterfactuals. However, these methods often generate counterfactuals that are far away from the query instance. CLEAR and LORE generate counterfactual examples using local linear models and decision trees trained on generated samples, respectively. A notable drawback of methods based on local surrogate models is that the generated "counterfactual" samples may not necessarily be classified into the desired class by the original model, i.e., fidelity is low. To address these issues, optimization-based approaches are employed. The Model-Agnostic Counterfactual Explanation (MACE) [108] method solves a series of satisfiability problems to generate counterfactual examples, representing the distance function and the model to be explained as logical formulae. The Distribution-Aware Counterfactual Explanation (DACE) [106] method formulates the counterfactual generation problem as a mixed-integer optimization problem, considering the correlation between features using the Mahalanobis distance and reducing the risk of generating outlier counterfactuals using local outlier factor [27]. It should be remarked that methods based on optimization problems are not efficient and even not applicable to large-scale tree ensembles.

5.2.2 Model-specific explanation methods

Due to the good performances and special structure of random forests, numerous specifically designed explanation methods have been proposed, both global and local.

Global explanation methods

Feature importance and feature visualization are indeed important global explanation methods for random forests. Feature importance is typically based on the special characteristics of trees, such as the impact of splits on reducing impurity [25, 125, 154, 159] and the frequency of feature usage for splitting [68]. In some applications, the importance of each value of each feature is considered, in addition to the overall feature importance [69]. In terms of visualization, different methods exist to represent the relationship between input features and the random forest output. Some examples include multidimensional scaling [123], weighted networks[84], self-organizing maps [161], mapping structure [213], and summarizing decision paths [236]. These visualization techniques aim to provide insights into the relationships and patterns within the random forest model and its predictions.

Feature importance helps in understanding the relative significance of different features in the model decision-making process. It allows us to identify the key variables that have the most impact on the model predictions. Visualization, on the other hand, provides a visual representation of the relationships between features and the predictions made by the random forest. Through visualizations, the users can gain a deeper understanding of how the model operates and identify patterns or trends in the data. They both play a crucial role in explaining the model and in enhancing its interpretability.

Model simplification techniques, as another form of global explanation, can be divided into two categories in the context of random forests: constructing a single surrogate decision tree or extracting rules to approximate the original random forest. There are broadly three methods for constructing a surrogate decision tree from a given random forest. The first one is based on sample approximation: the surrogate

decision tree is trained using the original training samples or samples generated through sampling, with labels provided by the random forest [26, 204]. The second one uses the decision paths of the decision trees as training samples to train the surrogate decision tree [177, 207]. The last one constructs the surrogate decision tree using the splits in the decision trees as training samples [205, 238]. In contrast, extracting rules from the original random forest is more straightforward. It involves selecting a small subset of decision paths from all decision trees in the forest as a partition of the entire sample space based on certain criteria and processes, such as quadratic programming [137], sparse linear programming [133], Bayesian model selection [92], or ranking decision paths based on their coverage, accuracy, and complexity [7, 22, 52, 160].

These model simplification methods aim to provide a more interpretable representation of the random forest by capturing its decision-making process through a single decision tree or a set of extracted rules. Simplifying the model allows to gain insights into the underlying patterns and rules that contribute to the random forest predictions, thus enhancing its interpretability and facilitating communication of its behavior to stakeholders and end-users.

Local explanation methods

The local explanation methods for random forests are mainly case-based, e.g., prototypes and counterfactuals. In random forests, the selection of prototypes relies on different similarity measures, such as the proportion of trees in the random forest in which two samples end up in the same leaf [192], the tree-ensemble kernel function based on decision paths, and leaf outputs.

There are also different methods for generating counterfactual explanations [28]. The simplest and most straightforward approach is to generate samples close to the query instance in each leaf of the decision trees, and then determine their predictions using the random forest [196]. The closest sample that meets the desired prediction is selected as the counterfactual explanation. Alternatively, the random forest can be transformed into a network by replacing the split in each node with

a differentiable sigmoid function, and counterfactuals can be obtained by solving an optimization problem [126]. The main drawback of these two methods is that they may not generate the proximal counterfactual explanations. Methods based on integer linear programming [42] and mixed-integer programming [155] convert the tree structure into integer constraints that can be used to compute counterfactuals, which theoretically guarantees the generation of the proximal counterfactuals. However, these methods suffer from low efficiency and are not suitable for large-scale random forests composed of a large number of trees with a significant depth. Another solution is to search counterfactuals around the query instance, with the search range determined by the initialized counterfactual example that is typically obtained from the training set or using heuristic algorithms [21, 71]. However, this approach also has a drawback: for samples far from the decision boundary, the search range can be very large, resulting in low efficiency.

In summary, each counterfactual generation method has its own advantages and limitations. The choice of a method should consider practical factors such as proximity requirements and computational aspects (which mainly depend on the number of trees in the random forest as well as on their depth).

5.3 Conclusion

In this chapter, we have reviewed the concepts of explainable artificial intelligence, and discussed its importance, and desirable properties of explanations We have also presented the main explanation methods according to the type of provided explanations.

Additionally, we have presented different ways of providing explanations for random forests, which is the classification technique we mainly considered in our works. We have classified explanations based on their dependency on a specific classification algorithm, their ability to provide global or local explanations, and the types of explanations provided. Furthermore, we have analyzed the advantages and disadvantages of each type of explanation method. An attempt to summarize all these

methods with pointers to the main references is presented in Fig. 5.1.

In the next chapter, based on the aforementioned tools in XAI, we will introduce our proposed framework to provide diverse explanations (mainly counterfactuals) for indeterminate predictions made by cautious random forests. This framework will encompass both local and global feature importance assessments to guide the generation of meaningful counterfactual samples, i.e., instances that elucidate how indeterminate predictions can be turned into desired determinate ones.

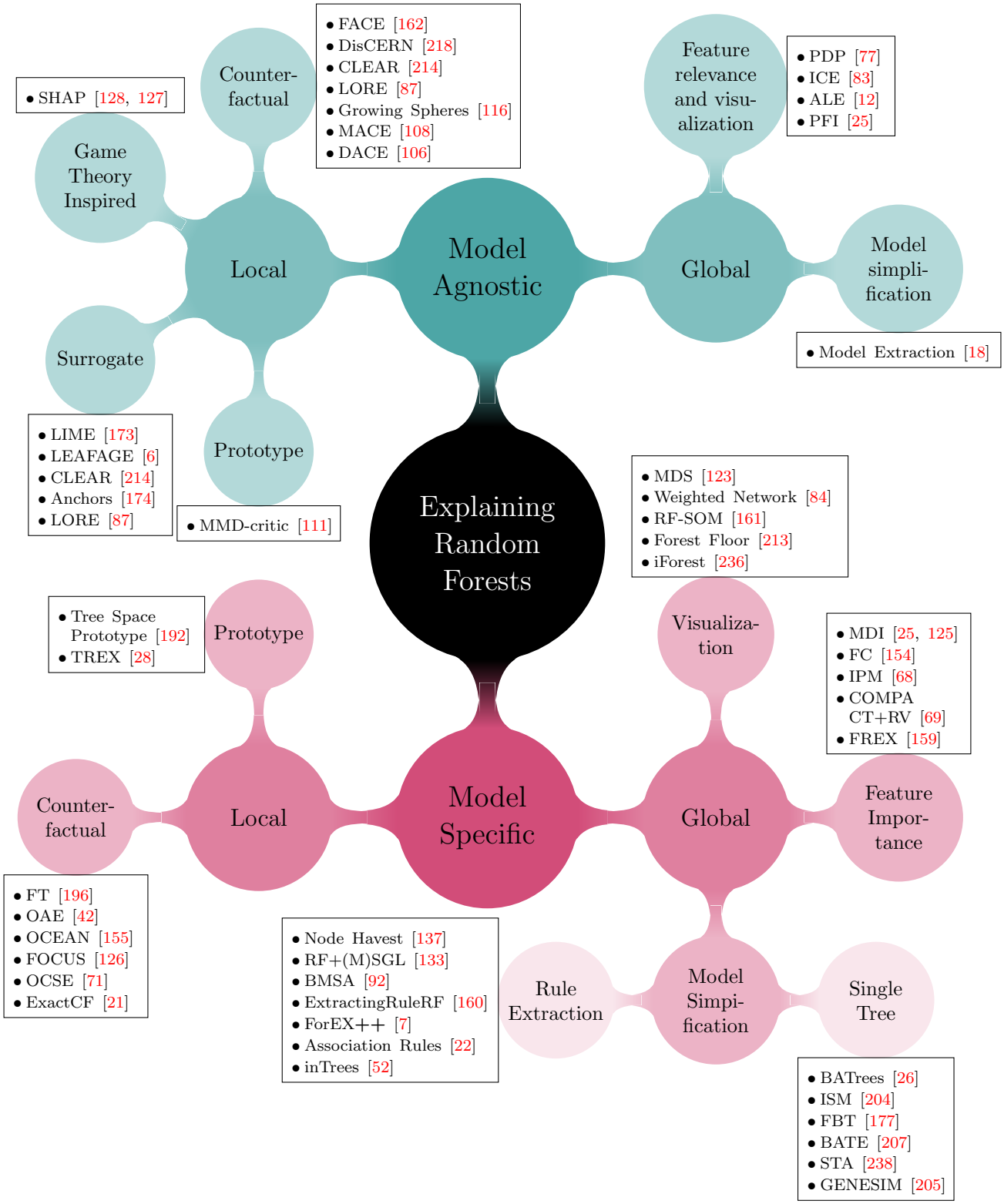


Figure 5.1: Mind-map of methods for explaining random forest.

Chapter 6

Resolving indeterminacy via counterfactuals

This chapter focuses on explaining the outputs of cautious random forests that can produce set-valued predictions when making a precise decision seems to entail risks. However, making imprecise predictions carries a cost, as resolving indeterminacy typically requires further analysis and manual intervention [231]. More precisely, in the context of cautious classifiers, it seems crucial for users to understand, for an input instance, what leads to an imprecise decision, and what could be done to turn it into a determinate one. Such questions fall under the emerging topic of eXplainable Artificial Intelligence (XAI). This general objective is similar to that of CLUE (Counterfactual Latent Uncertainty Explanations) [11], developed for explaining uncertainty estimates in differentiable probabilistic models, like Bayesian Neural Networks.

In this chapter, we propose to use counterfactual examples to explain the indeterminate predictions made by a cautious random forest. These generated counterfactual examples allow us to identify the minimal modifications on some feature values so as to obtain a desired determinate prediction and resolve the indeterminacy of the prediction. To address this problem, we first define counterfactual explanations for cautious predictions and then propose a model to generate counterfactual examples for indeterminate instances.

Our counterfactual generator is based on Blanchard's work, which divides the feature space into "pure regions" where all data points belong to one class and then uses branch-and-bound strategy in a search tree to find the closest counterfactual samples for a given query instance [21, 206]. Compared to the initial algorithm of Blanchard, we have improved the counterfactual example generation process in the following ways:

1. we simplified the decomposition of the feature space into "pure regions";
2. we introduced different distance measures that consider the scale difference of features to search for the closest counterfactual examples [209];
3. we improved the initialization of counterfactual examples to narrow the search region [231] and integrated feature constraints to ensure the actionability of

the generated counterfactual examples[104, 105];

4. we showed that the generated counterfactuals are consistent with the query instances, e.g., in terms of feature types;

5. we introduced a plausibility metric in the counterfactual generation process, so that the generated counterfactuals are as plausible as possible;

6. we proposed to use feature importance to determine the order of features and their levels in the search tree to accelerate the counterfactual example generation process.

This chapter is organized as follows. Section 6.1 provides the definition of counterfactual explanation and its application in cautious classification problems. Then, we describe our proposed approach of counterfactual example generation in Section 6.2: we explain how to generate proximal, plausible, and actionable counterfactuals for indeterminate instances in a given cautious random forest model and illustrate the effectiveness of our proposed approach with experiments. In Section 6.3, we show how to use feature importance to accelerate the counterfactual example generation process and present the experimental results about the acceleration impacts.

6.1 Counterfactual explanations for indeterminate predictions

The indeterminacy (cautiousness) of a prediction is designed to reduce the risk of making wrong decisions. However, it is natural for a user to ask for information on how indeterminacy can be resolved so that a precise decision is made. In real-world applications, leaving the entire responsibility of resolving the indeterminacy to the user would result in the decision process being very demanding. Therefore, we provide counterfactuals with known classes for each instance with an indeterminate

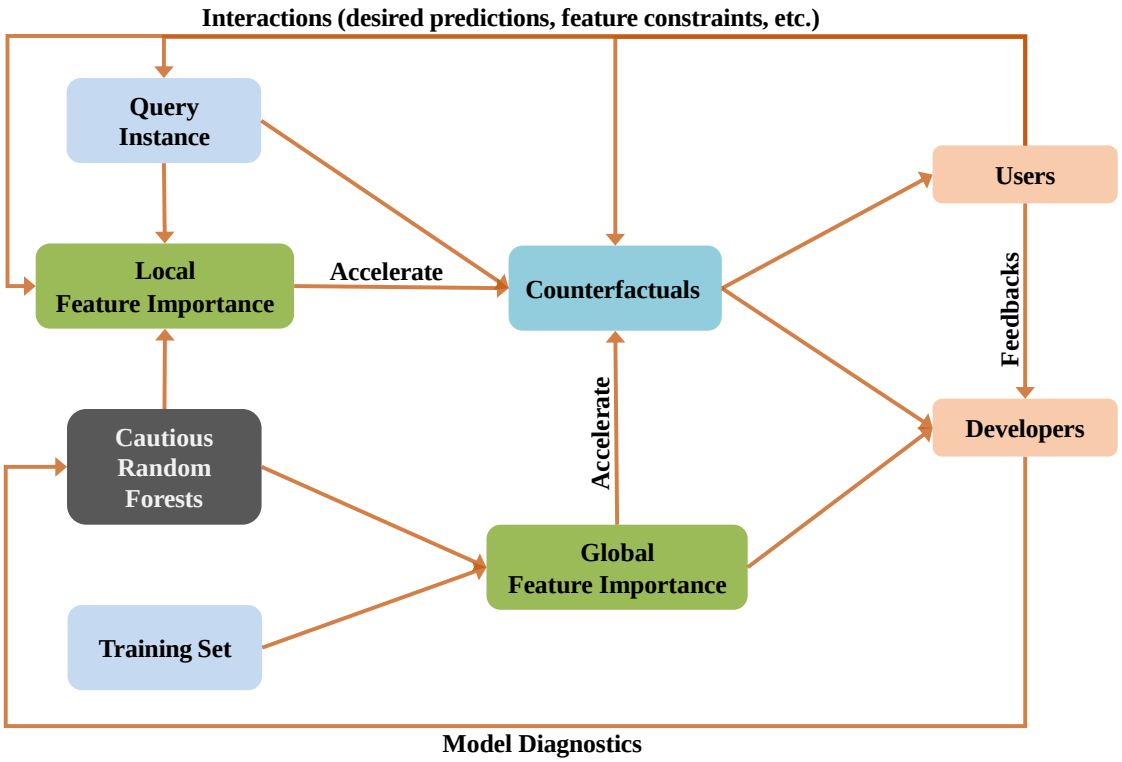


Figure 6.1: Flowchart of explanations for cautious random forests.

prediction, so that users know how to modify the values of some features to resolve indeterminacy in either way.

6.1.1 Counterfactual explanations for predictions

In the context of causality, a counterfactual explanation describes a causal relation in the form of "if a contradictory cause had occurred, a different event would have happened" [156]. In XAI, a counterfactual explanation for a prediction describes the smallest change to the feature values that turns the prediction into a predefined class [141]. For a given classifier $\boldsymbol{h}$, an instance $\boldsymbol{x}$ and a desired prediction $y' \neq \boldsymbol{h}(\boldsymbol{x})$, the counterfactual sample $\boldsymbol{x}'$ in the input space $\mathcal{X}$ can be described as:

$$\boldsymbol{x}' = \arg\min_{\boldsymbol{z}\in\mathcal{X}} d(\boldsymbol{x}, \boldsymbol{z}), \text{ s.t. } \boldsymbol{h}(\boldsymbol{z}) = y', \ \boldsymbol{h}(\boldsymbol{x}) \neq y', \tag{6.1}$$

where $d(\cdot)$ is a proximity measure that is usually based on a distance. As argued in [209], the Manhattan distance weighted with the inverse median absolute deviation

(MAD) is recommended. It is defined as follows:

$$d(\boldsymbol{x}, \boldsymbol{z}) = L_1^{mad}(\boldsymbol{x}, \boldsymbol{z}) = \sum_{j=1}^{M} \frac{|x^j - z^j|}{MAD^j}, \tag{6.2}$$

where x^j denotes the value of $\boldsymbol{x}$ on feature X^j and MAD is calculated on training set as

$$MAD^j = \underset{i\in\{1,\ldots,N\}}{median}(|x_i^j - \underset{l\in\{1,\ldots,N\}}{median}(x_l^j)|). \tag{6.3}$$

This distance captures scale difference among features and is more robust to outliers. Due to the properties of the L_1 norm, it results in sparser counterfactual samples (less features need to be modified). An alternative is the Euclidean distance weighted with the inverse standard deviation (STD) to adjust scale difference of the input features:

$$d(\boldsymbol{x}, \boldsymbol{z}) = L_2^{std}(\boldsymbol{x}, \boldsymbol{z}) = \sqrt{\sum_{j=1}^{M} \frac{(x^j - z^j)^2}{STD^j}}. \tag{6.4}$$

6.1.2 Counterfactuals in binary cautious classification

In our binary cautious classification setting with $\Omega = \{c_1, c_2\}$, we define the problem as the search of counterfactuals $\boldsymbol{x}^{c_1}$ and $\boldsymbol{x}^{c_2}$ for a given instance $\boldsymbol{x}$ such that $\boldsymbol{h}(\boldsymbol{x}) = \{c_1, c_2\}$ (indeterminate):

$$\boldsymbol{x}^{c_1} = \arg\min_{\boldsymbol{z}\in\mathcal{X}} d(\boldsymbol{x}, \boldsymbol{z}) \text{ s.t. } \boldsymbol{h}(\boldsymbol{z}) = \{c_1\}, \tag{6.5a}$$

$$\boldsymbol{x}^{c_2} = \arg\min_{\boldsymbol{z}\in\mathcal{X}} d(\boldsymbol{x}, \boldsymbol{z}) \text{ s.t. } \boldsymbol{h}(\boldsymbol{z}) = \{c_2\}, \tag{6.5b}$$

where the distance is defined by Eq. (6.2) or Eq. (6.4). The utility of these counterfactual examples can be summarized as follows:

1. they help determining the minimal modifications needed to obtain a desired precise prediction;
2. they help identifying the closest class to an indeterminate instance;
3. they facilitate understanding the differences between two classes.

Example 6.1 (Counterfactuals for indeterminate predictions)**.** *Based on the definition Eq.(6.5), we provide here two examples (on tabular and image data) to illustrate our counterfactual procedure for indeterminate predictions.*

The first example focuses on the Pima dataset that can be used to predict whether a patient has diabetes or not, based on various measurements: Pregnancies (PGs): number of times pregnant; Glucose; Blood Pressure (BP); Skin Thickness (ST); Insulin: 2-Hour serum insulin (mu U/ml); BMI: body mass index; Diabetes Pedigree Function (DPF); Age. The class is $c_1 = 0$ for a non-diabetic, $c_2 = 1$ for a diabetic. Here, Age, number of pregnancies, DPF values, and Skin Thickness are difficult to change (considered as protected features), while Glucose, Insulin, BMI, and blood pressure are actionable (mutable) features.

Table 6.1: Examples of counterfactual explanations from Pima dataset.

	PGs	Glucose	BP	ST	Insulin	BMI	DPF	Age
$\boldsymbol{x}_1$	0	165	90	33	680	52.3	0.427	23
$\boldsymbol{x}_1^0$	0	**154**$\downarrow$	90	33	680	**47.7**$\downarrow$	0.427	23
$\boldsymbol{x}_1^1$	0	**166**$\uparrow$	90	33	680	52.3	0.427	23
$\boldsymbol{x}_2$	1	122	90	51	220	49.7	0.325	31
$\boldsymbol{x}_2^0$	1	**121**$\downarrow$	90	51	**128**$\downarrow$	**49.05**$\downarrow$	0.325	31
$\boldsymbol{x}_2^1$	1	**127**$\uparrow$	90	51	220	49.7	0.325	31

In Table 6.1, the query instance $\boldsymbol{x}_1$ corresponds to a non-diabetic patient (actual label). First, note that $\boldsymbol{x}_1$ is close to being classified as diabetic since the counterfactual $\boldsymbol{x}_1^1$ of this class is very close. Note that this demonstrates how the cautious random forest can help managing the uncertainty arising around the classification boundary. Second, the non-diabetic counterfactual $\boldsymbol{x}_1^0$ suggests a possible way to maintain a healthy condition, i.e., reducing BMI and Glucose level. The query instance $\boldsymbol{x}_2$ corresponds to a diabetic patient (real label). The Glucose feature is the key to resolve imprecision since we can get a correct prediction (diabetic) by only modifying its value. On the other hand, to obtain the non-diabetic counterfactual $\boldsymbol{x}_2^0$, an important decrease of Insulin is needed, which is coherent with the fact that high 2-hour serum insulin levels are common for type-II diabetic patients.

The second example is based on the MNIST dataset, which is a large database of handwritten numbers containing about 60,000 training cases and 10,000 test cases. Samples corresponding to classes "4" and "9" are selected to construct a binary classification problem. In the dataset, we can find that some of the pictures are ambiguous, as it is difficult to tell whether the number is "4" or "9". The generation of counterfactuals of an ambiguous query instance helps in understanding which parts of the image are responsible for the indeterminacy of the decision. This point is illustrated using two instances drawn in Figure 6.2. We can see how the two indeterminate instances (center) could be modified to be determinately classified as either a "4" or as a "9", and that these modifications make sense.

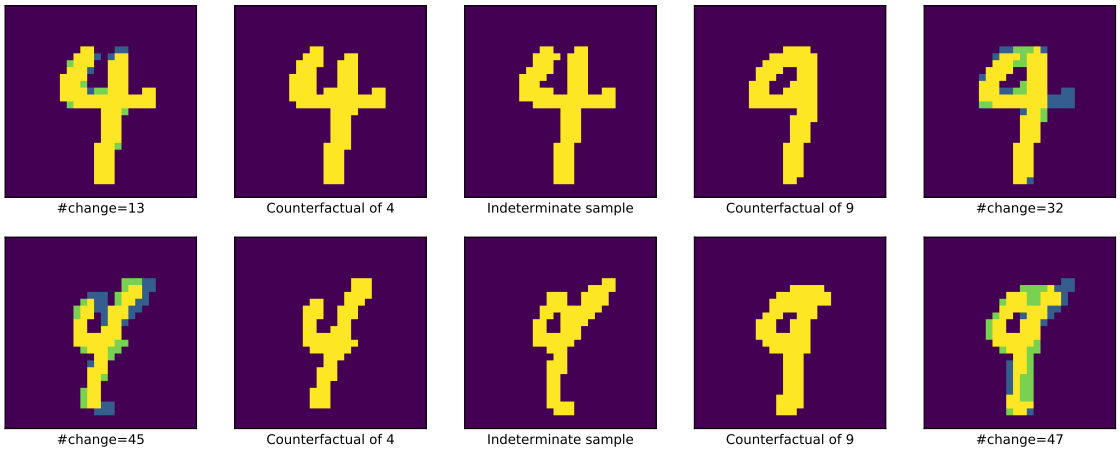


Figure 6.2: Examples of indeterminate numbers (center) and corresponding counterfactuals of class 4 (left) and 9 (right). Left- and right-most images display pixels to be added (green) and to be deleted (blue) in order to obtain the counterfactual.

6.1.3 Counterfactuals in multi-class cautious classification

In the multi-class setting, we can straightforwardly generalize the problem of finding a counterfactual example with a desired precise prediction $c_k \in \Omega$ for a given instance $\boldsymbol{x}$ with an indeterminate prediction $\boldsymbol{h}(\boldsymbol{x}) = A \subseteq \Omega$ and $|A| > 1$, as follows:

$$\boldsymbol{x}^{c_k} = \arg\min_{\boldsymbol{z}\in\mathcal{X}} d(\boldsymbol{x}, \boldsymbol{z}) \text{ s.t. } \boldsymbol{h}(\boldsymbol{z}) = \{c_k\},\ c_k \in \Omega, \tag{6.6}$$

where the distance is defined by Eq. (6.2) or Eq. (6.4).

However, the problem of generating counterfactual explanations with multi-class cautious classifiers appears to be potentially much richer. We summarize some ideas and discussions based on the so-called set-valued counterfactuals proposed in [82]. We denote the indeterminate prediction for $\boldsymbol{x}$ as $\boldsymbol{h}(\boldsymbol{x}) = A$ (which may be precise) and the desired prediction as $A' \subseteq \Omega,\ A' \neq \emptyset$. There are several ways to generalize the problem:

- case 1: find $\boldsymbol{x}'$ such that an exact target prediction is reached, i.e., $\boldsymbol{h}(\boldsymbol{x}') = A',\ A' \neq A$ (the problem defined in Eq. (6.6) is a special case);
- case 2: find $\boldsymbol{x}'$ such that $\boldsymbol{h}(\boldsymbol{x}') \subset A'$ (restrict the possible classes to A', i.e., refine the set of possible classes);
- case 3: find $\boldsymbol{x}'$ such that $\boldsymbol{h}(\boldsymbol{x}') \supset A'$ (enlarge the set of possible classes to A').

If the purpose is to reduce the indeterminacy of the current prediction, the problem can be formalized as finding the closest instance $\boldsymbol{x}'$ such that $\boldsymbol{h}(\boldsymbol{x}') \subset \boldsymbol{h}(\boldsymbol{x})$, for which finding a determinate counterfactual example ($|\boldsymbol{h}(\boldsymbol{x}')| = 1$) is a special case.

The above points may be further examined and discussed, especially when it comes to defining set-valued counterfactual explanations according to the application considered, and addressing how to generate them. In this thesis, we focus on the counterfactuals in binary cautious classification defined in Eq. (6.5).

6.2 Algorithmic resolution for counterfactual generation

Solving the counterfactual search problem defined in Eq. (6.5) is quite complex. A common approach is to construct a loss function without constraint and minimize it by a gradient-based optimization method. Wachter et al. [209] proposed the following optimization problem:

$$\boldsymbol{x}' = \arg\min_{\boldsymbol{z} \in \mathcal{X}} \lambda \mathcal{L}(\boldsymbol{h}(\boldsymbol{z}), y') + d(\boldsymbol{x}, \boldsymbol{z}), \tag{6.7}$$

where λ is usually large to guarantee that the counterfactuals obtained are valid (as ensured by minimizing the first term) while being close to the query instance (second term).

Besides validity and proximity, counterfactual explanations are often required to satisfy the following supplementary properties: sparsity (only a few features should be changed), plausibility (consistency with the underlying data distribution), diversity (it should make it possible to generate several distinct counterfactuals), and actionability (it should lead to reasonable feature changes, both in terms of feature choice and modification magnitude) [34, 86, 206]. These properties make counterfactual explanations user-friendly, i.e., contrastive, selective, and compatible. To address these issues, Dandl et al. combined terms accounting for validity, proximity, sparsity, and plausibility into a single loss function and solved a multi-objective optimization problem to generate good counterfactuals [43], while Mothilal et al. plugged the diversity term in Eq. (6.7) to generate diverse counterfactuals [115, 144]. Moreover, Jeyasothy et al. proposed a general framework to integrate user knowledge into post-hoc explanations [104].

However, for models without gradient information such as tree-based models, generating counterfactual explanations based on loss functions is difficult. In this case, either model-agnostic counterfactual explanation methods or counterfactual approaches designed specifically for tree-based ensembles can be applied—we refer the reader to Section 5.2 for a review. However, these methods either cannot generate valid and proximal counterfactuals or are inefficient. Moreover, almost no method considers the issues of plausibility and actionability of counterfactuals in tree ensembles.

To address these issues, we propose an approach based on a branch-and-bound algorithm [21] for generating counterfactuals in cautious random forests, taking into account the desirable properties (validity, proximity, plausibility, actionability, and efficiency) of counterfactual explanations. The main flowchart is shown in Fig. 6.3.

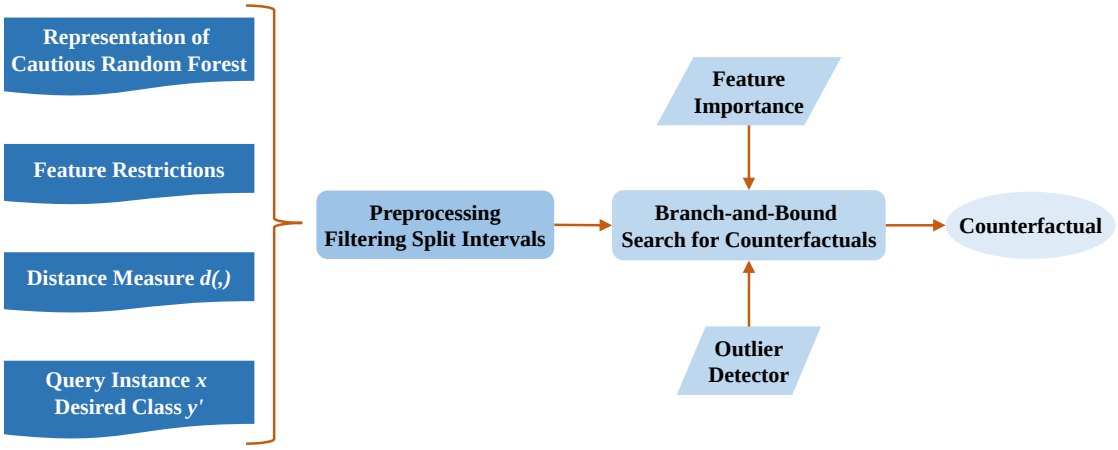


Figure 6.3: Overview of counterfactual generation framework for cautious random forest.

6.2.1 Representation of cautious random forests

The first step of our branch-and-bound method consists in converting each path from the root node to the leaf node in the tree structure into a multi-dimensional box form. Let $\mathbf{H} = \{\boldsymbol{h}_t,\ t = 1, \ldots, T\}$ be a cautious random forest consisting of T imprecise decision trees, trained on a binary classification dataset of M features, i.e., $\mathbf{X} \in \mathbb{R}^M$ and $\mathrm{Y} \in \Omega = \{c_1, c_2\}$.

For each imprecise tree $\boldsymbol{h}_t$, following the implementation of scikit-learn, we suppose all features to be numerical, and for each feature X^j, its domain is noted as $D^j = [\underline{D}^j, \overline{D}^j]$. In $\boldsymbol{h}_t$, each leaf corresponds to a partial region of the input space, which is determined by a series of split tests from the root to the leaf. For instance, a leaf determined by $(\mathrm{X}^1 \leq 4 \wedge \mathrm{X}^2 > 3 \wedge \mathrm{X}^1 > 2)$ can be represented as a region defined by a multi-dimensional box as $\{\mathrm{X}^1 \in]2, 4], \mathrm{X}^2 \in]3, \overline{D}^2]\}$. As we can see, if a feature is used several times for splitting in a root-leaf decision path, it can be summarized into a single interval. If a feature X^j is never used, it is represented as $]\underline{D}^j - \epsilon, \overline{D}^j]$, where ϵ is a small positive number to guarantee that $\underline{D}^j$ is included in the interval.

Formally, the decision path of leaf L_l is defined as

$$dp_l = \{]\underline{b}_l^j, \overline{b}_l^j],\ j = 1, \ldots M\}, \tag{6.8}$$

where $\underline{b}_l^j < \overline{b}_l^j$, $\underline{b}_l^j \in D^j \cup \{\underline{D}^j - \epsilon\}$ and $\overline{b}_l^j \in D^j$. In addition to its location, each leaf L_l is also associated with an estimated probability, or in our case the interval-valued probabilities defined by Eq. (3.1), denoted here as $ep_l = \{[\underline{p}_{lk}, \overline{p}_{lk}],\ k = 1, 2\}$. Therefore, $\boldsymbol{L}_t = \{\{dp_l, ep_l\},\ l = 1, \ldots, N_t\}$ is an equivalent representation of the imprecise decision tree structure $\boldsymbol{h}_t$, where N_t is the number of leaves of the tree. All leaves from different trees can be concatenated together, i.e., $\boldsymbol{L} = \bigcup_{t=1}^{T} \boldsymbol{L}_t$, and the total number of leaves is denoted as $N_L = \sum_{t=1}^{T} N_t$.

From the set of leaves $\boldsymbol{L}$, for each feature X^j, all of its corresponding split values can be extracted as follows:

$$SV^j = \{v_i^j,\ i = 1, \ldots, N_V^j\} = unique(\underline{b}_l^j, \overline{b}_l^j,\ l = 1, \ldots, N_L) \tag{6.9}$$

where N_V^j is the number of split values for feature X^j, and N_L is the total number of leaves in the forest. The split values in SV^j are then sorted in ascending order.

Based on split values, the domain D^j is split into $N_V^j - 1$ intervals, defined as

$$SI^j = \{]v_i^j, v_{i+1}^j],\ i = 1, \ldots, N_V^j - 1\}. \tag{6.10}$$

Hereafter, for the sake of simplification, $SI^j = \{]sv_i^j, ev_i^j],\ i = 1, \ldots, N_I^j\}$ is used as notation for split intervals, where N_I^j is the number of split intervals for feature X^j, and sv and ev respectively denote the start and end values. In this case, for a given instance $\boldsymbol{x}$ and the set of split intervals SI^j, the split interval of feature X^j that contains x^j can be found by using a location function:

$$SI_{\boldsymbol{x}}^j = Loc(x^j, SI^j) \text{ such that } x^j \in SI_{\boldsymbol{x}}^j. \tag{6.11}$$

Finally, we define a function to retrieve the set of leaves associated with a given split interval of feature X^j associated with $\boldsymbol{x}$ as follows:

$$\boldsymbol{L}|x^j = R(SI_{\boldsymbol{x}}^j, j, \boldsymbol{L}) = \{L_l : SI_{\boldsymbol{x}}^j \subseteq dp_l^j,\ l = 1, \ldots, N_L\},\ j = 1, \ldots, M. \tag{6.12}$$

The information of leaves (decision paths and estimated probabilities) and the set of leaves associated with each split interval are an equivalent representation of the trained cautious random forest because any partial region defined as $\{si^j : j = 1, \ldots, M, \ si^j \in SI^j\}$ is a minimum decision region (in which all samples have the same prediction) and their union spans the whole input space.

Computing the prediction associated with a given instance $\boldsymbol{x}$ can be done by applying the following steps:

1. computing the location of each feature value x^j by $SI^j_{\boldsymbol{x}} = Loc(x^j, SI^j)$;
2. retrieving the associated leaves of each feature via $\boldsymbol{L}|x^j = R(SI^j_{\boldsymbol{x}}, j, \boldsymbol{L})$;
3. computing the intersection $\boldsymbol{L}|x^j, \ j = 1, \ldots, M$ to obtain the set of leaves where $\boldsymbol{x}$ falls into the trees, i.e., $\boldsymbol{L}|\boldsymbol{x} = \cap \boldsymbol{L}|x^j, \ j = 1, \ldots, M$;
4. returning the prediction by applying the corresponding decision-making strategy to the estimated probability information in $\boldsymbol{L}|\boldsymbol{x}$.

Example 6.2 (Representation of cautious random forests)**.** *We consider an example of a cautious random forest consisting of two trees learned over a 2D input space, as displayed in Fig. 6.4. The information about the leaves is listed in Table 6.2.*

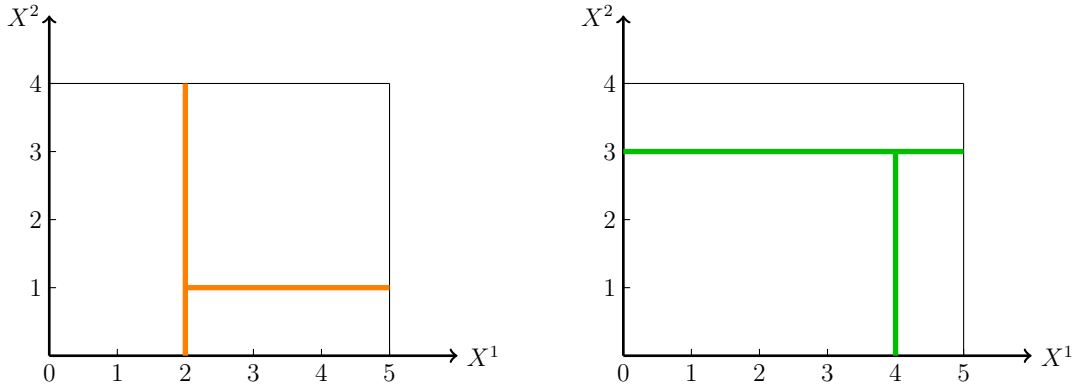


Figure 6.4: Example of two decision trees for the partition of the feature space.

Knowing that $D^1 = [0, 5]$ *and* $D^2 = [0, 4]$*, we can find that* $SV^1 = \{0 - \epsilon, 2, 4, 5\}$ *and* $SV^2 = \{0 - \epsilon, 1, 3, 4\}$*. Thus* $SI^1 = \{]0 - \epsilon, 2],]2, 4],]4, 5]\}$ *and* $SI^2 = \{]0 - \epsilon, 1],]1, 3],]3, 4]\}$*. Then, the leaves associated with each split interval are given in Table 6.3.*

Table 6.2: Example of leaves in a cautious random forest.

L	dp^1		dp^2	ep^1		ep^2
L_1	$]0-\epsilon, 2]$	,	$]0-\epsilon, 4]$	$[0.8, 0.9]$	,	$[0.1, 0.2]$
L_2	$]2, 5]$	,	$]0-\epsilon, 1]$	$[0.4, 0.6]$	,	$[0.4, 0.6]$
L_3	$]2, 5]$	,	$]1, 4]$	$[0.1, 0.2]$	,	$[0.8, 0.9]$
L_4	$]0-\epsilon, 4]$	,	$]0-\epsilon, 3]$	$[0.7, 0.8]$	,	$[0.2, 0.3]$
L_5	$]4, 5]$	,	$]0-\epsilon, 3]$	$[0.3, 0.4]$	,	$[0.6, 0.7]$
L_6	$]0-\epsilon, 5]$	,	$]3, 4]$	$[0, 0.2]$	,	$[0.8, 1]$

Table 6.3: Example of leaves associated with each split interval.

Feature X^1	Feature X^2
$R(]0-\epsilon, 2], 1, \boldsymbol{L}) = \{L_1, L_4\}$	$R(]0-\epsilon, 1], 2, \boldsymbol{L}) = \{L_1, L_2, L_4, L_5\}$
$R(]2, 4], 1, \boldsymbol{L}) = \{L_2, L_3, L_4, L_6\}$	$R(]1, 3], 2, \boldsymbol{L}) = \{L_1, L_3, L_4, L_5\}$
$R(]4, 5], 1, \boldsymbol{L}) = \{L_2, L_3, L_5, L_6\}$	$R(]3, 4], 2, \boldsymbol{L}) = \{L_1, L_3, L_6\}$

A test instance $\boldsymbol{x} = (1, 2)$*, is located in* $SI^1_{\boldsymbol{x}} =]0-\epsilon, 2]$ *and* $SI^2_{\boldsymbol{x}} =]1, 3]$*. Therefore, leaves associated with* $\boldsymbol{x}$ *are*

$$\boldsymbol{L}|\boldsymbol{x} = R(SI^1_{\boldsymbol{x}}, 1, \boldsymbol{L}) \cap R(SI^2_{\boldsymbol{x}}, 2, \boldsymbol{L}) = \{L_1, L_4\} \cap \{L_1, L_3, L_4, L_5\} = \{L_1, L_4\}.$$

According to the probabilities estimated in these leaves via Eq. (3.5), Eq. (3.6) and Eq. (3.7), the associated prediction is $\{c_1\}$*.*

6.2.2 Preprocessing

Assuming that each feature has N_I split intervals, a cautious random forest has a total of $(N_I)^M$ minimum decision regions. Searching for the best counterfactual sample is intractable when M is large.

However, the search procedure can be initialized using some simple methods, based on which some far-away regions can be filtered out. In addition, considerations

of actionability will also help to filter out some parts of the space. This is often the case in real applications, where some restrictions have to be applied to features, so as to guarantee actionability. Therefore, in this section, some preprocessing techniques are proposed to reduce the search region in the feature space and to satisfy possible feature restrictions.

Three kinds of restrictions on features can be considered: the value of the feature is immutable (it cannot be modified), or can be only increased or decreased. First, we remark that a counterfactual that satisfies all feature restrictions can be found or generated using simple heuristic methods to determine the upper bound on the distance between the query instance and its counterfactuals, noted as d_{sup}. One popular method is called the Minimum Observable (MO) approach, which searches in the training set for the closest counterfactual that satisfies feature restrictions. Alternatively, the One-Feature-Changed Counterfactual (OFCC) approach tries to vary the value of only one feature and keeps the remaining features unchanged to generate counterfactuals. Experimental results [231] showed that OFCC can generate closer counterfactual samples than MO, thus achieving a smaller initial distance d_{sup}.

Since the initial counterfactual sample satisfies feature restrictions, the distance of the closest counterfactual to $\boldsymbol{x}$ is no more than d_{sup}, which means that d_{sup} can be used to narrow down the search region in the feature space. Together with the feature restrictions due to actionability, split intervals of features can be filtered as follows:

1. if feature X^j has no restrictions, split intervals whose distance to x^j are larger than d_{sup} should be filtered out, i.e., the remaining split intervals are $SI^j_{rem} = \{SI^j_i : afd(x^j, SI^j_i) \leq d_{sup},\ i = 1, \ldots, N^j_I\}$.

2. If a feature X^j is immutable, this means that the value x_j of a query instance $\boldsymbol{x}$ and the value x'^j of its counterfactual sample should be the same. Thus, all split intervals in SI^j that do not contain x^j should be filtered out, i.e, the remaining split intervals are $SI^j_{rem} = \{SI^j_i : x_j \in SI^j_i,\ i = 1, \ldots, N^j_I\}$. This is equivalent to removing a dimension in the feature space.

3. If a feature X^j can be only increased, all split intervals whose upper bounds are smaller than x^j can be filtered out, i.e., the remaining split intervals are $SI^j_{rem} = \{SI^j_i : ev^j_i \geq x_j,\ afd(x^j, SI^j_i) \leq d_{sup},\ i = 1, \ldots, N^j_I\}$.

4. If a feature X^j can be only decreased, all split intervals whose lower bounds are larger than x^j can be filtered out, i.e., the remaining split intervals are $SI^j_{rem} = \{SI^j_i : sv^j_i \leq x_j, afd(x^j, SI^j_i) \leq d_{sup},\ i = 1, \ldots, N^j_I\}$.

Here, $afd(\cdot)$ is the adjusted distance along a single feature, defined as

$$afd(x^j,]sv^j, ev^j]) = \frac{fd(x^j,]sv^j, ev^j])}{MAD^j}, \tag{6.13}$$

or

$$afd(x^j,]sv^j, ev^j]) = \sqrt{\frac{fd(x^j,]sv^j, ev^j])^2}{STD^j}}, \tag{6.14}$$

where

$$fd(x^j,]sv^j, ev^j]) = \begin{cases} 0 & \text{if } sv^j < x^j \leq ev^j, \\ sv^j - x^j + \epsilon & \text{if } x^j \leq sv^j, \\ x^j - ev^j & \text{if } x^j > ev^j, \end{cases} \tag{6.15}$$

and

$$\epsilon = \frac{1}{2}\ \min\{ev^j_i - sv^j_i, \forall j = 1, \ldots, M, i = 1, \ldots, N^j_I\}. \tag{6.16}$$

The parameter ϵ is also used in the representation of the cautious random forest. If feature X^j is an integer, then

$$fd(x^j,]sv^j, ev^j]) = \begin{cases} \lceil sv^j + \epsilon \rceil - x^j & \text{if } x^j \leq sv^j, \\ \lceil x^j - ev^j \rceil & \text{if } x^j > ev^j. \end{cases} \tag{6.17}$$

Algorithm 5 compiles the preprocessing steps presented above and returns the regions constructed by the remaining split intervals, where the closest counterfactuals are located.

Algorithm 5: Preprocessing

Input: Query instance $\boldsymbol{x}$; initial upper bound distance d_{sup}; leaves of cautious random forest $\boldsymbol{L}$; feature restrictions; distance measure $afd(\cdot)$.

Output: Remaining split intervals SI_{rem}.

$SI_{rem} = \{\}$
for $j = 1, \ldots, M$ **do**
  $SI^j_{rem} = \{\}$
  if X^j *is immutable* **then**
    for $i = 1, \ldots, N^j_I$ **do**
      if $sv^j_i < x^j \leq ev^j_i$ **then**
        $SI^j_{rem} = SI^j_{rem} \cup SI^j_i$
  else if X^j *is only increasing* **then**
    for $i = 1, \ldots, N^j_I$ **do**
      if $ev^j_i \geq x^j$ *AND* $afd(x^j, SI^j_i) \leq d_{sup}$ **then**
        $SI^j_{rem} = SI^j_{rem} \cup SI^j_i$
  else if X^j *is only decreasing* **then**
    for $i = 1, \ldots, N^j_I$ **do**
      if $sv^j_i \leq x^j$ *AND* $afd(x^j, SI^j_i) \leq d_{sup}$ **then**
        $SI^j_{rem} = SI^j_{rem} \cup SI^j_i$
  else
    for $i = 1, \ldots, N^j_I$ **do**
      if $afd(x^j, SI^j_i) \leq d_{sup}$ **then**
        $SI^j_{rem} = SI^j_{rem} \cup SI^j_i$
  $SI_{rem} = SI_{rem} \cup SI^j_{rem}$
return SI_{rem}

Example 6.3 (Preprocessing procedure)**.** *Fig. 6.5 illustrates the preprocessing procedure with an example applied to a cautious random forest trained on 2D binary classification data.*

The two classes are depicted by the blue and orange regions, while the gray regions correspond to indeterminate decisions. The problem at hand involves finding a counterfactual of the orange class for a query instance $\boldsymbol{x}$ (represented by the black point) with an indeterminate prediction.

The initial counterfactual can be obtained through the MO method (as denoted

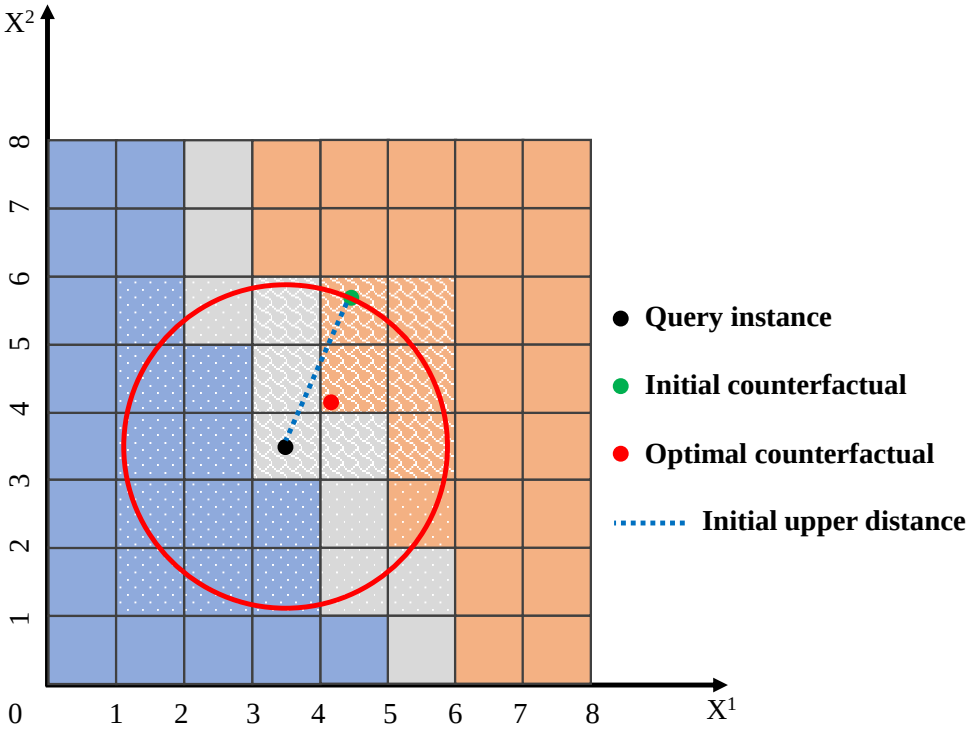


Figure 6.5: An example of a cautious random forest based on 2D data.

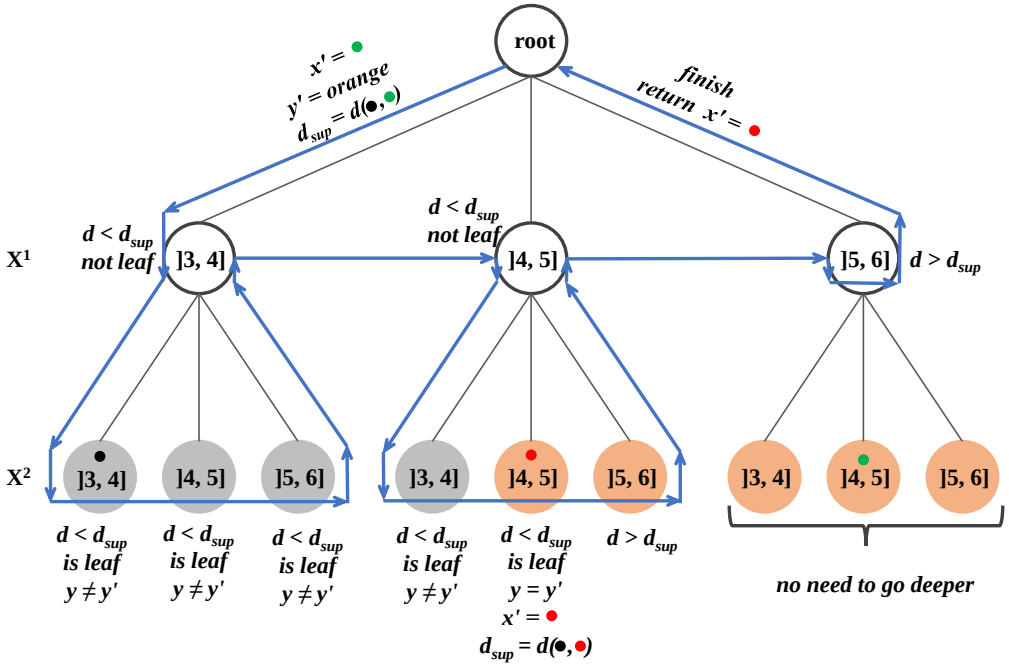


Figure 6.6: An example of search tree and the corresponding brand-and-bound search process for the closest counterfactual example based on Fig. 6.5.

by the green point), and the initial upper bound distance d_{sup} can be computed. Subsequently, it can be affirmed that the optimal counterfactual must necessarily lie within the circle centered on $\boldsymbol{x}$ with a radius of d_{sup} (or within the hyper-sphere for high-dimensional data).

The red point is the closest counterfactual sample classified into the orange class. If there are no constraints on the features, then the search can be limited to the textured area in the graph, consisting of 25 bins, rather than the entire feature space of 64 bins. Additionally, if it is assumed that the values of two features cannot be decreased, the search range can be further narrowed down to the area with a wavy texture, consisting of only 9 bins.

6.2.3 Branch-and-bound search for counterfactuals

We present here our branch-and-bound algorithm for counterfactual generation, given a set of split intervals and leaf information. The main idea of this algorithm is to start from the region containing $\boldsymbol{x}$ and expand the search to further regions. The distance of the explored region to $\boldsymbol{x}$ cannot exceed the currently known upper bound distance d_{sup}.

This process can be represented by a search tree, where each level corresponds to a feature (the depth being at most M), and where each node on a given level is a split interval for the corresponding feature. Thus, the path from the root to the leaf corresponds to a sequence of M split intervals, leading to a decision region where the prediction is pure. Assume that for each feature X^j, split intervals in SI^j_{rem} (after preprocessing) are sorted in ascending order according to their distance to x^j. Then, the leftmost node on each level in the search tree is the split interval where the corresponding feature value of $\boldsymbol{x}$ is located. In this way, the generation of counterfactuals can be achieved by a pre-order traversal of the search tree. Example 6.4 and Fig. 6.6 provide an example for this search process for the query instance displayed in Fig. 6.5.

Example 6.4 (Branch-and-bound search for the closest counterfactuals)**.** *Assume*

the search tree has been constructed using the information related to the query instance and to the additional actionability constraints (explained in Fig.6.5).

The search starts at the root, and explores the regions such that $3 < \mathrm{X}^1 \leq 4$*, corresponding to the leftmost node at the first level. The first child of this node corresponds to the region* $[\mathrm{X}^1, \mathrm{X}^2] \in]3, 4] \times]3, 4]$*, which is within the upper-bound distance but does not correspond to the desired prediction—as its two sibling nodes.*

The search therefore goes back to the previous level in the tree and proceeds with exploring the regions such that $4 < \mathrm{X}^1 \leq 5$*, corresponding to the children of the central node. The first child does not give the desired prediction. However, the second one (*$[\mathrm{X}^1, \mathrm{X}^2] \in]4, 5] \times]4, 5]$*) produces the desired one and is within the upper-bound distance. A counterfactual sample is generated here and the upper-bound distance is updated. We note that the third child exceeds the upper-bound distance.*

The search goes back again to the first level to explore the rightmost node, i.e., the regions such that $5 < \mathrm{X}^1 \leq 6$*. Since the distance to the root exceeds the current upper-bound distance, there is no need to explore its subtree and the search finishes with going back to the root.*

In order to apply the branch-and-bound search strategy, the cumulative distance from the root to any node in the search tree should be kept track of. Suppose that the search process arrives at level $J \leq M$ of the search tree, then the path from the root to current node can be represented by a region in the form of J-dimensional box $\boldsymbol{R}^J = \{]sv^1, ev^1], \ldots,]sv^J, ev^J]\}$, where each element $]sv^j, ev^j]$ is the corresponding node of level j. Based on $\boldsymbol{R}^J$, the cumulative distance is defined according to the given distance measurement type as follows:

$$cd(\boldsymbol{x}, \boldsymbol{R}^J, L_1^{mad}) = \sum_{j=1}^{J} \frac{fd(x^j,]sv^j, ev^j])}{MAD^j}, \tag{6.18}$$

or

$$cd(\boldsymbol{x}, \boldsymbol{R}^J, L_2^{std}) = \sqrt{\sum_{j=1}^{J} \frac{fd(x^j,]sv^j, ev^j])^2}{STD^j}}, \tag{6.19}$$

where $fd(\cdot)$ is defined by Eq. (6.15).

Based on the cumulative distance, the forward and backward movements of the branch-and-bound search can be easily managed. If the sub-tree of the current node is incompletely explored and the cumulative distance from the root node to the current node does not exceed the upper bound distance d_{sup}, then the unexplored sub-tree should be prioritized for forward movement. Conversely, if the sub-tree of the current node is entirely explored, the algorithm should step back to the previous level and select a new subtree to explore. In cases where the cumulative distance from the root node to the current node exceeds d_{sup}, the exploration of all remaining subtrees of its parent node should be terminated, given that those subtrees to its right are guaranteed to be further away, and the cumulative distance of any node in these remaining sub-trees is consequently guaranteed to exceed d_{sup}.

Upon arriving at a leaf node of the search tree with a cumulative distance smaller than the current upper bound distance, the algorithm checks the prediction of the corresponding decision region. If the prediction matches the given desired class, a counterfactual instance is generated within this region, and the upper bound distance d_{sup} is updated to the (smaller) current cumulative distance before resuming the search. To generate a counterfactual instance $\boldsymbol{x}'$ of the query instance $\boldsymbol{x}$ in this multivariate decision region represented by a M-dimensional box $\boldsymbol{R}^M =]sv^1, ev^1], \ldots,]sv^M, ev^M]\}$, we define

$$x'^j = \begin{cases} x^j & \text{if } sv^j < x^j \leq ev^j, \\ sv^j + \epsilon & \text{if } x^j \leq sv^j, \\ ev^j & \text{if } x^j > ev^j, \end{cases} \tag{6.20}$$

where $j = 1, \ldots, M$ and ϵ is defined by Eq (6.16), which prevents the feature value from spanning entire intervals to ensure that the counterfactual feature value must

be within the intervals. If feature X^j is an integer, then

$$x'^j = \begin{cases} x^j & \text{if } sv^j < x^j \leq ev^j, \\ \lceil sv^j + \epsilon \rceil & \text{if } x^j \leq sv^j, \\ \lfloor ev^j \rfloor & \text{if } x^j > ev^j. \end{cases} \tag{6.21}$$

In order to guarantee the plausibility of the counterfactual instances produced by the algorithm, outlier detection techniques such as the *Local Outlier Factor* [27, 9] and the *connectedness* of counterfactual instances to training data [117] are employed in the process prior to updating $\boldsymbol{x}'$ and d_{sup}.

Algorithm 6 encapsulates all of the branch-and-bound search based counterfactual generation procedures presented above. Algorithm 7 presents the complete process for generating counterfactuals. It features a reconstruction step, which is required to further ensure the generated counterfactuals are effective. The justification for this is twofold. First, since the order of features may be different to the original one (for example ordered according to the feature importance), the feature order needs to be reset to the default one in the dataset. Second, the counterfactual samples generated using Eq. (6.20) do not consider the data type of the features. Notably, as the implementation is based on the scikit-learn random forest model, it only supports numerical features. Thus, the reconstruction process should take into account discrete numerical features, such as integers, to ensure the plausibility of the generated counterfactuals. Additionally, some groups of features derived from the encoding should be checked, e.g., in a set of one-hot encoding features, there should be one and only one feature equal to one, while the others should be zero.

6.2.4 Comparison of counterfactual generation methods

In this subsection, we evaluate the effectiveness of the counterfactuals generated by our method and their efficiency in handling large-scale problems by comparing them with those produced by different approaches.

Algorithm 6: Branch-and-bound search for counterfactual

Input: Query instance x; desired class y'; intial counterfactual cf_{init}; initial upper bound distance d_{sup}; Sorted split intervals SI, leaves of cautious random forest $\boldsymbol{L}$; distance measure along a single feature $fd(\cdot)$; outlier detector $OD(\cdot)$.

Output: Counterfactual example x'.

$x' = cf_{init}$
$n\_intervals = [NI^1, \ldots, NI^M]$
$n\_checked\_interval = [0, \ldots, 0]$
$cumu\_dist = 0$
$dim\_dist = [\emptyset, \ldots, \emptyset]$
$dim\_leaves = [\emptyset, \ldots, \emptyset]$
$region = [\emptyset, \ldots, \emptyset]$
$j = 1$
while *True* **do**
  if $j = 0$ **then**
    return x'
  else if $n\_checked\_intervals^j = n\_intervals^j$ **then**
    $n\_checked\_intervals^j = 0$
    $j = j - 1$
  else
    $n\_checked\_intervals^j = n\_checked\_intervals^j + 1$
    $i = n\_checked\_intervals^j$
    $interval = SI_i^j$
    $region^j = interval$
    $dim\_dist^j = fd(x^j, interval)$
    $cumu\_dist = cumu\_dist(dim\_dist^1, \ldots, dim\_dist^j)$
    if $cumu\_dist > d_{sup}$ **then**
      $n\_checked\_intervals^j = 0$
      $j = j - 1$
    else
      $dim\_leaves^j = R(interval, j, L)$
      if $j{=}M$ **then**
        $leaves = dim\_leaves^1 \cap \cdots \cap dim\_leaves^M$
        if *predict(leaves)* $= y'$ **then**
          $cf\_candidate = generate\_cf\_in\_region(x, region)$ via Eq. (6.20)
          if $OD(cf\_candidate)$ *is plausible* **then**
            $x' = cf\_candidate$
            $d_{sup} = cumu\_dist$
      else
        j = j + 1

Algorithm 7: Counterfactual Generation

Input: Query instance $\boldsymbol{x}$; classes $\Omega = \{c_1, c_2\}$; leaves of cautious random forest $\boldsymbol{L}$; distance measure $d(\cdot)$; feature restrictions FR; feature ordering FO; outlier detector $OD(\cdot)$.

Output: Counterfactuals $cfs\{\boldsymbol{x}^{c_1}, \boldsymbol{x}^{c_2}\}$.

$cfs = \{\}$
for $c_k \in \Omega$ **do**
  Initialize $\boldsymbol{x}^{c_k}$ via MO or OFCC method taking c_k
  $d_{sup} = d(\boldsymbol{x}, \boldsymbol{x}^{c_k})$
  SI = Preprocessing($\boldsymbol{x}$, d_{sup}, $\boldsymbol{L}$, RF, $d(\cdot)$)
  Sort SI^j according to $fd(\boldsymbol{x}, SI_i^j)$, $j = 1, \ldots, M$, $i = 1, \ldots, N_I^j$
  Reorder features in $\boldsymbol{x}$, SI, and $\boldsymbol{L}$ according to feature ordering FO
  $\boldsymbol{x}^{c_k}$ = Branch-and-Bound Search($\boldsymbol{x}$, c_k, d_{sup}, SI, $\boldsymbol{L}$, $d(\cdot)$, $OD(\cdot)$)
  x^{c_k} = Reconstruction($\boldsymbol{x}^{c_k}$, FI, FR) $cfs = cfs \cup x^{c_k}$
return cfs

Table 6.4: Datsets used in experiments.

Data name	Abbreviation	n-feature	n-sample
Adult	ADLT	11	45222
Biodeg	BIOD	41	1053
Compas	COMP	6	2652
German	GERM	24	1000
Heloc	HELO	23	10459
Liver	LIVR	6	345
Mammographic	MAMO	5	830
Pima	PIMA	8	768
Spam	SPAM	57	4594
Wine	WINE	11	1599

The experiments are performed on ten distinct binary classification datasets sourced from the UCI repository. Table 6.4 provides a summary of their characteristics, giving their numbers of features and samples. Each cautious random forest is made of 100 decision trees, all trained to maximum purity (i.e., purity of 1 in each leaf), and the imprecise Dirichlet model parameter is set to s=2. The random state of the scikit-learn implementation is set to 42 for all experiments. The reported results are the average of five-fold cross-validation. Evaluated results are reported from Table 6.5 to Table 6.9. In each table, the best result for each dataset is printed in bold.

In this section, different counterfactual generation methods are compared in

terms of the properties of counterfactual samples. The evaluation is based on the following criteria:

1. proximity (L_2^{std} and L_1^{mad}): the average distance between query instances and their generated counterfactual examples;
2. sparsity: the average number of modified features to obtain desired counterfactual examples;
3. plausibility: the proportion of generated counterfactual examples that are detected as non-outlier by the Local Outlier Factor method;
4. efficiency: the average elapsed time to generate counterfactual examples.

Since all methods implemented here generate valid counterfactuals by design, the validity is not reported. The compared methods in this experimentation are:

1. MO (Minimum Observable), which searches for the closest counterfactual in the training set [71];
2. DisCERN (Discovering Counterfactual Explanations using Relevance Features from Neighbourhoods), which replaces feature values ordered by feature importance with the corresponding feature values from MO until valid counterfactuals are returned [218];
3. OFCC (One-Feature-Changed Counterfactual), which tries to vary the value of only one feature while keeping the remaining ones unchanged to generate counterfactuals [231].

The results in Table 6.5 and Table 6.6 show that whatever the choice of the distance metric, the proposed approach generates the closest counterfactual samples. This is due to initializing the search with the results of DisCERN and OFCC before proceeding with exploring finer regions. These results indicate that our approach makes it possible to reduce the efforts of modifications to obtain desired counterfactual examples.

Table 6.5: Comparison of different counterfactual generation approaches in terms of L_2^{std}. The values in parentheses indicate the ratio between the distance to the counterfactuals generated by the different methods and the ones generated by our approach.

Data	MO	DisCERN	OFCC	Ours
ADLT	2.638 (2.76)	1.935 (2.03)	2.899 (3.04)	**0.955 (1.00)**
BIOD	2.800 (6.89)	0.891 (2.19)	0.419 (1.03)	**0.406 (1.00)**
COMP	0.926 (2.40)	0.731 (1.89)	0.552 (1.43)	**0.386 (1.00)**
GERM	4.142 (3.05)	2.700 (1.99)	1.625 (1.20)	**1.358 (1.00)**
HELO	9.733 (6.13)	5.415 (3.41)	1.789 (1.13)	**1.589 (1.00)**
LIVR	3.415 (6.54)	2.184 (4.18)	0.729 (1.40)	**0.522 (1.00)**
MAMO	0.935 (1.43)	0.807 (1.23)	0.784 (1.20)	**0.655 (1.00)**
PIMA	4.605 (6.61)	3.051 (4.38)	0.855 (1.23)	**0.697 (1.00)**
SPAM	5.301 (27.07)	1.986 (10.14)	0.202 (1.03)	**0.196 (1.00)**
WINE	1.479 (18.14)	0.734 (9.01)	0.133 (1.63)	**0.082 (1.00)**

Table 6.6: Comparison of different counterfactual generation approaches in terms of L_1^{mad}. The values in parentheses indicate the ratio between the distance to the counterfactuals generated by the different methods and the ones generated by our approach.

Data	MO	DisCERN	OFCC	Ours
ADLT	2.172 (10.29)	1.507 (7.14)	0.212 (1.00)	**0.211 (1.00)**
BIOD	17.024 (15.60)	4.154 (3.81)	1.095 (1.00)	**1.091 (1.00)**
COMP	0.413 (3.54)	0.351 (3.01)	0.150 (1.28)	**0.117 (1.00)**
GERM	7.981 (5.91)	3.828 (2.84)	1.393 (1.03)	**1.350 (1.00)**
HELO	16.564 (16.14)	5.007 (4.88)	1.073 (1.05)	**1.026 (1.00)**
LIVR	4.735 (12.72)	2.095 (5.63)	0.459 (1.23)	**0.372 (1.00)**
MAMO	0.475 (2.32)	0.346 (1.69)	0.250 (1.22)	**0.205 (1.00)**
PIMA	5.467 (12.44)	2.434 (5.54)	0.474 (1.08)	**0.439 (1.00)**
SPAM	7.796 (82.84)	1.820 (19.34)	0.146 (1.55)	**0.094 (1.00)**
WINE	7.638 (21.01)	2.569 (7.07)	0.393 (1.08)	**0.364 (1.00)**

Sparsity (see Table 6.7) is hard to satisfy by searching counterfactual samples in the training data, especially for datasets with a high number of continuous features such as the Wine dataset, while generating counterfactual instances via MO requires modifying almost all features. DisCERN significantly reduces the number of modified features by using feature importance, which nevertheless remains relatively high. OFCC is designed to modify only one feature, but for some samples, it is insufficient to alter the prediction. To ensure the validity in the implementation, if OFCC fails to generate valid counterfactuals, it returns the counterfactuals generated by DisCERN. This is why the average number of features changed by OCCF on some datasets may slightly exceed one. Our approach is reasonable in terms of the number of features changed, which overall does not exceed three features. This level of sparsity is consistent with the cognitive constraints of the human brain. Additionally, owing to the characteristics of the L1-norm, it can be observed that the counterfactual instances generated using L_1^{mad} are sparser in terms of features to be modified compared to those generated using L_2^{std}.

Table 6.7: Number of modified features in average (L_0-*norm* or sparsity) when optimizing L_2^{std} and L_1^{mad}, respectively.

Data	When optimizing L_2^{std}				When optimizing L_1^{mad}			
	MO	DisCERN	OFCC	Ours	MO	DisCERN	OFCC	Ours
ADLT	3.235	2.083	**1.000**	1.181	2.600	1.813	**1.000**	1.004
BIOD	18.575	5.681	**1.856**	2.081	18.425	5.981	**1.888**	1.950
COMP	1.916	1.514	**1.018**	1.176	1.680	1.464	**1.021**	1.078
GERM	8.216	4.041	**1.367**	1.680	6.644	3.431	**1.326**	1.423
HELO	16.305	5.519	**1.470**	1.584	14.435	5.034	**1.382**	1.411
LIVR	5.346	2.331	**1.000**	1.723	4.992	2.246	**1.000**	1.323
MAMO	1.730	1.404	**1.070**	1.189	1.578	1.337	**1.056**	1.152
PIMA	6.551	2.826	**1.000**	1.464	6.444	2.719	**1.000**	1.242
SPAM	15.673	2.615	**1.000**	3.978	15.038	2.942	**1.000**	1.038
WINE	10.155	3.158	**1.048**	2.951	10.030	3.075	**1.040**	2.381

In Table 6.8, the plausibility of counterfactuals generated via our approach is often less than MO and DisCERN because these latter return existing samples or mixtures of existing samples. However, checking the plausibility before updating the upper bound distance d_{sup} can guarantee that our approach generates counterfactual samples that are equally or more plausible than the initialization method.

Table 6.8: Plausibility of generated counterfactuals.

Data	MO	DisCERN	OFCC	Ours
ADLT	**0.976**	0.954	0.906	0.908
BIOD	**0.962**	0.947	0.938	0.938
COMP	**0.899**	0.884	0.792	0.807
GERM	**0.998**	0.996	0.978	0.986
HELO	0.998	0.998	0.998	**1.000**
LIVR	**0.988**	0.981	0.962	0.962
MAMO	**0.978**	0.974	0.974	0.974
PIMA	**0.988**	0.975	0.978	0.978
SPAM	**0.981**	0.875	0.846	0.846
WINE	**0.982**	0.930	0.933	0.938

Table 6.9 reports the average time for generating a counterfactual sample. MO and DisCERN exhibit the highest efficiency and are contingent upon the training set size. OFCC also exhibits strong performance, contingent upon both the number of features present in the data and the corresponding number of split values for each feature. Although our approach does not appear to be particularly efficient, we believe that generating the closest counterfactual instances within ten seconds using a random forest composed of 100 decision trees trained to maximum depth remains very acceptable. On most datasets, the generation of counterfactuals takes less than five seconds, and some only require approximately one second. This level of efficiency is highly favorable when compared to optimization-based counterfactual generation methods for large random forests.

Table 6.9: Average time to generate one counterfactual sample (seconds).

Data	MO	DisCERN	OFCC	Ours
ADLT	**0.228**	0.266	0.388	0.570
BIOD	**0.010**	0.142	3.555	5.232
COMP	**0.012**	0.034	0.481	0.570
GERM	**0.006**	0.057	0.503	2.116
HELO	**0.048**	0.098	1.880	5.518
LIVR	**0.006**	0.036	0.439	1.180
MAMO	**0.006**	0.034	0.128	0.191
PIMA	**0.006**	0.036	1.720	5.040
SPAM	**0.038**	0.126	3.957	7.574
WINE	**0.006**	0.041	2.120	8.324

6.3 Increasing efficiency using feature importance

The efficiency of the approaches proposed above for generating counterfactuals is heavily related to the order of features, i.e., the levels at which they are located in the search tree. If a decisive feature (a feature that needs to be modified in order to get the desired prediction) is at the top of the search tree, then searching all sub-trees of the node containing the value of the query instance on the decisive feature would be pointless. If the number of split intervals for each feature is N, the complexity would be $O(N^M)$ in the worst case. Conversely, if $J < M$ decisive features are at the bottom of the search tree, exploring all the split intervals of the decisive features would only require the exploration of the minimum sub-tree, which is equivalent to decreasing the depth of the search tree and thus reducing the search complexity to $O(N^J)$.

Therefore, to improve the efficiency of the procedure, features that are more likely to be modified to obtain counterfactuals should be located close to the bottom of the search tree. Thus, effective counterfactuals can be encountered earlier in the search process, allowing the upper bound distance to be updated to a smaller value quickly. This, in turn, quickly narrows the search range and eliminates unnecessary regions.

In previous works, feature importance is widely used for assessing which features may need to be mutated to generate counterfactuals. For example, Keane et al. proposed in [110] to mutate only features guided by the counterfactuals of instances close to the query instance, as they believe that similar instances already queried for are significant to obtain a counterfactual. In [170, 218], the authors proposed to mutate only features that contribute against the desired predictions, i.e., features with negative SHAP values associated with the query instance and the desired prediction.

We propose establish the feature order by evaluating feature importance, i.e., features with a higher importance are more likely to be modified during counterfactual generation and should be positioned close to the bottom of the search tree.

Given a list of feature importance values $\mathbf{\Phi} = \{\phi^1, \ldots, \phi^M\}$, the feature ordering $\preceq_{\mathbf{\Phi}}$ is defined as

$$\mathrm{X}^j \preceq_{\mathbf{\Phi}} \mathrm{X}^{j'} \text{ if } \phi^j \leq \phi^{j'},\ \forall j \in \{1, \ldots, M\},\ j \neq j', \tag{6.22}$$

where the estimated feature importance can be obtained via the methods described in the following sections. In addition to determining the order of features in the search tree, the feature importance assessed here is an explanation in itself, since it amounts to point out the features that are expected to be important.

6.3.1 Local feature importance assessment

Indeterminacy measures in predictions

Feature importance must be associated with an evaluation metric, such as the accuracy in precise classification problems. In our case, we focus on the indeterminacy of predictions. Thus, we propose two measures of indeterminacy, one written Imp_1 which indicates whether the prediction is determinate or not, and the other written Imp_2 which is a measure of uncertainty based on the prediction intervals. If the output $\boldsymbol{h}(\boldsymbol{x})$ for an instance $\boldsymbol{x}$ consists of a set of classes, we propose to define Imp_1 as follows:

$$\mathrm{Imp}_1(\boldsymbol{h}, \boldsymbol{x}) = \begin{cases} 0 & \text{if } |\boldsymbol{h}(\boldsymbol{x})| = 1, \\ 1 & \text{otherwise.} \end{cases} \tag{6.23}$$

If we consider interval-valued outputs ($I_1 = [\underline{p}_1, \overline{p}_1] = [bel_1(\boldsymbol{x}), pl_1(\boldsymbol{x})]$ and $I_2 = [\underline{p}_2, \overline{p}_2] = [bel_2(\boldsymbol{x}), pl_2(\boldsymbol{x})]$), following the definition in [98], we may alternatively define the indeterminacy measure Imp_2 as

$$\mathrm{Imp}_2(\boldsymbol{h}, \boldsymbol{x}) = \min(\overline{p}_1, \overline{p}_2). \tag{6.24}$$

By duality, the uncertainty quantification can calculated either using I_1 or I_2 due to the relationship $\overline{p}_1 = 1 - \underline{p}_2$ and $\overline{p}_2 = 1 - \underline{p}_1$, i.e., $\mathrm{Imp}_2(\boldsymbol{h}, \boldsymbol{x}) = \min(\overline{p}_1, 1 - \underline{p}_1)$ or $\mathrm{Imp}_2(\boldsymbol{h}, \boldsymbol{x}) = \min(1 - \underline{p}_2, \overline{p}_2)$.

Remark 6.1. *Since* $\overline{p}_1 = \overline{p}_1 + \underline{p}_1 - \underline{p}_1$ *and* $1 - \underline{p}_1 = 1 - \underline{p}_1 + \overline{p}_1 - \overline{p}_1$*, we have*

$$\begin{aligned}\mathrm{Imp}_2(\boldsymbol{h}, \boldsymbol{x}) &= \min(\overline{p}_1, 1 - \underline{p}_1) \\ &= \min(\underline{p}_1 + \overline{p}_1 - \underline{p}_1, 1 - \overline{p}_1 + \overline{p}_1 - \underline{p}_1) \\ &= \min(\underline{p}_1, 1 - \overline{p}_1) + \overline{p}_1 - \underline{p}_1 \\ &= \min(\underline{p}_1, \underline{p}_2) + \overline{p}_1 - \underline{p}_1.\end{aligned}$$

The former part is called aleatoric uncertainty and the latter, representing the length of the probability intervals I_1 *and* I_2*, is known as the epistemic uncertainty in the prediction.*

According to the decision strategy in Eq. (3.7), all determinate predictions ($\underline{p}_1 \leq 0.5$ *or* $\underline{p}_2 \leq 0.5$*) yield a valeur of* Imp_2 *less than or equal to 0.5, and* Imp_2 *approaches zero when* $\underline{p}_1$ *tends to 1 or* $\overline{p}_1$ *tends to 0. In contrast, the value of* Imp_2 *for indeterminate predictions (*$\overline{p}_1 > 0.5$ *and* $\overline{p}_2 > 0.5$*) is greater than 0.5 and tends to 1 when* $\underline{p}_1$ *tends to 0 and* $\overline{p}_1$ *tends to 1.*

Local permutation feature importance

Assume that $\boldsymbol{x}^{\sim j}$ is an observation (of the random vector $\mathrm{X}^{\sim j}$) obtained by conducting an aleatory modification on the feature X^j of $\boldsymbol{x}$ while all other features are kept unchanged, i.e., the *j-th* element x^j of vector $\boldsymbol{x}$ is replaced by a new value z^j, a conditional distribution can be generated as follows:

$$f_{\mathbf{X}^{\sim j}} = f_{\mathrm{X}^j|\boldsymbol{x}^{\text{-}j}}(z^j) = \mathbb{P}(\mathrm{X}^j = z^j | \mathrm{X}^{j'} = x^{j'}, \forall j' \neq j), \tag{6.25}$$

where $\boldsymbol{x}^{\text{-}j}$ represents the sub-vector observation consisting of all elements of $\boldsymbol{x}$ except the *j-th* one. Respectively, the random vector $\mathbf{X}^{\text{-}j}$ is obtained by marginalizing the random variable X^j out of $\mathbf{X}$.

Example 6.5. *Assume that* $\boldsymbol{x} = (3, 4, 2)^T$ *and we want to calculate the feature*

importance of X^2*, the random variable* $\mathrm{X}^{\sim 2}$ *should be considered:*

$$f_{\mathbf{X}^{\sim 2}}(z) = \mathbb{P}(\mathrm{X}^2 = z | \mathrm{X}^1 = 3, \mathrm{X}^3 = 2).$$

Remark 6.2. *In general, the conditional distribution* $f_{X^{\sim j}}$ *is difficult to calculate analytically; a notable exception is the Gaussian case, where any conditioning of a Gaussian random vector by a sub-vector is Gaussian, with the conditional distribution allowing an analytic expression.*

Given a test instance $\boldsymbol{x}$, we can define the local feature importance of X^j as the expected gain on determinacy if only feature value x^j associated with instance $\boldsymbol{x}$ is allowed to be modified:

$$\phi(j; \boldsymbol{h}, \boldsymbol{x}) = \mathbb{E}_{\mathbf{X}^{\sim j}} \left[\mathrm{Imp}(\boldsymbol{h}(\boldsymbol{x})) - \mathrm{Imp}(\boldsymbol{h}(\mathbf{X}^{\sim j})) \right] \tag{6.26a}$$

$$= \mathrm{Imp}(\boldsymbol{h}(\boldsymbol{x})) - \mathbb{E}_{\mathbf{X}^{\sim j}} \left[\mathrm{Imp}(\boldsymbol{h}(\mathbf{X}^{\sim j})) \right]. \tag{6.26b}$$

In practice, for each of these instance $\boldsymbol{x}$, we can estimate the local feature importance (6.26) by averaging the determinacy gain over the instances similar to $\boldsymbol{x}$ except for the value x^j , i.e., obtained by replacing the *j-th* value of $\boldsymbol{x}$ by values sampled according to the conditional distribution of the random variable $\mathbf{X}^{\sim j}$. Alternatively, from a sufficiently large validation set, we can also estimate the local feature importance by selecting samples similar to $\boldsymbol{x}$ except for value x^j.

Local Interpretable Model-agnostic Explanations (LIME)

For a given query instance $\boldsymbol{x}$ and a given model $\boldsymbol{h}$ to be explained, the idea of LIME is to locally approximate $\boldsymbol{h}$ by an interpretable surrogate model $\boldsymbol{\xi}$. For this purpose, LIME creates a new dataset by generating samples around $\boldsymbol{x}$ and collects the corresponding predictions provided by $\boldsymbol{h}$. A new interpretable model, e.g., linear regression, is trained using the new dataset in which each sample is then weighted according to its proximity to $\boldsymbol{x}$. The search for the best surrogate model is formulated

mathematically as follows:

$$\boldsymbol{\xi}(\boldsymbol{x}) = \arg\min_{\boldsymbol{h}' \in \mathcal{H}} \mathcal{L}(\boldsymbol{h}, \boldsymbol{h}', \boldsymbol{\pi}_{\boldsymbol{x}}) + O(\boldsymbol{h}'), \tag{6.27}$$

where $\mathcal{L}(\cdot)$ is a loss function measuring the fidelity of $\boldsymbol{h}'$ to $\boldsymbol{h}$, $O(\cdot)$ measures the model complexity, and $\boldsymbol{\pi}_{\boldsymbol{x}}$ measures the proximity between $\boldsymbol{x}$ and the instances sampled around it. In our case, the outputs of model $\boldsymbol{h}$ and $\boldsymbol{h}'$ are replaced by the indeterminacy measures of predictions, which are defined in Eq. (6.23) and Eq. (6.24).

SHapley Additive exPlanations (SHAP)

Unlike LIME, which learns an approximate model, SHAP makes use of the Shapley value [182], a concept from cooperative game theory that measures the contribution of each player to the total payoff of a game. The basic idea behind SHAP is to assign a score $\phi(\boldsymbol{x}, j)$ to each feature X^j that measures its contribution to the model prediction for a given instance $\boldsymbol{x}$. This score is based on the difference between the model prediction for the instance with and without the feature X^j. More precisely, the SHAP score for feature X^j and instance $\boldsymbol{x}$ is defined as:

$$\phi(j; \boldsymbol{x}, \boldsymbol{h}) = \sum_{S \subseteq \{1,2,\ldots,M\} \setminus \{j\}} \frac{|S|!(M - |S| - 1)!}{M!} (f_{\boldsymbol{h}}(\boldsymbol{x}^{S \cup \{j\}}) - f_{\boldsymbol{h}}(\boldsymbol{x}^S)), \tag{6.28}$$

where M is the total number of features, S is a subset of the features excluding feature X^j, $f_{\boldsymbol{h}}(\boldsymbol{x}^S)$ is a function associated with model $\boldsymbol{h}$ for instance $\boldsymbol{x}$ with only the features in S, and $f_{\boldsymbol{h}}(x^{S \cup \{j\}})$ is the one associated with model $\boldsymbol{h}$ for instance $\boldsymbol{x}$ with the feature X^j added to S. In our case, the payoff function $f_{\boldsymbol{h}}(\cdot)$ is replaced by the indeterminacy measures of predictions, i.e., $\mathrm{Imp}_1(\boldsymbol{h}(\cdot))$ and $\mathrm{Imp}_2(\boldsymbol{h}(\cdot))$ that are defined in Eq. (6.23) and Eq. (6.24), respectively.

Intuitively, the SHAP score measures the average marginal contribution of feature X^j across all possible subsets of features that do not contain X^j. It can be interpreted as the importance of feature X^j for explaining the prediction of the model output for $\boldsymbol{x}$. In order to estimate the SHAP score, Lundberg et al. proposed

Kernel SHAP [128] and Tree SHAP [127], which are respectively model-agnostic and specific to tree-based models. Actually, Kernel SHAP connects LIME and Shapley values, i.e., it is a kind of LIME if the proximity measurement $\boldsymbol{\pi}_{\boldsymbol{x}}$ in LIME is no longer based on distance but replaced by the SHAP kernel.

6.3.2 Global feature importance measurements

In addition to the feature importance at the individual level, a global feature importance may be more helpful for the purpose of understanding the model. The following three different methods are the most applied to global feature importance evaluation.

Mean Decrease in Impurity (MDI)

The strategy of growing a decision tree is to determine the best splits in internal nodes in order to decrease the impurity in their descendant nodes. For a given feature, its MDI is the average decrease in the impurity of the nodes where it is selected as the split feature, weighted by the proportion of samples in the nodes [24, 122]. Intuitively, features achieving a great decrease of impurity across a tree are more discriminative and are important for the separation of the feature space. The MDI of a random forest can be defined as the average MDI across all trees. MDI is a model-specific method, only suitable for tree-based models.

Features with a large MDI generally appear in the shallow layers of the trees, corresponding to regions that are easy to separate, while features of small MDI often appear at the bottom, corresponding to areas where different classes are highly mixed in the feature space, where the indeterminacy of our cautious random forests comes from. Therefore, features with a high MDI are likely to be helpful in resolving indeterminacy.

Permutation Feature Importance (PFI)

It measures the decrease in the prediction accuracy of the model after the feature values have been swapped. It is also referred to as the mean decrease in accuracy [141]. Machine learning models learn the association between input features and outputs. Therefore, if there is a significant decrease in the prediction accuracy when a feature in the dataset is perturbed, it indicates that the model prediction highly relies on this feature [24]. PFI is a model-agnostic method, and the performance measurement can also be different from the accuracy.

In our case, we apply PFI to explain the relationship between the features and the indeterminacy of the cautious random forest model. Since our purpose is to resolve indeterminacy of set-valued predictions, we consider only observations $\boldsymbol{x}$ such that $|\boldsymbol{h}(x)| > 1$, i.e., consider the conditional distribution defined as follows:

$$f_{\mathbf{X}\,||\boldsymbol{h}(\mathbf{X})|>1}(\boldsymbol{x}) = \mathbb{P}(\mathbf{X} = \boldsymbol{x} \,|\, |\boldsymbol{h}(\boldsymbol{x})| > 1). \tag{6.29}$$

The global feature importance of X^j is defined as the expected gain of determinacy associated with a classifier output based on the conditional distribution $f_{\mathbf{X}\,||\boldsymbol{h}(\mathbf{X})|>1}$:

$$\Phi(j; \boldsymbol{h}) = \mathbb{E}_{\mathbf{X}\,||\boldsymbol{h}(\mathbf{X})|>1}\left[\phi(j; \boldsymbol{h}, \boldsymbol{x})\right], \tag{6.30}$$

where $\phi(j; \boldsymbol{h}, \boldsymbol{x})$ is the measure of local feature importance of X^j associated with a specific observation $\boldsymbol{x}$ of conditional distribution $f_{\mathbf{X}\,||\boldsymbol{h}(\mathbf{X})|>1}$.

The theoretical calculation of the global feature importance is generally difficult. However, Eq. (6.30) can be practically estimated by replacing the expectation with an empirical average calculated on the instances of a set (e.g., test or validation set) for which the model outputs are indeterminate.

SHAP Feature Importance (SHAP-FI)

Since Shapley values are considered to be consistent, the local feature importance measures can be used to construct a global one. This is not the case for LIME that estimates feature importance using perturbed data. The SHAP feature importance measure (SHAP-FI) is calculated by averaging the absolute Shapley values per feature across all instances with indeterminate predictions:

$$\Phi(j; \boldsymbol{h}) = \frac{1}{N_{imp}} \sum_{\boldsymbol{x} \,|\, |\boldsymbol{h}(\boldsymbol{x})|>1} |\phi(j; \boldsymbol{h}, \boldsymbol{x})|, \tag{6.31}$$

where N_{imp} is the number of instances with imprecise predictions in the dataset and $\phi(j; \boldsymbol{h}, \boldsymbol{x})$ is defined by Eq. (6.28). Besides, SHAP-FI can provide a summary plot for all features to illustrate the relation between SHAP values and feature effects, which enables us to better understand the model dependency on features.

6.3.3 Evaluation of counterfactual generation acceleration

Since we aim at using feature importance as a guide for resolving the indeterminacy in predictions, we compare different feature importance measures via their proneness to accelerate the generation of counterfactual examples for instances with indeterminate predictions.

This section focuses on evaluating the effectiveness of three global feature importance assessment methods (MDI, PFI, and SHAP-FI) and two local methods (SHAP and LIME) in accelerating the branch-and-bound search for valid counterfactual instances without a preprocessing step. For LIME and SHAP, we use Imp_2 defined in Eq. (6.24) as a measure of indeterminacy because it is stated that LIME and SHAP are more capable to deal with continuous outputs.

Table 6.10 displays the time and efficiency improvements achieved by incorporating a feature ordering based on these feature importance methods in comparison to the default feature order. Remarkably, global feature importance plays a more significant role in improving the efficiency of branch-and-bound search, without any

Table 6.10: Impact of the use of feature importance for the acceleration of the branch-and-bound search for counterfactuals, reported with the average elapsed time (seconds) and the percentage of improvement in parentheses (%). The best results are printed in bold.

Data	Original	MDI (Global)	PFI (Global)	SHAP-FI (Global)	SHAP (Local)	LIME (Local)
ADLT	0.3019	**0.2022**	0.2162	0.2555	0.2743	0.3076
		(33.02)	(28.39)	(15.37)	(9.14)	(-1.89)
BIOD	2.3946	**0.0100**	0.0353	0.0626	1.6894	1.2866
		(99.58)	(98.53)	(97.39)	(29.45)	(46.27)
COMP	0.0161	**0.0122**	0.0123	0.0123	0.0133	0.0588
		(24.22)	(23.60)	(23.60)	(17.39)	(-265.22)
GERM	2.2392	**0.0279**	0.0280	0.0280	0.2275	0.3108
		(98.75)	(98.75)	(98.75)	(89.84)	(86.12)
HELO	5.2826	**2.6608**	2.8008	2.8615	3.7376	4.5674
		(49.63)	(46.98)	(45.83)	(29.25)	(13.54)
LIVR	0.9362	0.8745	0.8639	**0.8119**	0.9183	0.9839
		(6.59)	(7.72)	**(13.28)**	(1.91)	(-5.10)
MAMO	0.0046	**0.0006**	0.0008	0.0009	0.0010	0.0035
		(86.96)	(82.61)	(80.43)	(78.26)	(23.91)
PIMA	3.9084	**3.6413**	3.6650	3.6688	4.4324	4.1295
		(6.83)	(6.23)	(6.13)	(-13.41)	(-5.66)
SPAM	4.3789	3.9231	4.2194	3.7764	3.4110	3.7912
		(10.41)	(3.64)	(13.76)	**(22.10)**	(13.42)
WINE	8.2288	7.9983	**7.8751**	7.9673	8.3679	8.1653
		(2.80)	**(4.30)**	(3.18)	(-1.69)	(0.77)

counterpart in terms of sacrificing efficiency. Among the three global feature importance assessment methods considered, MDI exhibits the best performance. This is due to the way MDI calculates feature importance being based on the tree splits and impurity information, allowing it to identify features that are frequently used for splitting and thus that can quickly reduce impurity, which is directly linked to the elimination of indeterminacy in predictions.

Note also that local feature importance may lead to a reduction in terms of search efficiency, i.e., feature orderings determined by them for some samples to be highly inconsistent with the features actually needed to resolve indeterminacy. One possible reason may be that local feature importance measures are unstable. Another reason may be that LIME and SHAP build the linear surrogate model based on binary feature values (1 means including that feature and 0 without), which biases

the evaluation of feature importance. Using a global feature importance measure mitigates these shortcomings, resulting in a better overall performance across the entire dataset.

6.4 Conclusion

In this chapter, we have proposed a framework where counterfactual examples are used to explain indeterminate predictions made by a cautious random forest. The generated counterfactual explanations aim to address the questions of why a given input instance is classified indeterminately and how to modify some feature values to achieve a determinate prediction. We have proposed a branch-and-bound approach to search for the closest counterfactual examples and integrated plausibility and actionability considerations into the process. For accelerating the generation of proximal counterfactual examples, we have proposed to use local and global feature importance measures to determine the features that are more probable to be modified to get determinate predictions.

By comparing our counterfactual generation method with other methods, we have demonstrated the advantages of our method in terms of the proximity, sparsity, and plausibility of generated counterfactual examples, and a slight lack of efficiency. We have also shown that feature importance plays a significant role in enhancing the generation of counterfactual examples.

Conclusion and perspectives

Summary of the contributions

This thesis focuses on two important challenges in modern machine learning: making cautious under uncertainty and interpreting models and model outputs. We proposed a cautious random forest, a robust and explainable classifier, rooted in the imprecise probabilistic (the imprecise Dirichlet model) and belief-theoretic frameworks. This classification strategy can make indeterminate predictions when the uncertainty is too high, especially for test instances near the classification boundaries, which is essential to reduce the risk of making wrong decisions. In addition, by leveraging explainable AI concepts, we can provide explanations for indeterminate classifier outputs, including evaluating the importance of features for resolving indeterminacy and generating counterfactual samples to help users resolve indeterminacy.

Cautious random forests

When the available data or the information learned by the model is not sufficient to make reliable decisions, or when the outputs of the individual base learners in ensemble learning exhibit high levels of conflict, we advocate using a cautious approach where reliability is preserved, possibly at the expense of determinacy. A cautious classifier makes indeterminate (set-valued) predictions for such samples which pose challenges in terms of classification, thereby reducing the risk of making incorrect decisions.

In Chapter 3, we proposed a novel aggregation strategy to learn a cautious random forest within the context of binary imprecise classification. Our approach is formalized within the theory of belief functions, interpreting the imprecise tree outputs as pieces of evidence about the actual class of a test instance in the form of closed random intervals defined on $[0, 1]$. These posterior probability intervals are then aggregated into belief and plausibility degrees that can subsequently be used in a cautious decision-making process (such as the interval dominance principle) to indicate whether one of the two classes is strictly preferable to the other or not. Our strategy for aggregating these imprecise trees can be viewed as an extension of the weighted voting mechanism. Additionally, we developed a method to assign weights to individual trees by optimizing a novel cost function that takes both determinacy and accuracy into account. This weight assignment strategy allows us to strike a better balance between cautiousness and accuracy.

In Chapter 4, we extended the concept of cautious random forests to encompass multi-class classification problems. We introduced two cautious decision-making strategies to combine imprecise trees. They can be regarded as generalizations of averaging and voting in tree ensembles, which construct a mass function by either averaging probability intervals or applying the interval dominance principle. Subsequently, based on the obtained mass function, both of these strategies amount to maximizing the lower expected discounted utility to select the optimal subset of classes as indeterminate predictions, rather than the expected utility as done in the conventional case. It should be noted that this approach can be applied to any kind of classifier ensemble where classifier outputs are probability intervals; however, it is particularly well-suited to tree ensembles.

Explaining indeterminate predictions

In Chapter 6, we developed a framework to provide insights into indeterminate predictions made by a cautious random forest model. We make use of counterfactual examples to explain these indeterminate predictions; essentially, this amounts to answering two fundamental questions: why a particular input instance receives an

indeterminate classification and how the feature values can be adjusted to achieve a definitive prediction. To generate the nearest counterfactual examples, we implemented a branch-and-bound approach, incorporating considerations on the actionability of the features and the plausibility of the instances to be generated into the process. Additionally, to accelerate the generation of counterfactual examples, we leveraged both local and global feature importance measures to identify the features that are most likely to be modified to achieve determinate predictions.

Perspectives

1. Efficient cautious decision-making under belief functions

As mentioned in the first chapter, there are two strategies to perform imprecise classification: building partial preorders among precise assignments or building complete preorders among all possible partial assignments. The latter turns out to be very costly.

Our proposed cautious decision-making approach based on the theory of belief functions maximizes the lower expected discounted utility, which corresponds to a conservative or pessimistic decision-making strategy. This results in a partial preorder among all possible partial assignments from which the greatest partial assignment can be selected as the prediction. The high efficiency of the decision-making process relies on the special property of the definition and calculation of the belief degree of a given subset of the frame of discernment: only a part of focal elements need to be considered.

One of the drawbacks of our proposed approach is the difficulty of adjusting the level of cautiousness of the model. In future work, we may consider using the Hurwicz criterion to make decisions with different levels of cautiousness, so as to adapt to the needs of users or the specificities of datasets. The Hurwicz criterion takes both belief and plausibility degrees into account. However, the calculation of the plausibility degree of a given subset of the frame of discernment requires

considering all focal elements that have a nonempty intersection with it, which is expensive. Therefore, implementing this approach requires to identify an efficient way to compute the plausibility degrees.

2. Set-valued counterfactual examples

In precise classification problems, counterfactual examples are associated with determinate (precise) predictions. However, in cautious classification problems, the desired predictions can be imprecise [82].

This extension broadens the spectrum of the counterfactual generation framework. A user may desire to generate counterfactual examples with either a specific set-valued prediction or any subset (superset) of such a specified set. In this case, the existence of such counterfactual examples is questionable. Being able to evaluate the degree to which users' requirements are (possibly partially) fulfilled seems to be an important step in proposing a loss function for counterfactual example generation.

In future work, based on the proposals and discussions in [82], we may consider providing a theoretical formulation of the set-valued counterfactual generation problem for various kinds of user demands. Besides, a practical study of the real settings, in which set-valued counterfactual examples would be of interest, should be conducted, as the exact meaning of set-valued counterfactual explanations is not straightforward and may highly depend on the setting.

3. Efficiency of counterfactual example generation

The cautious random forest model captures instances near the classification boundary and assigns imprecise predictions to them due to class ambiguity. Therefore, the region where we search for counterfactual examples can be greatly narrowed by the preprocessing presented in Chapter 6, allowing the branch-and-bound approach to efficiently find the closest counterfactual examples.

However, for samples far from the classification boundary, the above preprocess-

ing cannot significantly narrow the range of the search for counterfactual examples. This happens in the following two kinds of scenarios. In precise classification, all samples may be screened so as to find counterfactual examples, among which a significant number may lie far from the classification boundary. In imprecise classification, some cautious classifiers are able to capture epistemic uncertainty, i.e., to assign indeterminate predictions to samples in low-density regions that are usually far from the classification boundary.

In future work, we may investigate new efficient preprocessing procedures to provide an efficient way of counterfactual generation for samples in low-density regions from the classification boundary.

4. Causality for counterfactual explanations in XAI

In causality, the cognitive ability of knowledge organization is described through three distinct levels (ladder of causation): association, intervention, and counterfactuals [156]. Conventional AI approaches stand on the first level (association), which means that models learn correlations rather than cause-effect relations. The second level (intervention) involves predicting the effects of actions on the environment. The highest level (counterfactuals) corresponds to modifying the course of events.

As we have seen in this manuscript, XAI gives a different meaning to the term "counterfactual" [30, 41, 44], which refers to the minimal modifications that must be made to a feature vector so as to change the prediction for an instance with a classifier.

Recently, several researchers argued that in order to achieve human-level explainability for a black-box machine learning model or its predictions, explanations should reflect causal relationships [96]. However, few counterfactual generation algorithms in XAI consider causality, especially model-agnostic ones, which leads to the explanations generated by these methods reflecting correlations rather than causality [35].

Therefore, the following research directions may be considered. First, the def-

initional ambiguity around the notion of counterfactual explanations and counterfactuals should be elucidated. A precise definition and an alignment of these two concepts would help bridging the gaps between the two fields. Based on this, it is essential to investigate how causality could be leveraged for counterfactual explanations as intended in XAI [41, 44]. Finally, it would be interesting to explore which kind of additional information is needed when using counterfactuals (i.e., at the highest level of causality) so as to explain machine learning outputs (and thus at a lower level of causality).

Appendix

A Gradient of the cost function

In this appendix, we provide the expression for the gradient of the proposed cost function. Considering the sigmoid function $\sigma(x)$ defined by Equation (3.14), it should be noted that

$$\sigma'(x) = \alpha\sigma(x)(1-\sigma(x)). \tag{32}$$

If we write

$$\underline{\mu}_i = \sigma(\boldsymbol{w}^\top \underline{\boldsymbol{\delta}}_i - 0.5), \tag{33a}$$

$$\overline{\mu}_i = \sigma(\boldsymbol{w}^\top \overline{\boldsymbol{\delta}}_i - 0.5), \tag{33b}$$

$$\mu_i = \sigma\left((\boldsymbol{w}^\top \underline{\boldsymbol{\delta}}_i - 0.5)(\boldsymbol{w}^\top \overline{\boldsymbol{\delta}}_i - 0.5)\right), \tag{33c}$$

the cost function (3.16) can be rewritten as:

$$\mathcal{L}(\boldsymbol{w}) = -\frac{1}{N}\sum_{i=1}^{N}\left\{z_i \ln(\underline{\mu}_i) + (1-z_i)\ln(1-\overline{\mu}_i) + \gamma\ln(1-\mu_i)\right\} + \frac{1}{2}\lambda||\boldsymbol{w}||_2^2. \tag{34}$$

Obviously, we have

$$\nabla_{\boldsymbol{w}} z_i \ln(\underline{\mu}_i) = \alpha z_i(1-\underline{\mu}_i)\underline{\boldsymbol{\delta}}_i, \tag{35a}$$

$$\nabla_{\boldsymbol{w}}(1-z_i)\ln(1-\overline{\mu}_i) = -\alpha(1-z_i)\overline{\mu}_i\overline{\boldsymbol{\delta}}_i, \tag{35b}$$

$$\nabla_{\boldsymbol{w}}\ln(1-\mu_i) = -\alpha\mu_i\left[(\underline{\boldsymbol{\delta}}_i\overline{\boldsymbol{\delta}}_i^\top + \overline{\boldsymbol{\delta}}_i\underline{\boldsymbol{\delta}}_i^\top)\boldsymbol{w} - 0.5(\underline{\boldsymbol{\delta}}_i + \overline{\boldsymbol{\delta}}_i)\right]. \tag{35c}$$

If we write $\boldsymbol{\delta}_i = (\underline{\boldsymbol{\delta}}_i\overline{\boldsymbol{\delta}}_i^\top + \overline{\boldsymbol{\delta}}_i\underline{\boldsymbol{\delta}}_i^\top)\boldsymbol{w} - 0.5(\underline{\boldsymbol{\delta}}_i + \overline{\boldsymbol{\delta}}_i)$, the gradient for the cost function writes as

$$\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w}) = -\frac{\alpha}{N}\sum_{i=1}^{N}\Big\{z_i(1-\underline{\mu}_i)\underline{\boldsymbol{\delta}}_i - (1-z_i)\overline{\mu}_i\overline{\boldsymbol{\delta}}_i - \gamma\mu_i\boldsymbol{\delta}_i\Big\} + \lambda\boldsymbol{w}. \tag{36}$$

B Hessian and convexity of the cost function

In this appendix, we provide the Hessian matrix of the cost function (3.16) and the proof that it is positive semi-definite, so as to prove the convexity of the cost function. First, the Hessian matrix can be calculated separately for each part of the cost function:

$$\boldsymbol{H}(z_i\ln(\underline{\mu}_i)) = -\alpha^2 z_i\underline{\mu}_i(1-\underline{\mu}_i)\underline{\boldsymbol{\delta}}_i\underline{\boldsymbol{\delta}}_i^\top, \tag{37a}$$

$$\boldsymbol{H}((1-z_i)\ln(1-\overline{\mu}_i)) = -\alpha^2(1-z_i)\overline{\mu}_i(1-\overline{\mu}_i)\overline{\boldsymbol{\delta}}_i\overline{\boldsymbol{\delta}}_i^\top, \tag{37b}$$

$$\boldsymbol{H}(\frac{1}{2}\lambda||\boldsymbol{w}||_2^2) = \lambda I, \tag{37c}$$

$$\boldsymbol{H}(\gamma\ln(1-\mu_i)) = -\alpha^2\gamma\mu_i(1-\mu_i)\boldsymbol{\delta}_i\boldsymbol{\delta}_i^\top - \alpha\gamma\mu_i(\underline{\boldsymbol{\delta}}_i\overline{\boldsymbol{\delta}}_i^\top + \overline{\boldsymbol{\delta}}_i\underline{\boldsymbol{\delta}}_i^\top). \tag{37d}$$

Consequently, the complete Hessian matrix writes as:

$$\begin{aligned}\boldsymbol{H}(\mathcal{L}(\boldsymbol{w})) =& \frac{\alpha^2}{N}\sum_{i=1}^{N}\Big\{z_i\underline{\mu}_i(1-\underline{\mu}_i)\underline{\boldsymbol{\delta}}_i\underline{\boldsymbol{\delta}}_i^\top + (1-z_i)\overline{\mu}_i(1-\overline{\mu}_i)\overline{\boldsymbol{\delta}}_i\overline{\boldsymbol{\delta}}_i^\top \\ &+\gamma\mu_i(1-\mu_i)\boldsymbol{\delta}_i\boldsymbol{\delta}_i^\top + \frac{1}{\alpha}\gamma\mu_i(\underline{\boldsymbol{\delta}}_i\overline{\boldsymbol{\delta}}_i^\top + \overline{\boldsymbol{\delta}}_i\underline{\boldsymbol{\delta}}_i^\top)\Big\} + \lambda I.\end{aligned} \tag{38}$$

All the matrices of the form $\xi\boldsymbol{a}\boldsymbol{a}^\top$, where ξ is a non-negative real number and $\boldsymbol{a}$ is a vector, are symmetric positive semi-definite. Moreover, λI is obviously symmetric positive definite. According to the theorem stating that the sum of two symmetric positive semi-definite matrices is also symmetric positive semi-definite, a sufficient and necessary condition for $\boldsymbol{H}(J(\boldsymbol{w}))$ to be a symmetric positive semi-definite matrix is that the last term in the sum be symmetric positive semi-definite as well.

Since $(\underline{\boldsymbol{\delta}}_i\overline{\boldsymbol{\delta}}_i^\top + \overline{\boldsymbol{\delta}}_i\underline{\boldsymbol{\delta}}_i^\top)^\top = \underline{\boldsymbol{\delta}}_i\overline{\boldsymbol{\delta}}_i^\top + \overline{\boldsymbol{\delta}}_i\underline{\boldsymbol{\delta}}_i^\top$, it is symmetric. Suppose we have two

non-zero vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$, and let $A = \boldsymbol{a}\boldsymbol{b}^\top$. All rows of A are linearly dependent. Therefore, $\det A = 0$, and $Rank(A) = 1$. Since the rank of a matrix is equal to the number of non-zero eigenvalues and its trace is equal to the sum of its eigenvalues, matrix A has only one non-zero eigenvalue and its value is equal to its trace, which is $Tr(A) = \boldsymbol{a}^\top \boldsymbol{b}$.

Bibliography

[1] Joaquín Abellán, Carlos J Mantas, and Javier G Castellano. "A random forest approach using imprecise probabilities". In: *Knowledge-Based Systems* 134 (2017), pp. 72–84.

[2] Joaquín Abellán and Andrés R Masegosa. "Imprecise classification with credal decision trees". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 20.5 (2012), pp. 763–787.

[3] Joaquín Abellán and Serafín Moral. "Building classification trees using the total uncertainty criterion". In: *International Journal of Intelligent Systems* 18.12 (2003), pp. 1215–1225.

[4] Joaquín Abellán, Serafín Moral, Manuel Gómez, and Andrés Masegosa. "Varying parameter in classification based on imprecise probabilities". In: *Advances in Soft Computing* 37 (2006), pp. 231–239.

[5] Amina Adadi and Mohammed Berrada. "Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)". In: *IEEE access* 6 (2018), pp. 52138–52160.

[6] Ajaya Adhikari, David MJ Tax, Riccardo Satta, and Matthias Faeth. "LEAFAGE: Example-based and Feature importance-based Explanations for Black-box ML models". In: *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE. 2019, pp. 1–7.

[7] Md Nasim Adnan, Md Zahidul Islam, et al. "Forex++: A new framework for knowledge discovery from decision forests". In: *Australasian Journal of Information Systems* 21 (2017), pp. 1–20.

[8] Yonatan Carlos Carranza Alarcon and Sébastien Destercke. "Imprecise gaussian discriminant classification". In: *Pattern Recognition* 112 (2021), p. 107739.

[9] Omar Alghushairy, Raed Alsini, Terence Soule, and Xiaogang Ma. "A review of local outlier factor algorithms for outlier detection in big data streams". In: *Big Data and Cognitive Computing* 5.1 (2020), p. 1.

[10] Ambika and Santosh Biradar. "Survey on Prediction of Loan Approval Using Machine Learning Techniques". In: *International Journal of Advanced Research in Science, Communication and Technology* (2021), pp. 449–454.

[11] Javier Antorán, Umang Bhatt, Tameem Adel, Adrian Weller, and José Miguel Hernández-Lobato. "Getting a clue: A method for explaining uncertainty estimates". In: *arXiv preprint arXiv:2006.06848* (2020).

[12] Daniel W Apley and Jingyu Zhu. "Visualizing the effects of predictor variables in black box supervised learning models". In: *arXiv preprint arXiv:1612.08468* (2016).

[13] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". In: *Information fusion* 58 (2020), pp. 82–115.

[14] Thomas Augustin, Frank PA Coolen, Gert De Cooman, and Matthias CM Troffaes. *Introduction to imprecise probabilities*. John Wiley & Sons, 2014.

[15] Kevin Bache and Moshe Lichman. *UCI machine learning repository*. 2013.

[16] Bart Baesens, Tony Van Gestel, Stijn Viaene, Maria Stepanova, Johan Suykens, and Jan Vanthienen. "Benchmarking state-of-the-art classification algorithms for credit scoring". In: *Journal of the operational research society* 54.6 (2003), pp. 627–635.

[17] Gagan Bansal. "Explanatory dialogs: Towards actionable, interactive explanations". In: *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. 2018, pp. 356–357.

[18] Osbert Bastani, Carolyn Kim, and Hamsa Bastani. "Interpreting blackbox models via model extraction". In: *arXiv preprint arXiv:1705.08504* (2017).

[19] Jean-Marc Bernard. "An introduction to the imprecise Dirichlet model for multinomial data". In: *International Journal of Approximate Reasoning* 39.2-3 (2005), pp. 123–150.

[20] Battista Biggio and Fabio Roli. "Wild patterns: Ten years after the rise of adversarial machine learning". In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. 2018, pp. 2154–2156.

[21] Pierre Blanchart. "An exact counterfactual-example-based approach to tree-ensemble models interpretability". In: *arXiv preprint arXiv:2105.14820* (2021).

[22] Henrik Boström, Ram B Gurung, Tony Lindgren, and Ulf Johansson. "Explaining random forest predictions with association rules". In: *Archives of Data Science, Series A* 5.1 (2018), A05.

[23] Leo Breiman. "Bagging predictors". In: *Machine Learning* 24.2 (1996), pp. 123–140.

[24] Leo Breiman. *Classification and regression trees*. Routledge, 2017.

[25] Leo Breiman. "Random forests". In: *Machine Learning* 45.1 (2001), pp. 5–32.

[26] Leo Breiman and Nong Shang. "Born again trees". In: *University of California, Berkeley, CA, Technical Report* 1.2 (1996), p. 4.

[27] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. "LOF: identifying density-based local outliers". In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 2000, pp. 93–104.

[28] Jonathan Brophy and Daniel Lowd. "TREX: Tree-Ensemble Representer-Point Explanations". In: *arXiv preprint arXiv:2009.05530* (2020).

[29] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. "Model compression". In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2006, pp. 535–541.

[30] Gianluca Carloni, Andrea Berti, and Sara Colantonio. "The role of causality in explainable artificial intelligence". In: *arXiv preprint arXiv:2309.09901* (2023).

[31] Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. "Ensemble selection from libraries of models". In: *Proceedings of the 21st international conference on Machine learning*. 2004, p. 18.

[32] Alison Cawsey. "User modelling in interactive explanations". In: *User Modeling and User-Adapted Interaction* 3 (1993), pp. 221–247.

[33] Tianqi Chen and Carlos Guestrin. "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd ACM SigKDD international conference on knowledge discovery and data mining*. 2016, pp. 785–794.

[34] Yu-Liang Chou, Catarina Moreira, Peter Bruza, Chun Ouyang, and Joaquim Jorge. "Counterfactuals and causability in explainable artificial intelligence: Theory, algorithms, and applications". In: *Information Fusion* 81 (2022), pp. 59–83.

[35] Yu-Liang Chou, Catarina Moreira, Peter Bruza, Chun Ouyang, and Joaquim Jorge. "Counterfactuals and causability in explainable artificial intelligence: Theory, algorithms, and applications". In: *Information Fusion* 81 (2022), pp. 59–83.

[36] C Chow. "On optimum recognition error and reject tradeoff". In: *IEEE Transactions on information theory* 16.1 (1970), pp. 41–46.

[37] R Dennis Cook. "Detection of influential observation in linear regression". In: *Technometrics* 42.1 (2000), pp. 65–68.

[38] Thomas Cover and Peter Hart. "Nearest neighbor pattern classification". In: *IEEE transactions on information theory* 13.1 (1967), pp. 21–27.

[39] Fabio G Cozman. "Credal networks". In: *Artificial intelligence* 120.2 (2000), pp. 199–233.

[40] Mark Craven and Jude Shavlik. "Extracting Tree-Structured Representations of Trained Networks". In: *Advances in Neural Information Processing Systems*. Vol. 8. MIT Press, 1995, pp. 24–30.

[41] Riccardo Crupi, Beatriz San Miguel González, Alessandro Castelnovo, and Daniele Regoli. "Leveraging Causal Relations to Provide Counterfactual Explanations and Feasible Recommendations to End Users." In: *ICAART (2)*. 2022, pp. 24–32.

[42] Zhicheng Cui, Wenlin Chen, Yujie He, and Yixin Chen. "Optimal action extraction for random forests and boosted trees". In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 2015, pp. 179–188.

[43] Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. "Multi-objective counterfactual explanations". In: *Parallel Problem Solving from Nature–PPSN XVI: 16th International Conference, PPSN 2020, Part I*. Springer. 2020, pp. 448–469.

[44] Saloni Dash, Vineeth N Balasubramanian, and Amit Sharma. "Evaluating and mitigating bias in image classifiers: A causal perspective using counterfactuals". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. IEEE. 2022, pp. 915–924.

[45] Brittany Davis, Maria Glenski, William Sealy, and Dustin Arendt. "Measure utility, gain trust: practical advice for XAI researchers". In: *2020 IEEE Workshop on TRust and EXpertise in Visual Analytics (TREX)*. IEEE. 2020, pp. 1–8.

[46] Cassio Polpo De Campos and Fabio Gagliardi Cozman. "The inferential complexity of Bayesian and credal networks". In: *IJCAI*. Vol. 5. Citeseer. 2005, pp. 1313–1318.

[47] Luis M De Campos, Juan F Huete, and Serafin Moral. "Probability intervals: a tool for uncertain reasoning". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 2.2 (1994), pp. 167–196.

[48] Juan José Del Coz, Jorge Díez, and Antonio Bahamonde. "Learning Nondeterministic Classifiers." In: *Journal of Machine Learning Research* 10.10 (2009), pp. 2273–2293.

[49] A. P. Dempster. "Upper and Lower Probabilities Generated by a Random Closed Interval". In: *The Annals of Mathematical Statistics* 39.3 (1968), pp. 957–966.

[50] Arthur P Dempster. "Upper and Lower Probabilities Induced by a Multivalued Mapping". In: *The Annals of Mathematical Statistics* 38 (1967), pp. 325–339.

[51] Janez Demšar. "Statistical comparisons of classifiers over multiple data sets". In: *The Journal of Machine Learning Research* 7 (2006), pp. 1–30.

[52] Houtao Deng. "Interpreting tree ensembles with intrees". In: *International Journal of Data Science and Analytics* 7.4 (2019), pp. 277–287.

[53] Thierry Denœux. "Constructing belief functions from sample data using multinomial confidence regions". In: *International Journal of Approximate Reasoning* 42.3 (2006), pp. 228–252.

[54] Thierry Denœux. "Extending stochastic ordering to belief functions on the real line". In: *Information Sciences* 179.9 (2009), pp. 1362–1376.

[55] Thierry Denoeux. "A k-nearest neighbor classification rule based on Dempster-Shafer theory". In: *IEEE transactions on systems, man, and cybernetics* 25.5 (1995), pp. 804–813.

[56] Thierry Denoeux. "A neural network classifier based on Dempster-Shafer theory". In: *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 30.2 (2000), pp. 131–150.

[57] Thierry Denoeux. "Analysis of evidence-theoretic decision rules for pattern classification". In: *Pattern recognition* 30.7 (1997), pp. 1095–1107.

[58] Thierry Denoeux. "Decision-making with belief functions: a review". In: *International Journal of Approximate Reasoning* 109 (2019), pp. 87–110.

[59] Sebastien Destercke. "A k-nearest neighbours method based on imprecise probabilities". In: *Soft Computing* 16.5 (2012), pp. 833–844.

[60] Sébastien Destercke and Benjamin Quost. "Combining binary classifiers with imprecise probabilities". In: *Integrated Uncertainty in Knowledge Modelling and Decision Making: International Symposium, IUKM 2011, Hangzhou, China, October 28-30, 2011. Proceedings*. Springer. 2011, pp. 219–230.

[61] Jacek P Dmochowski, Paul Sajda, and Lucas C Parra. "Maximum Likelihood in Cost-Sensitive Learning: Model Specification, Approximations, and Upper Bounds". In: *Journal of Machine Learning Research* 11.12 (2010), pp. 3313–3332.

[62] Finale Doshi-Velez and Been Kim. "Towards a rigorous science of interpretable machine learning". In: *arXiv preprint arXiv:1702.08608* (2017).

[63] Mengnan Du, Ninghao Liu, Qingquan Song, and Xia Hu. "Towards explanation of dnn-based prediction with guided feature inversion". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 1358–1367.

[64] Didier Dubois and Henri Prade. "On the use of aggregation operations in information fusion processes". In: *Fuzzy sets and systems* 142.1 (2004), pp. 143–161.

[65] Didler Dubois and Henri Prade. "Representation and combination of uncertainty with belief functions and possibility measures". In: *Computational intelligence* 4.3 (1988), pp. 244–264.

[66] Sean R Eddy. "Hidden markov models". In: *Current opinion in structural biology* 6.3 (1996), pp. 361–365.

[67] Ward Edwards. "The theory of decision making." In: *Psychological bulletin* 51.4 (1954), pp. 380–417.

[68] Irene Epifanio. "Intervention in prediction measure: a new approach to assessing variable importance for random forests". In: *BMC bioinformatics* 18.1 (2017), pp. 1–16.

[69] Fabio Fabris, Aoife Doherty, Daniel Palmer, João Pedro De Magalhães, and Alex A Freitas. "A new approach for interpreting random forest models and its application to the biology of ageing". In: *Bioinformatics* 34.14 (2018), pp. 2449–2456.

[70] Marco Farina, Yuichiro Nakai, and David Shih. "Searching for new physics with deep autoencoders". In: *Physical Review D* 101.7 (2020), p. 075021.

[71] Rubén R Fernández, Isaac Martín De Diego, Víctor Aceña, Alberto Fernández-Isabel, and Javier M Moguerza. "Random forest explainability using counterfactual sets". In: *Information Fusion* 63 (2020), pp. 196–207.

[72] Paul Fink. "Ensemble methods for classification trees under imprecise probabilities". MA thesis. Ludwig Maximilian University of Munich, 2012.

[73] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. "All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously." In: *J. Mach. Learn. Res.* 20.177 (2019), pp. 1–81.

[74] Kenneth R Foster, Robert Koprowski, and Joseph D Skufca. "Machine learning, medical diagnosis, and biomedical engineering research-commentary". In: *Biomedical engineering online* 13.1 (2014), pp. 1–9.

[75] Yoav Freund, Robert Schapire, and Naoki Abe. "A short introduction to boosting". In: *Journal-Japanese Society For Artificial Intelligence* 14.5 (1999), pp. 771–780.

[76] Yoav Freund and Robert E. Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *Journal of Computer and System Sciences* 55.1 (1997), pp. 119–139.

[77] Jerome H Friedman. "Greedy function approximation: a gradient boosting machine". In: *Annals of statistics* (2001), pp. 1189–1232.

[78] Jerome H Friedman and Bogdan E Popescu. "Predictive learning via rule ensembles". In: *The annals of applied statistics* (2008), pp. 916–954.

[79] Milton Friedman. "A comparison of alternative tests of significance for the problem of m rankings". In: *The Annals of Mathematical Statistics* 11.1 (1940), pp. 86–92.

[80] Nicholas Frosst and Geoffrey Hinton. "Distilling a neural network into a soft decision tree". In: *arXiv preprint arXiv:1711.09784* (2017).

[81] Giorgio Fumera, Fabio Roli, and Giorgio Giacinto. "Reject option with multiple thresholds". In: *Pattern recognition* 33.12 (2000), pp. 2099–2101.

[82] Gabriele Gianini, Jianyi Lin, Corrado Mio, and Ernesto Damiani. "Set-Based Counterfactuals in Partial Classification". In: *The 19th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*. Springer. 2022, pp. 560–571.

[83] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. "Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation". In: *journal of Computational and Graphical Statistics* 24.1 (2015), pp. 44–65.

[84] Hudson F Golino, Cristiano Mauro Assis Gomes, et al. "Visualizing random forest's prediction results". In: *Psychology* 5.19 (2014), pp. 2084–2098.

[85] Yves Grandvalet. "Bagging equalizes influence". In: *Machine Learning* 55.3 (2004), pp. 251–270.

[86] Riccardo Guidotti. "Counterfactual explanations and how to find them: literature review and benchmarking". In: *Data Mining and Knowledge Discovery* (2022), pp. 1–55.

[87] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. "Local rule-based explanations of black box decision systems". In: *arXiv preprint arXiv:1805.10820* (2018).

[88] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. "A survey of methods for explaining black box models". In: *ACM computing surveys (CSUR)* 51.5 (2018), pp. 1–42.

[89] Shanti S Gupta. "On some multiple decision (selection and ranking) rules". In: *Technometrics* 7.2 (1965), pp. 225–245.

[90] Thien M Ha. "The optimum class-selective rejection rule". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.6 (1997), pp. 608–615.

[91] James A Hanley and Barbara J McNeil. "A method of comparing the areas under receiver operating characteristic curves derived from the same cases." In: *Radiology* 148.3 (1983), pp. 839–843.

[92] Satoshi Hara and Kohei Hayashi. "Making tree ensembles interpretable: A bayesian model selection approach". In: *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*. PMLR. 2018, pp. 77–85.

[93] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. "Support vector machines". In: *IEEE Intelligent Systems and their applications* 13.4 (1998), pp. 18–28.

[94] Kilian Hendrickx, Lorenzo Perini, Dries Van der Plas, Wannes Meert, and Jesse Davis. "Machine learning with a reject option: A survey". In: *arXiv preprint arXiv:2107.11277* (2021).

[95] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network". In: *arXiv preprint arXiv:1503.02531* (2015).

[96] Andreas Holzinger, Georg Langs, Helmut Denk, Kurt Zatloukal, and Heimo Müller. "Causability and explainability of artificial intelligence in medicine". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 9.4 (2019), e1312.

[97] Ling Huang, Su Ruan, Pierre Decazes, and Thierry Denœux. "Lymphoma segmentation from 3D PET-CT images using a deep evidential network". In: *International Journal of Approximate Reasoning* 149 (2022), pp. 39–60.

[98] Eyke Hüllermeier, Sébastien Destercke, and Mohammad Hossein Shaker. "Quantification of credal uncertainty in machine learning: A critical analysis and

empirical comparison". In: *Proceedings of the 38th Conference on Uncertainty in Artificial Intelligence*. PMLR. 2022, pp. 548–557.

[99] Eyke Hüllermeier and Willem Waegeman. "Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods". In: *Machine Learning* 110.3 (2021), pp. 457–506.

[100] Leonid Hurwicz. "The generalized Bayes minimax principle: a criterion for decision making under uncertainty". In: *Cowles Comm. Discuss. Paper Stat* 335 (1951), p. 1950.

[101] Abdelhak Imoussaten and Lucie Jacquin. "Cautious classification based on belief functions theory and imprecise relabelling". In: *International Journal of Approximate Reasoning* 142 (2022), pp. 130–146.

[102] Folasade Olubusola Isinkaye, YO Folajimi, and Bolande Adefowoke Ojokoh. "Recommendation systems: Principles, methods and evaluation". In: *Egyptian informatics journal* 16.3 (2015), pp. 261–273.

[103] Raban Iten, Tony Metger, Henrik Wilming, Lídia Del Rio, and Renato Renner. "Discovering physical concepts with neural networks". In: *Physical review letters* 124.1 (2020), p. 010508.

[104] Adulam Jeyasothy, Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. "A general framework for personalising post hoc explanations through user knowledge integration". In: *International Journal of Approximate Reasoning* 160 (2023), p. 108944.

[105] Adulam Jeyasothy, Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. "Integrating Prior Knowledge in Post-hoc Explanations". In: *The 19th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*. Springer. 2022, pp. 707–719.

[106] Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, and Hiroki Arimura. "DACE: Distribution-Aware Counterfactual Explanation by Mixed-Integer Linear Optimization." In: *IJCAI*. 2020, pp. 2855–2862.

[107] Abdul Karim, Avinash Mishra, MA Newton, and Abdul Sattar. "Machine Learning Interpretability: A Science rather than a tool". In: *arXiv preprint arXiv:1807.06722* (2018).

[108] Amir-Hossein Karimi, Gilles Barthe, Borja Balle, and Isabel Valera. "Model-agnostic counterfactual explanations for consequential decisions". In: *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 895–905.

[109] Mark T Keane, Eoin M Kenny, Eoin Delaney, and Barry Smyth. "If only we had better counterfactual explanations: Five key deficits to rectify in the evaluation of counterfactual xai techniques". In: *arXiv preprint arXiv:2103.01035* (2021).

[110] Mark T Keane and Barry Smyth. "Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable AI (XAI)". In: *Case-Based Reasoning Research and Development: 28th International Conference, ICCBR 2020, Proceedings 28*. Springer. 2020, pp. 163–178.

[111] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. "Examples are not enough, learn to criticize! criticism for interpretability". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Curran Associates, Inc., 2016.

[112] Hyunjoong Kim, Hyeuk Kim, Hojin Moon, and Hongshik Ahn. "A weight-adjusted voting algorithm for ensembles of classifiers". In: *Journal of the Korean Statistical Society* 40.4 (2011), pp. 437–449.

[113] David G Kleinbaum, K Dietz, M Gail, Mitchel Klein, and Mitchell Klein. *Logistic regression*. Springer, 2002.

[114] Pang Wei Koh and Percy Liang. "Understanding black-box predictions via influence functions". In: *International conference on machine learning*. PMLR. 2017, pp. 1885–1894.

[115] Thibault Laugel, Adulam Jeyasothy, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. "Achieving Diversity in Counterfactual Explanations: a Review and Discussion". In: *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*. 2023, pp. 1859–1869.

[116] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. "Comparison-based inverse classification for interpretability in machine learning". In: *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations: 17th International Conference, IPMU 2018*. Springer. 2018, pp. 100–111.

[117] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. "The dangers of post-hoc interpretability: Unjustified counterfactual explanations". In: *arXiv preprint arXiv:1907.09294* (2019).

[118] Eric Lefevre, Olivier Colot, and Patrick Vannoorenberghe. "Belief function combination and conflict management". In: *Information fusion* 3.2 (2002), pp. 149–162.

[119] Isaac Levi. *The enterprise of knowledge: An essay on knowledge, credal probability, and chance*. MIT press, 1980.

[120] Hong Bo Li, Wei Wang, Hong Wei Ding, and Jin Dong. "Trees weighting random forest method for classifying high-dimensional noisy data". In: *2010 IEEE 7th international conference on e-business engineering*. IEEE. 2010, pp. 160–163.

[121] Na Li, Arnaud Martin, and Rémi Estival. "Heterogeneous information fusion: Combination of multiple supervised and unsupervised classification methods based on belief functions". In: *Information Sciences* 544 (2021), pp. 238–265.

[122] Xiao Li, Yu Wang, Sumanta Basu, Karl Kumbier, and Bin Yu. "A debiased MDI feature importance measure for random forests". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Vol. 32. Curran Associates Inc., 2019.

[123] Andy Liaw, Matthew Wiener, et al. "Classification and regression by randomForest". In: *R news* 2.3 (2002), pp. 18–22.

[124] Zachary C Lipton. "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery." In: *Queue* 16.3 (2018), pp. 31–57.

[125] Gilles Louppe, Louis Wehenkel, Antonio Sutera, and Pierre Geurts. "Understanding variable importances in forests of randomized trees". In: *Proceedings of the 26th International Conference on Neural Information Processing Systems*. Vol. 26. Curran Associates Inc., 2013, pp. 431–439.

[126] Ana Lucic, Harrie Oosterhuis, Hinda Haned, and Maarten de Rijke. "FOCUS: Flexible optimizable counterfactual explanations for tree ensembles". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 5. 2022, pp. 5313–5322.

[127] Scott M Lundberg, Gabriel G Erion, and Su-In Lee. "Consistent individualized feature attribution for tree ensembles". In: *arXiv preprint arXiv:1802.03888* (2018).

[128] Scott M Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Vol. 30. Curran Associates Inc., 2017, pp. 4768–4777.

[129] Liyao Ma and Thierry Denoeux. "Partial classification in the belief function framework". In: *Knowledge-Based Systems* 214 (2021), p. 106742.

[130] Batta Mahesh. "Machine learning algorithms-a review". In: *International Journal of Science and Research (IJSR)* 9 (2020), pp. 381–386.

[131] Francesca Mangili and Alessio Benavoli. "New prior near-ignorance models on the simplex". In: *International Journal of Approximate Reasoning* 56 (2015), pp. 278–306.

[132] Carlos J Mantas and Joaquín Abellán. "Analysis and extension of decision trees based on imprecise probabilities: Application on noisy data". In: *Expert Systems with Applications* 41.5 (2014), pp. 2514–2525.

[133] Morteza Mashayekhi and Robin Gras. "Rule extraction from decision trees ensembles: new algorithms based on heuristic search and sparse group lasso methods". In: *International Journal of Information Technology & Decision Making* 16.6 (2017), pp. 1707–1727.

[134] Denis D Mauá, Fabio G Cozman, Diarmaid Conaty, and Cassio P Campos. "Credal sum-product networks". In: *Proceedings of the Tenth International Symposium on Imprecise Probability: Theories and Applications*. PMLR. 2017, pp. 205–216.

[135] Markus Maurer, J Christian Gerdes, Barbara Lenz, and Hermann Winner. *Autonomous driving: technical, legal and social aspects*. Springer Nature, 2016.

[136] Quinn McNemar. "Note on the sampling error of the difference between correlated proportions or percentages". In: *Psychometrika* 12.2 (1947), pp. 153–157.

[137] Nicolai Meinshausen. "Node harvest". In: *Annals of Applied Statistics* 4.4 (2010), pp. 2049–2072.

[138] Jacob Metcalf, Emanuel Moss, Elizabeth Anne Watkins, Ranjit Singh, and Madeleine Clare Elish. "Algorithmic impact assessments and accountability: The co-construction of impacts". In: *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. 2021, pp. 735–746.

[139] Charles E Metz. "Basic principles of ROC analysis". In: *Seminars in nuclear medicine*. Vol. 8. 4. Elsevier. 1978, pp. 283–298.

[140] Tim Miller. "Explanation in artificial intelligence: Insights from the social sciences". In: *Artificial intelligence* 267 (2019), pp. 1–38.

[141] Christoph Molnar. *Interpretable machine learning*. Lulu.com, 2020.

[142] Serafín Moral-García, Carlos J Mantas, Javier G Castellano, Maria D Benitez, and Joaquín Abellán. "Bagging of credal decision trees for imprecise classification". In: *Expert Systems with Applications* 141 (2020), p. 112944.

[143] Thomas Mortier, Marek Wydmuch, Krzysztof Dembczyński, Eyke Hüllermeier, and Willem Waegeman. "Efficient set-valued prediction in multi-class classification". In: *Data Mining and Knowledge Discovery* 35.4 (2021), pp. 1435–1469.

[144] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. "Explaining machine learning classifiers through diverse counterfactual explanations". In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 2020, pp. 607–617.

[145] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. "Definitions, methods, and applications in interpretable machine learning". In: *Proceedings of the National Academy of Sciences* 116.44 (2019), pp. 22071–22080.

[146] Catherine K Murphy. "Combining belief functions when evidence conflicts". In: *Decision support systems* 29.1 (2000), pp. 1–9.

[147] Peter Bjorn Nemenyi. *Distribution-free multiple comparisons*. Princeton University, 1963.

[148] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. "Plug & play generative networks: Conditional iterative generation of images in latent space". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4467–4477.

[149] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2016, pp. 3395–3403.

[150] Vu-Linh Nguyen, Haifei Zhang, and Sébastien Destercke. "Learning Sets of Probabilities Through Ensemble Methods". In: *European Conference on Symbolic and Quantitative Approaches with Uncertainty*. Springer. 2023, pp. 270–283.

[151] Raymond S Nickerson. "Confirmation bias: A ubiquitous phenomenon in many guises". In: *Review of general psychology* 2.2 (1998), pp. 175–220.

[152] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. "Feature visualization". In: *Distill* 2.11 (2017), e7.

[153] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. "The building blocks of interpretability". In: *Distill* 3.3 (2018), e10.

[154] Anna Palczewska, Jan Palczewski, Richard Marchese Robinson, and Daniel Neagu. "Interpreting random forest models using a feature contribution method". In: *The 14th International Conference on Information Reuse & Integration (IRI)*. IEEE. 2013, pp. 112–119.

[155] Axel Parmentier and Thibaut Vidal. "Optimal counterfactual explanations in tree ensembles". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8422–8431.

[156] Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic books, 2018.

[157] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[158] Gean T Pereira and André CPLF de Carvalho. "Bringing robustness against adversarial attacks". In: *Nature Machine Intelligence* 1.11 (2019), pp. 499–500.

[159] Dragutin Petkovic, Ali Alavi, DanDan Cai, Jizhou Yang, and Sabiha Barlaskar. "RFEX: Simple Random Forest Model and Sample Explainer for non-Machine Learning experts". In: *bioRxiv* (2019), p. 819078.

[160] Lu Thi Kim Phung, Vo Thi Ngoc Chau, and Nguyen Hua Phung. "ExtractingRuleRF in Educational Data Classification: From a Random Forest to Interpretable Refined Rules". In: *2015 International Conference on Advanced Computing and Applications, ACOMP 2015*. Institute of Electrical and Electronics Engineers Inc., 2015, pp. 20–27. ISBN: 9781467382342.

[161] Piotr Płoński and Krzysztof Zaremba. "Visualizing random forest with self-organising map". In: *International Conference on Artificial Intelligence and Soft Computing*. Springer. 2014, pp. 63–71.

[162] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. "FACE: Feasible and actionable counterfactual explanations". In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 2020, pp. 344–350.

[163] Romila Pradhan, Jiongli Zhu, Boris Glavic, and Babak Salimi. "Interpretable data-based explanations for fairness debugging". In: *arXiv preprint arXiv:2112.09745* (2021).

[164] Foster Provost and Tom Fawcett. "Robust classification for imprecise environments". In: *Machine learning* 42.3 (2001), pp. 203–231.

[165] J Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[166] J Ross Quinlan. "Induction of decision trees". In: *Machine Learning* 1 (1986), pp. 81–106.

[167] Benjamin Quost and Sébastien Destercke. "Classification by pairwise coupling of imprecise probabilities". In: *Pattern Recognition* 77 (2018), pp. 412–425.

[168] Arthur Ramer. "Uniqueness of information measure in the theory of evidence". In: *Fuzzy Sets and Systems* 24.2 (1987), pp. 183–196.

[169] Pramila Rani, Changchun Liu, Nilanjan Sarkar, and Eric Vanman. "An empirical study of machine learning techniques for affect recognition in human–robot interaction". In: *Pattern Analysis and Applications* 9 (2006), pp. 58–69.

[170] Shubham Rathi. "Generating counterfactual and contrastive explanations using SHAP". In: *arXiv preprint arXiv:1906.09293* (2019).

[171] Atul Rawal, James McCoy, Danda B Rawat, Brian M Sadler, and Robert St Amant. "Recent Advances in Trustworthy Explainable Artificial Intelligence: Status, Challenges, and Perspectives". In: *IEEE Transactions on Artificial Intelligence* 3.6 (2021), pp. 852–866.

[172] Douglas A Reynolds et al. "Gaussian mixture models." In: *Encyclopedia of biometrics* 741 (2009), pp. 659–663.

[173] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1135–1144.

[174] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Anchors: High-precision model-agnostic explanations". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1. 2018.

[175] Irina Rish et al. "An empirical study of the naive Bayes classifier". In: *IJCAI 2001 workshop on empirical methods in artificial intelligence*. Vol. 3. 22. 2001, pp. 41–46.

[176] Marko Robnik-Šikonja and Marko Bohanec. "Perturbation-based explanations of prediction models". In: *Human and Machine Learning: Visible, Explainable, Trustworthy and Transparent*. Springer, 2018, pp. 159–175.

[177] Omer Sagi and Lior Rokach. "Explainable decision forest: Transforming a decision forest into an interpretable tree". In: *Information Fusion* 61 (2020), pp. 124–138.

[178] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. "Grad-cam: Visual explanations from deep networks via gradient-based localization". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626.

[179] Arash Shaban-Nejad, Martin Michalowski, John S Brownstein, and David L Buckeridge. "Guest editorial explainable AI: towards fairness, accountability, transparency and trust in healthcare". In: *IEEE Journal of Biomedical and Health Informatics* 25.7 (2021), pp. 2374–2375.

[180] Glenn Shafer. *A mathematical theory of evidence*. Princeton university press, 1976.

[181] Glenn Shafer and Vladimir Vovk. "A Tutorial on Conformal Prediction." In: *Journal of Machine Learning Research* 9.3 (2008), pp. 371–421.

[182] Lloyd S Shapley et al. *A value for n-person games*. Princeton University Press Princeton, 1953.

[183] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps". In: *arXiv preprint arXiv:1312.6034* (2013).

[184] Florentin Smarandache and Jean Dezert. "Information fusion based on new proportional conflict redistribution rules". In: *The 7th international conference on information fusion*. Vol. 2. IEEE. 2005, pp. 907–914.

[185] Philippe Smets. "Constructing the Pignistic Probability Function in a Context of Uncertainty." In: *UAI '89: Proceedings of the Fifth Annual Conference on Uncertainty in Artificial Intelligence*. Vol. 89. 1989, pp. 29–40.

[186] Philippe Smets. "Decision making in the TBM: the necessity of the pignistic transformation". In: *International journal of approximate reasoning* 38.2 (2005), pp. 133–147.

[187] Philippe Smets. "The combination of evidence in the transferable belief model". In: *IEEE Transactions on pattern analysis and machine intelligence* 12.5 (1990), pp. 447–458.

[188] Francesco Sovrano, Fabio Vitali, and Monica Palmirani. "Making things explainable vs explaining: Requirements and challenges under the GDPR". In: *International Workshop on AI Approaches to the Complexity of Legal Systems*. Springer. 2021, pp. 169–182.

[189] Lars St, Svante Wold, et al. "Analysis of variance (ANOVA)". In: *Chemometrics and intelligent laboratory systems* 6.4 (1989), pp. 259–272.

[190] Alexander Stevens, Peter Deruyck, Ziboud Van Veldhoven, and Jan Vanthienen. "Explainability and fairness in machine learning: Improve fair end-to-end lending for kiva". In: *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. 2020, pp. 1241–1248.

[191] Klemens Szaniawski. "Some remarks concerning the criterion of rational decision making". In: *Studia Logica* 9.1 (1960), pp. 221–239.

[192] Sarah Tan, Matvey Soloviev, Giles Hooker, and Martin T Wells. "Tree space prototypes: Another look at making tree ensembles interpretable". In: *Proceedings of the 2020 ACM-IMS on foundations of data science conference*. 2020, pp. 23–34.

[193] Vagan Terziyan and Oleksandra Vitko. "Causality-Aware Convolutional Neural Networks for Advanced Image Classification and Generation". In: *Procedia Computer Science* 217 (2023), pp. 495–506.

[194] Alaa Tharwat. "Linear vs. quadratic discriminant analysis classifier: a tutorial". In: *International Journal of Applied Pattern Recognition* 3.2 (2016), pp. 145–180.

[195] Eva Thelisson, Kirtan Padh, and L Elisa Celis. "Regulatory mechanisms and algorithms towards trust in AI/ML". In: *Proceedings of the IJCAI 2017 workshop on explainable artificial intelligence (XAI)*. 2017, pp. 19–21.

[196] Gabriele Tolomei, Fabrizio Silvestri, Andrew Haines, and Mounia Lalmas. "Interpretable predictions of tree-based ensembles via actionable feature tweaking". In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017, pp. 465–474.

[197] Zheng Tong, Philippe Xu, and Thierry Denoeux. "An evidential classifier based on Dempster-Shafer theory and deep learning". In: *Neurocomputing* 450 (2021), pp. 275–293.

[198] Zheng Tong, Philippe Xu, and Thierry Denoeux. "Evidential fully convolutional network for semantic segmentation". In: *Applied Intelligence* 51 (2021), pp. 6376–6399.

[199] Tom Trabasso and Jake Bartolone. "Story understanding and counterfactual reasoning." In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 29.5 (2003), pp. 904–923.

[200] Matthias CM Troffaes. "Decision making under uncertainty using imprecise probabilities". In: *International journal of approximate reasoning* 45.1 (2007), pp. 17–29.

[201] Matthias CM Troffaes and Gert De Cooman. *Lower previsions*. John Wiley & Sons, 2014.

[202] Lev V Utkin. "An imprecise deep forest for classification". In: *Expert Systems with Applications* 141 (2020), p. 112978.

[203] Lev V Utkin, Maxim S Kovalev, and Frank PA Coolen. "Imprecise weighted extensions of random forests for classification and regression". In: *Applied Soft Computing* 92 (2020), p. 106324.

[204] Anneleen Van Assche and Hendrik Blockeel. "Seeing the forest through the trees: Learning a comprehensible model from an ensemble". In: *European Conference on machine learning*. Springer. 2007, pp. 418–429.

[205] Gilles Vandewiele, Kiani Lannoye, Olivier Janssens, Femke Ongenae, Filip De Turck, and Sofie Van Hoecke. "A genetic algorithm for interpretable model extraction from decision tree ensembles". In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2017, pp. 104–115.

[206] Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan E Hines, John P Dickerson, and Chirag Shah. "Counterfactual explanations and algorithmic re-

courses for machine learning: A review". In: *arXiv preprint arXiv:2010.10596* (2020).

[207] Thibaut Vidal and Maximilian Schiffer. "Born-again tree ensembles". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9743–9753.

[208] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Vol. 29. Springer, 2005.

[209] Sandra Wachter, Brent Mittelstadt, and Chris Russell. "Counterfactual explanations without opening the black box: Automated decisions and the GDPR". In: *Harv. JL & Tech.* 31 (2017), p. 841.

[210] Abraham Wald. "Statistical decision functions which minimize the maximum risk". In: *Annals of Mathematics* (1945), pp. 265–280.

[211] Peter Walley. "Inferences from Multinomial Data: Learning About a Bag of Marbles". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58 (1996), pp. 3–34.

[212] Peter Walley. *Statistical reasoning with imprecise probabilities*. Vol. 42. Springer, 1991.

[213] Soeren H Welling, Hanne HF Refsgaard, Per B Brockhoff, and Line H Clemmensen. "Forest floor visualizations of random forests". In: *arXiv preprint arXiv:1605.09196* (2016).

[214] Adam White and Artur d'Avila Garcez. "Measurable counterfactual local explanations for any classifier". In: *arXiv preprint arXiv:1908.03020* (2019).

[215] Frank Wilcoxon. "Individual Comparisons by Ranking Methods". In: *Biometrics Bulletin* 1.6 (1945), pp. 80–83.

[216] Hénoïk Willot, Sébastie Destercke, and Khaled Belahcene. "Explaining Robust Classification Through Prime Implicants". In: *Scalable Uncertainty Management: 15th International Conference, SUM 2022*. Springer. 2022, pp. 361–369.

[217] Hénoïk Willot, Sébastie Destercke, and Khaled Belahcene. "Prime implicants as a versatile tool to explain robust classification". In: *Proceedings of the 13th International Symposium on Imprecise Probability: Theories and Applications*. PMLR. 2023, pp. 461–471.

[218] Nirmalie Wiratunga, Anjana Wijekoon, Ikechukwu Nkisi-Orji, Kyle Martin, Chamath Palihawadana, and David Corsar. "Discern: discovering counterfactual explanations using relevance features from neighbourhoods". In: *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE. 2021, pp. 1466–1473.

[219] David H. Wolpert. "Stacked generalization". In: *Neural Networks* 5.2 (1992), pp. 241–259.

[220] Fuyuan Xiao, Zehong Cao, and Alireza Jolfaei. "A novel conflict measurement in decision-making and its application in fault diagnosis". In: *IEEE Transactions on Fuzzy Systems* 29.1 (2020), pp. 186–197.

[221] Philippe Xu, Franck Davoine, Jean-Baptiste Bordes, Huijing Zhao, and Thierry Denœux. "Multimodal information fusion for urban scene understanding". In: *Machine Vision and Applications* 27.3 (2016), pp. 331–349.

[222] Yaoyu Xu, Yuan Li, Yijing Wang, Dexing Zhong, and Guanjun Zhang. "Improved few-shot learning method for transformer fault diagnosis based on approximation space and belief functions". In: *Expert Systems with Applications* 167 (2021), p. 114105.

[223] Ronald R Yager. "On ordered weighted averaging aggregation operators in multicriteria decisionmaking". In: *IEEE Transactions on systems, Man, and Cybernetics* 18.1 (1988), pp. 183–190.

[224] Ronald R Yager. "On the Dempster-Shafer framework and new combination rules". In: *Information sciences* 41.2 (1987), pp. 93–137.

[225] Gen Yang, Sébastien Destercke, and Marie-Hélène Masson. "Cautious classification with nested dichotomies and imprecise probabilities". In: *Soft Computing* 21 (2017), pp. 7447–7462.

[226] Marco Zaffalon. "The naive credal classifier". In: *Journal of statistical planning and inference* 105.1 (2002), pp. 5–21.

[227] Marco Zaffalon, Giorgio Corani, and Denis Mauá. "Evaluating credal classifiers by utility-discounted predictive accuracy". In: *International Journal of Approximate Reasoning* 53 (2012), pp. 1282–1301.

[228] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*. Springer. 2014, pp. 818–833.

[229] Haifei Zhang, Benjamin Quost, and Marie-Hélène Masson. "Cautious Decision-Making for Tree Ensembles". In: *European Conference on Symbolic and Quantitative Approaches with Uncertainty*. Springer. 2023, pp. 3–14.

[230] Haifei Zhang, Benjamin Quost, and Marie-Hélène Masson. "Cautious weighted random forests". In: *Expert Systems with Applications* 213 (2023), p. 118883.

[231] Haifei Zhang, Benjamin Quost, and Marie-Hélène Masson. "Explaining Cautious Random Forests via Counterfactuals". In: *Building Bridges between Soft and Statistical Methodologies for Data Science*. Springer, 2022, pp. 390–397.

[232] Haifei Zhang, Benjamin Quost, and Marie-Hélène Masson. "Explications contrefactuelles pour les forêts aléatoires prudentes". In: *31ièmes Rencontres Francophones sur la Logique Floue et ses Applications*. 2022, pp. 27–34.

[233] Haifei Zhang, Benjamin Quost, and Marie-Hélène Masson. "Forêts aléatoires prudentes: une nouvelle stratégie de décision et quelques expériences". In: *31ièmes Rencontres Francophones sur la Logique Floue et ses Applications*. 2022, pp. 95–101.

[234] Haifei Zhang, Benjamin Quost, and Marie-Helène Masson. "Cautious Random Forests: a new decision strategy and some experiments". In: *Proceedings of the 12th International Symposium on Imprecise Probability: Theories and Applications*. PMLR. 2021, pp. 369–372.

[235] Hepeng Zhang and Yong Deng. "Weighted belief function of sensor data fusion in engine fault diagnosis". In: *Soft computing* 24 (2020), pp. 2329–2339.

[236] Xun Zhao, Yanhong Wu, Dik Lun Lee, and Weiwei Cui. "iforest: Interpreting random forests via visual analytics". In: *IEEE transactions on visualization and computer graphics* 25.1 (2018), pp. 407–416.

[237] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. "Learning deep features for discriminative localization". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2921–2929.

[238] Yichen Zhou and Giles Hooker. "Interpreting models via single tree approximation". In: *arXiv preprint arXiv:1610.09036* (2016).

[239] Maede Zolanvari, Zebo Yang, Khaled Khan, Raj Jain, and Nader Meskin. "Trust xai: Model-agnostic explanations for ai with a case study on iiot security". In: *IEEE internet of things journal* (2021).

[240] Mark H Zweig and Gregory Campbell. "Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine". In: *Clinical chemistry* 39.4 (1993), pp. 561–577.