Faculty of Science and Bio-Engineering Sciences
Department of Computer Science
Artificial Intelligence Lab

Self-labeling Grey-box Model: An Interpretable Semi-supervised Classifier

Dissertation submitted in fulfillment of the requirements for the degree of Doctor of Science: Computer Science

Isel del Carmen Grau García

October 9th, 2020

Promotors: Prof. dr. A. Nowé | Vrije Universiteit Brussel, Belgium
Prof. dr. M. M. García | Universidad Central de Las Villas, Cuba
Advisor: Dr. D. Sengupta | Vrije Universiteit Brussel, Belgium
| Queens University Belfast, United Kingdom

2021 Uitgeverij VUBPRESS Brussels University Press
VUBPRESS is an imprint of ASP nv (Academic & Scientific Publishers nv)
Keizerslaan 34
B-1000 Brussels
Tel. +32 (0)2 289 26 50
Fax +32 (0)2 289 26 59
E-mail: info@aspeditions.be
www.aspeditions.be

ISBN 978 94 6117 204 4
NUR 980
Legal deposit D/2021/11.161/119

Jury Members

Chairman

Prof. Dr. Beat Signer
Vrije Universiteit Brussel, Belgium

Secretary

Prof. Dr. Wim Vranken
Vrije Universiteit Brussel, Belgium

Promotors

Prof. Dr. Ann Nowé
Vrije Universiteit Brussel, Belgium

Prof. Dr. Maria M. García Lorenzo
Universidad Central de Las Villas, Cuba

Internal members

Prof. Dr. Tom Lenaerts
Vrije Universiteit Brussel, Belgium

Prof. Dr. Sonia Van Dooren
Centre for Medical Genetics, UZ Brussel,
Vrije Universiteit Brussel, Belgium

External members

Prof. Dr. Koen Vanhoof
Universiteit Hasselt, Belgium

Prof. Dr. Chris Cornelis
Universiteit Gent, Belgium

Abstract

In the context of some machine learning applications, obtaining data points is a relatively simple process, yet labeling them could become quite expensive or tedious. Such scenarios lead to datasets with few labeled points and a higher number of unlabeled ones. Semi-supervised classification techniques combine labeled and unlabeled data during the learning process in order to improve baseline supervised methods that use only labeled data. Unfortunately, most successful semi-supervised classifiers are complex structures that do not allow explaining their predictions, thus behaving like black boxes. However, there is an increasing number of problem domains in which experts demand a clear understanding of the decision process. Intrinsically interpretable classifiers (i.e., white-box models) are transparent structures that allow performing predictions, obtaining an associated explanation, and inspecting the model as a whole. Nevertheless, these advantages generally come at the cost of performance in terms of accuracy.

In this thesis, we propose the self-labeling grey-box model, a semi-supervised classifier aiming at providing a suitable balance between accuracy and interpretability. The self-labeling grey-box uses an accurate black-box classifier for labeling the unlabeled data and a white-box surrogate classifier for building an interpretable model. Since the self-labeling process can propagate errors, we propose two amending procedures based on class membership probabilities and certainty measures coming from the field of rough sets theory. The experimental study shows the influence of increasing ratios of labeled and unlabeled data across benchmark datasets. Moreover, we study the effect of different black-box, white-box base classifiers, as well as the two proposed amending procedures in terms of both accuracy and interpretability. The results support the interpretability of our classifier using simplicity and transparency as proxies while attaining superior prediction rates when compared with state-of-the-art self-labeling classifiers. Additionally, we illustrate the applicability of the self-labeling grey-box

classifier with preliminary results in two case studies from the field of bioinformatics. The first task concerns the detection of disease-causing genomic variants in a rare disease, while the second application tackles the prediction of early folding in proteins. Both case studies require an interpretable model able to leverage extra unlabeled data.

Samenvatting

In de context van Machine Learning is het verkrijgen van data punten meestal vrij eenvoudig, het labelen van de data daarentegen kan een duur of tijdrovend proces zijn. In dergelijke gevallen kunnen we werken met data sets waarvan slechts een aantal datapunten gelabeled zijn. Semi-supervised classificatie, laat namelijk toe om met deels gelabbelde data en deels ongelabelde data te werken, en resulteert in betere modellen t.o.v. wanneer alleen de gelabelde data zou worden gebruikt. Een nadeel van deze benadering is de complexiteit van de modellen, en het feit dat de classificatie van nieuwe datapunten niet transparant is voor een gebruiker. Maw. de modellen gedragen zich als zwarte dozen. Er is echter een toenemend aantal applicatiedomeinen waar de gebruiker een transparante beslissing verwacht. Begijpbare classificatie (white-boxes) bieden hier een antwoord. Het zijn transparante structuren die voorspellingen maken en daarbij een uitleg. Desalniettemin zijn deze voordelen meestal ten koste van performantie in termen van accuraatheid.

In dit proefschrift introduceren we een zelf-labeling en grijze modellen, d.i. een semi- gesuperviseerde classificator die een goede balans bieden tussen accuraatheid en interpreteerbaarheid. De zelf-labelende grijze dozen gebruiken een zwarte doos classificator voor het labelen van ongelabelde data en een white-box surrogaat classificator garandeert de interpreteerbaarheid. Omdat het zelf-labelings proces fouten kan propageren stellen we twee uitbreidingen voor. Een gebaseerd op gecalibreerde probabiliteiten en een tweede op basis van Rough Sets Theory. Experimenten tonen aan dat onze aanpak een belangrijke meerwaarde biedt wanneer er beperkte data is en de ratio van ongelabelde data versus de gelabelde data hoog is en dit voor verschillende benchmark datasets. Daarnaast vergelijken we ook het effect van verschillende zwarte en witte doos classificatoren en de twee voorgestelde aangepaste methoden in functie van zowel accuraatheid als interpreteerbaarheid. De resultaten tonen aan dat de classificator interpreteerbaar is en superieure predicties maakt vergeleken met state-of-the-art

self-labeling classifiers. Daarnaast illustreren we ook praktische de toepasbaarheid van de zelf-labelende grijze doos classificator aan de hand van initiÃńle resultaten in twee semi-gesuperviseerde classificatie taken uit de bio-informatica. De eerste betreft de classificatie van pathogeniciteit van genomische varianten bij een zeldzame ziekte en de tweede betreft de voorspelling van eiwitvouwing.

Contents

List of Figures

List of Tables

List of Symbols

The next list describes several symbols which are used within the scope of the thesis:

Classification and prediction nomenclature

f	Supervised classification function
F	Hypothesis space of function f
h	Scoring function
g	Semi-supervised classification function
p	Probability function
x	Instance, data point
X	Set of instances
y	Decision class
Y	Set of decision classes
i or j	Generic indexes for instances or decision classes.
c	Number of decision classes
a	Attribute, feature
A	Set of attributes
r	Number of attributes

l	Labeled instance
L	Set of labeled instances
m	Number of labeled instances
u	Unlabeled instance
U	Set of unlabeled instances
n	Number of unlabeled instances
k	Index for unlabeled instances.
w	Weight function
$L_{[y_i]}$	Subset of labeled instances with class i
$L_{[y_{min}]}$	Subset of labeled instances with minority class

Rough set theory nomenclature

DS	Decision system in Rough Sets Theory (RST)
$\mathcal{U}$	Universe of RST objects
d	Decision class in RST, later $d = y$
B	Subset of attributes in A
b	Attribute in B
t	Index for attributes in B
$X_{[y_i]}$	Subset of X with decision class i
$\underline{B}X$	Lower approximation of X, according to attributes in B
$\overline{B}X$	Upper approximation of X, according to attributes in B
$[x]_B$	Equivalence class of x, according to B
$\mathcal{P}(X)$	Positive region of X
$\mathcal{B}(X)$	Boundary region of X
$\mathcal{N}(X)$	Negative region of X
$\mathcal{R}$	Similarity relation

ψ	Similarity function
δ	Distance function
ω	Attribute weight based on information gain
ρ	Heterogeneous Euclidean-Overlap Metric function
μ	RST region membership degree
ϕ	Sigmoid function

Interpretability nomenclature

E^g	Set of rules produced by the SlGb
E^w	Set of rules produced by baseline white-box
Γ	Growth ratio measure
Υ	Simplicity measure
κ	Cohen's kappa measure
Ψ	Utility measure
α	Mixing parameter of Ψ
θ_1	Upper bound of generalized sigmoid function (simplicity measure)
θ_2	Lower bound of generalized sigmoid function (simplicity measure)
λ	Slope of generalized sigmoid function (simplicity measure)
η	Shift of generalized sigmoid function (simplicity measure)
ν	Skew of generalized sigmoid function (simplicity measure)

Other symbols for specific methods

β	Independent term in linear regression
e	Approximation error in linear regression
k	Number of neighbors for k-nearest neighbors
φ	Shapley values
ex	Explanation function in robustness measure

1 | Introduction

1.1 Motivation

The digitalization of society has enabled organizations to generate massive volumes of data. However, getting valuable insights into the collected data continues to be a challenge, even with the striking success of machine learning algorithms in solving complex prediction tasks. When the goal is learning to make predictions based on experience (i.e., supervised learning), these techniques need to learn from high amounts of labeled data. In some domains, the process of labeling data points is often expensive in terms of time, experts, or equipment.

For example, when using machine learning techniques as an aid in personalized medicine, a large amount of data can be obtained from the patients' records. These data can be from the clinical, genomic, or psychosocial dimensions and can contain valuable interactions that can be unveiled with machine learning techniques [121]. However, when tackling classification problems associated with rare diseases we have limited labeled data, or in other words, per definition we only have limited data of people tested for that disease. For example, identifying disease-causing mutations in the genomic data of a patient with a rare disease is a process that needs experts and time. Another example is the study of protein folding dynamics, which requires highly time-sensitive and complex experiments [124]. As a result of this type of scenario, the number of unlabeled data points largely exceeds the number labeled ones available for supervised classification.

Semi-supervised classification (SSC) [188] emerged as an alternative to supervised classification, aiming to leverage the unlabeled data as well. The main goal is to

improve the performance compared to using the labeled data alone in a common supervised task. However, SSC should not be seen just as a direct way of increasing performance by adding unlabeled data. It is rather a suitable alternative for the scenario where unlabeled data is available and can help obtain a classifier that reflects better the data distribution. SSC techniques rely on a group of assumptions about the distribution of the labeled and the unlabeled data. These different assumptions produce a wide variety of SSC methods reporting attractive performance in terms of accuracy.

However, the responsible use of machine learning [5] has added another variable to this equation: the interpretability component. There exist multiple application domains in which making a prediction with high accuracy is not enough. Often, it is also required to explain why or how an intelligent algorithm made a decision or took an action. That is particularly relevant for high stakes decisions affecting humans. For example, when a machine learning model is used as an aid in personalized medicine or for predicting recidivism of a convicted person. Ensuring interpretability can also be a tool for inspecting for fairness or troubleshooting a prediction model.

While state-of-the-art SSC methods are capable of producing high prediction rates, they regularly fail to provide an introspection mechanism into their decision process. This means that they perform like black boxes, thus making them less suited for application domains where interpretability is needed. This thesis tackles the lack of an interpretability component involved in the state-of-the-art of SSC. We propose a rather simple and versatile approach to solve structured SSC problems with high accuracy. The fact that our proposal is simple and involves a white-box model is considered an added value for gaining transparency. However, the accuracy and transparency of machine learning models are often conflicting objectives. Reaching a trade-off between these components is one of the most interesting challenges in today's machine learning.

1.2 Scope and Research Goals

The aim of this research is *to propose an interpretable semi-supervised classifier capable of achieving a good balance between accuracy and interpretability.* To fulfill our goal, we build upon self-labeling methods, which are ensemble classifiers that use a base model for predicting unlabeled data, assuming these predictions are correct to some extent. Self-labeling offers room for incorporating interpretability into their model. However, they are prone to propagate errors when assigning labels to unlabeled data. This problem is taken into account in our solution by proposing amending strategies. In addition, a proper evaluation of our proposal in terms of accuracy and interpretability is needed. Therefore, the research aim can be divided into several research objectives that address the aforementioned challenges:

1. To propose a general architecture of an interpretable semi-supervised classifier, based on the self-labeling technique.
2. To propose strategies for amending the errors generated in the self-labeling process.
3. To propose measures for evaluating and comparing interpretability in the context of the proposal.
4. To evaluate the influence of different choices of base classifiers on the accuracy and interpretability of the proposed method.
5. To compare the prediction capability of the proposal with state-of-the-art self-labeling methods in benchmark datasets.
6. To illustrate the usability of the proposal in case studies of semi-supervised classification where interpretability is a requirement.

1.3 Overview of the Proposed Solution

In this research, we build upon the state-of-the-art in SSC. After analyzing the potential for obtaining interpretability of different families of methods, we rely on the self-labeling technique [177, 96, 158]. In self-labeling, a base classifier is trained on limited labeled data points and subsequently used for predicting the labels of the unlabeled data points, forming an enlarged dataset. This wrapper behavior encapsulating a base classifier shows a clear connection with the use of a white-box classifier as a global surrogate for gaining interpretability [64, 65, 53, 7]. Both strategies encapsulate a base classifier for optimizing different objectives. While self-labeling attempts to improve the accuracy in a semi-supervised setting, the global surrogate trains a white-box model on mimicking the predictions of a base black box for improving interpretability. This resemblance is the starting point of our proposal, combining a black-box classifier for the self-labeling with a white-box one for gaining interpretability in what we call a *self-labeling grey-box* (SlGb).

The assumption of self-labeling that their predictions are correct can potentially propagate errors in the enlarged dataset. To overcome this drawback, we propose two amending procedures that weigh the importance of each data point in the learning process. The first amending strategy considers the class membership probability estimated by the base classifier performing the self-labeling. The second amending is not only limited to the class noise produced during self-labeling but also tackles the inconsistency that emerges in the entire enlarged dataset. Using rough sets theory (RST) [128] for partitioning the decision space in regions of data with positive, negative, or

hesitant evidence towards a given class, helps weight each data point according to their membership to these regions. These weights are used for guiding the white-box classifier in learning from the most confident information.

In addition, we propose some measures to evaluate the performance of our proposal in terms of accuracy and interpretability. These interpretability measures are based on using transparency as a proxy, and specifically, the number of rules that the grey box produces as an indicator of the size of its structure. These measures are based on a functional analysis approach [40] for evaluating interpretability, where no experts are involved.

Finally, we illustrate the usability of SlGb through two SSC tasks. The first task concerns the prediction of disease-causing (pathogenic [87]) genomic variants from a rare disease [74]. The second case study involves the prediction of early folding residues [135] in proteins. In both case studies, labeled data is difficult to obtain either because it comes from a manual process involving experts or it needs complex experiments. We show how different configurations of our proposal perform for these applications.

1.4 Main Contributions

This research comprises four main contributions, namely: 1) the SlGb classifier which achieves a good balance between accuracy and interpretability, 2) the amending strategies for the self-labeling process that guide the grey box towards learning from more confident data points, 3) the interpretability measures for rule-based grey boxes, and 4) the application of the SlGb in real application problems from the bioinformatics domain. These contributions are detailed as follows:

1. An interpretable semi-supervised classifier named *self-labeling grey-box* is proposed. This classifier uses a black-box base classifier for labeling the unlabeled data, using a self-labeling approach, and obtaining an enlarged dataset. Later on, a second white-box classifier is trained for mimicking the predictions of the black box, forming a grey-box model. This combination incorporates interpretability in the semi-supervised classifier.

2. Two amending procedures are proposed for correcting the misclassifications of the self-labeling in the enlarged dataset. The first one is based on the class membership probabilities estimated by the black box, whereas the second one considers the class noise that can emerge in the enlarged dataset. Using RST for the second amending allows relying on positive, negative, and hesitant information for guiding the grey box to learn explanations for the most confident data points.

3. Three measures related to interpretability are proposed. These measures are based on the number of rules generated by the grey box as an indicator of the size of the structure. The first measure is applicable in the context of self-labeling and is the relative growth in structure compared to a white-box baseline. The second measure uses a growth curve for estimating the simplicity of the model and can be easily generalized to any white-box classifier. The third measure combines performance and interpretability with a utility function.

4. Two case studies are used for illustrating the usability of the proposed SlGb in real applications from the field of bioinformatics. In both case studies, we analyze the performance of several configurations of SlGb. We show that, for these applications, SlGb is a good-performing predictor that is also transparent and allows obtaining explanations and an interpretable view of the predictions. In the discussion with experts, some rules obtained by SlGb were identified as expected patterns supported by domain knowledge, while others showed meaningful new patterns that should be investigated further.

1.5 Thesis Organization

The rest of this thesis is organized as follows. Chapter 2 revises the state-of-the-art on interpretable machine learning. It covers the fundamental concepts, methods, measures, and open problems in this re-emerging field. This chapter serves as a reference for the terminology on interpretability that is used in the rest of the dissertation.

Chapter 3 studies the state-of-the-art of SSC, grouping each family of methods based on assumptions that these methods need. This chapter contributes our view in the interpretability potential of SSC methods which is a frequently neglected topic in semi-supervised learning reviews.

Chapter 4 presents the main contribution of this research: the self-labeling grey-box method. In addition, this chapter also discusses the amending procedures proposed for the self-labeling correction. Additionally, we outline the requirements of the base classifiers that are part of SlGb.

Chapter 5 evaluates the performance in terms of accuracy and interpretability of SlGb in a wide collection of benchmark data. The experiments include exploring different combinations of base classifiers and amending procedures, the influence of the number of labeled and unlabeled data points, and comparing its performance against other four state-of-the-art self-labeling classifiers.

Chapters 6 and 7 describe the application of SlGb in the prediction of disease-causing genomic variants of a rare disease and the early folding residues in proteins. Both are case studies from the bioinformatics field which explore the usability of SlGb in real applications.

Chapter 8 discusses the results and concluding remarks on this research. We outline interesting future research directions that were identified through the course of this research. Additionally, Appendixes A, B, and C complement the experimental results from Chapters 5 and 6. Finally, the publications linked to this research and the references cited through the document are listed.

2 | Interpretability in Machine Learning

This chapter introduces the state-of-the-art on interpretable machine learning and motivates the reasons to advocate for a broader explainable artificial intelligence concept. Then, it focuses on the progress of machine learning interpretability, more specifically on concepts, methods, measures and open problems in the field. This chapter serves as the definition of the framework for interpretability that will be referred to in the rest of the dissertation.

2.1 Explainable Artificial Intelligence

In recent years, artificial intelligence and particularly machine learning fields have experienced a clear increase in interest from outside academia. The unprecedented performance of machine learning algorithms in solving complex tasks from high volume of structured and unstructured data caught the attention of industry, governments and society. The use of sub-symbolic ensembles or deep learning techniques led to this massive capacity of performance in very specific tasks. However, in contrast to what is sometimes wrongly spread by media, current artificial intelligence solutions are far from thinking by themselves or having some kind of consciousness.

Like any other technology or scientific breakthrough, it needs regulation and control for ensuring safe, responsible and meaningful use. This is evidenced by major governmental efforts such as the well-known European Union's right to explanation

regulation [62] or United States' DARPA explainable artificial intelligence (XAI) research program [66]. Therefore, the research community should aim to provide artificial intelligence algorithms with the capacity of explaining their predictions or actions, especially when these decisions affect humans' lives.

The high performing sub-symbolic ensembles and deep learning techniques have the disadvantage of being less interpretable compared to more symbolic approaches that were integrated into industry solutions in the past (e.g. rule-based expert systems). The need for interpretability, accountability and fairness for the responsible deploying of artificial intelligence in society constitutes the main fuel of the XAI research field.

2.2 Interpretability in Machine Learning

The context of this work focus on the interpretability of machine learning techniques. Initially, the notion of machine learning black boxes, as opposed to white boxes, was taken from engineering. Black boxes refer to models which have been learned from data and are difficult to understand at a global level. White boxes refer to models which are built based on prior knowledge of the problem domain and/or learned from data but their structure allows for a degree of interpretability since pure white boxes rarely exist [116]. Grey-box models are somewhere in between the spectrum, referring to models that provide some degree of interpretability. Generally speaking, black-box models tend to be more accurate than white-box ones since they are able to approximate better more complex functions, leading to a known trade-off between accuracy and interpretability.

Recently, several efforts have been done in further defining and clarifying the terminology around interpretability in machine learning. In a first approach, Lipton [101] makes a distinction between the transparency notions of a machine learning model (on the global model, parameters and algorithm level) and the post-hoc methods for gaining on interpretability. Doshi-Velez and Kim [40] define a broad concept of interpretability as *"the ability to explain or to present in understandable terms to a human"* and focuses on general guidelines for evaluating interpretability with and without human intervention. Miller [110] also distinguishes between interpretable models and the generation of explanations arguing that the XAI field should build upon research on explanations from the philosophy, psychology, and cognitive science fields. Barredo et al. [5] remark the dependency on the users when generating explanations. Other concurrent review papers [1, 56, 39, 5] elaborate more detailed (re-)definitions which sometimes result in overlapping or contradictory concepts. Therefore, the intention of the remainder of this section is to set the terminology clear for the scope of the dissertation.

Based on the current state-of-the-art, this work proposes the following definitions:

Explanation: A justification given for a model prediction.

Interpretation: The action of giving an explanation that can be directly mapped to the problem domain.

Explainability: The ability of a model to provide explanations.

Interpretability: The ability of a model to provide interpretations that allows a human to reason about the model as a whole while mapping it to the problem domain.

Explanations can be composed by cognitive units [40] such as raw features, prototypes, derived features, instances, etc. Explanations can be formed in structured ways such as rules, hierarchies, similar examples, etc., which determine its expressive power [143]. When building an interpretable model or an explanations model, the type of explanations being generated must comply with the purpose and the users of the model [5, 73].

Transparency is defined as a property of a machine learning model that serves as a proxy for interpretability. The levels of transparency of a machine learning model [101] can be characterized as:

Algorithmically transparent: The condition of a model of being easy to inspect by a human, thus allowing to obtain an output from an input in a transparent process.

Decomposability: The condition of a model of being able to explain each of its parts (e.g. input, parameters) without the need for additional tools, i.e. they can be directly mapped to the problem domain.

Simulatability: The condition of a model of being reproducible or thought about as a whole by a human, *with a reasonable amount of effort*. A simulatable model needs to be algorithmically transparent and decomposable.

The third property is subjective in its definition. Therefore, this work considers a model to be interpretable when it complies with the first two properties and with the third one to some extent, depending on the purpose and the target users of the model.

The next section further explores the differences between intrinsically interpretable and post-hoc methods for generating explanations while reviewing the state-of-the-art.

2.3 Methods for Interpretability and Explainability

Similarly to [101, 112, 5], this research makes a clear distinction between intrinsically interpretable methods relying on transparency and post-hoc methods for generating

explanations. Additionally, the section locates global surrogate models on a special category in between.

2.3.1 Intrinsically Interpretable Models

According to the aforementioned definitions, a model is considered to be interpretable if its level of transparency allows obtaining a transparent prediction, the components can be mapped to the task domain and the model can be managed or thought about as a whole to some extent. Next subsections describe well-known machine learning techniques for obtaining interpretable models and how this definition of interpretability applies to each of them.

Linear Regression

Linear regression models [70] predict the dependency of a continuous variable y as a weighted sum of a set of features A using a set of instances X as:

$$y = \beta + \sum_{a \in A} w(a)x(a) + e \tag{2.1}$$

where $w(a)$ is the weight or coefficient for attribute a, $x(a)$ is the value of instance x for the attribute a, β is the intercept and e is the error of the approximation. The simplicity of the linear relation described by the model makes it algorithmically transparent, as it is possible for a human to inspect the process of getting an output y from a given input x.

Each of the components of the model has a given interpretation adding decomposability. However, the notion of interpreting the weights or coefficients $w(a)$ directly as feature importance is incorrect. The interpretation of each weight is subject to its statistical significance given by the confidence interval associated with it. Statistically significant weights are interpreted as: "an increase in the feature a by one unit changes the estimated outcome y by $w(a)$ units when the rest of the feature values are fixed". In case the feature is nominal instead of being numerical, then the interpretation is subject to a reference category [112]: "a change from the reference category to another category in feature a changes the estimated outcome y by $w(a)$ units when the rest of the feature values are fixed". The intercept β can be interpreted as the prediction of the model when all numerical features are zero-valued and the categorical ones are at their reference value. This can be useful when the feature space is standardized and the null instance represents the mean of the dataset [112]. Finally, e represents the approximation error of the model concerning the ground truth.

Regarding the simulatability of linear regression, sparsity (i.e. a small number of non-zero coefficients) plays a fundamental role. Several optimization algorithms, e.g.

Lasso heuristic, have been proposed for learning more sparse regression models [71]. A fewer number of coefficients to take into account allows the user to reason about the model as a whole more easily.

Logistic Regression

Logistic regression [70] is an extension of linear regression for classification tasks with two possible decisions, where the output y is a probability of belonging to one of the two classes. The sigmoid function is used for obtaining the probabilities:

$$p(y = 1) = \frac{1}{1 + \mathrm{e}^{-(\beta + \sum w(a)x(a))}}. \tag{2.2}$$

As an extension of linear regression, its interpretability analysis is similar regarding the three properties. However, the interpretation of the coefficients is less straightforward since they influence the predicted probability in a non-linear way. The change in the coefficient will now affect the odds ratio of one class over the other, a full mathematical derivation of this can be found in [112]. The interpretation of a weight change can be formulated as: "an increase in the feature a by one unit, changes the estimated odds of $y = 1$ by a factor of $\mathrm{e}^{w(a)}$ units, when the rest of the feature values are fixed". For a categorical feature, the change relies on a reference value and the multiplicative increment in the odds is also expressed as a factor of $\mathrm{e}^{w(a)}$.

A substantial disadvantage of linear and logistic regressions is their assumption of the absence of multicollinearity. When multicollinearity is present, the precision of the weights is less trustful since it is more difficult to attribute the changes between two correlated features.

Decision Tree

Decision tree models have a flow-like structure where nodes represent subsets of data or regions. Each internal node tests an attribute a, where one branch is assigned for each possible value (nominal attribute) or a threshold is determined (numerical attribute) obtaining generally two branches. The nodes are subsequently split until a leaf node is reached. All instances must belong to a leaf and the prediction y for an instance x can be obtained by traversing the tree from the root to the assigned leaf:

$$y = \sum c_r I(x \in R_r) \tag{2.3}$$

where R_r represents the region of a leaf node r, I is function returning 1 when $x \in R_r$ or 0 otherwise and c_r is the expected prediction (mean or majority vote) for all instances in the leaf r. There are several algorithms for learning the structure of the tree, for example C4.5 [134] and CART [19]. C4.5 is used for growing decision trees

when solving classification tasks where both categorical and numerical features are present. At each step it greedily chooses the node that maximizes the information gain to maximize the "purity" of the nodes, therefore minimizing the classification error. CART is similar to C4.5 but it also supports regression tasks by minimizing the variance of y. CART builds binary trees by splitting each node using a threshold for numerical attributes or subgroups for nominal ones. When nominal attributes are present, C4.5 tends to build shallower tree structures than CART.

Decision trees are algorithmically transparent since it is possible to obtain the prediction of an instance by traversing the tree from the root to the relevant leaf. It is considered a decomposable model since the decision nodes and branches represent features and their possible values or thresholds of the problem domain. Decision trees can be easily translated into a set of *if-then* rules with the form: "if $condition(a_1)$ and $condition(a_2)$ (...) and $condition(a_r)$ then $y = prediction$". In fact, the whole tree can be represented as a disjunctive normal form, where a target value is true if and only if the input attributes satisfy one of the paths leading to a leaf with the target value [146]. Therefore, decision trees can be considered to be simulatable as long as the structure is manageable by a human as a whole, otherwise, its interpretability is confined only to algorithm transparency and decomposability. The number of leafs or rules can help judge whether or not the structure is manageable, which is subject to the target users and the purpose of the model. The depth of a path in the tree can be seen as a local measure representing the maximum number of features needed to produce an explanation for the prediction of a given instance. The usefulness of these explanations depends on the accuracy and the support of the rule. More details on measures will be discussed in Section 2.4.

Decision Lists

Decision lists are sets of *if-then* rules where the condition is a conjunction of feature evaluations and the conclusion is the prediction of the target value. For an unseen instance, decision lists are evaluated in order such that a default rule with no condition is used when no other rule applies. The entire list can be analyzed as a whole by interpreting conditions from more specific to more general.

Decision lists is a widely studied field with several algorithms for inducing rule lists [54]. Sequential covering is a common divide-and-conquer strategy for building decision lists. Here, a rule is induced from data and the covered instances are removed before inducing the next rule, until all instances are covered by rules or a default rule is needed. PART decision lists [49] is one of the many models implementing this strategy, where rules are iteratively induced as the most covered one from a partial decision tree. RIPPER [28] is another representative algorithm that uses reduced error

pruning and the minimum description length heuristic to replace or revise the induced rules.

Similarly to decision trees, decision lists are algorithmically transparent and easily decomposable since the explanations that can be generated are rules using features and values of the problem domain. Decision list algorithms tend to generate more compact rule sets than those obtained from a decision tree while being equally expressive [112, 49], therefore making them more simulatable.

Naive Bayes Classifier

Naive Bayes classifier [3] models are a probabilistic classifiers based on Bayes' theorem assuming a strong feature independence. The class y is assigned to an instance x by computing the class probability times the feature probability given that class:

$$p(y_i \mid x) = \frac{1}{Z} p(y_i) \prod_a p(x(a) \mid y_i) \tag{2.4}$$

where Z is a normalizing constant based on the values of the features of x.

Naive Bayes classifiers are algorithmically transparent since it is possible to induce the classification of an instance by taking the class with the highest probability from the above function. It can be considered to be decomposable since the parameter Z is known and dependent on values from the domain. Conditional probabilities are directly related to the problem domain and help elucidate how much a feature contributes to a certain class prediction. However, the understanding of conditional probabilities is subject to the target users of the model. A possible interpretation of a prediction of Naive Bayes classifier is "y_i is the most probable value of y for x because $x(a_1) = v_1$ and $x(a_2) = v_2$ (...) and $x(a_r) = v_r$ have a high probability when $y = y_i$, assuming independence of all features". Regarding the simulatability, it is less clear to think about the model as a whole, but it is reproducible for a small number of features.

k-Nearest Neighbor

k-Nearest Neighbors (kNN) [3] is an instance-based learning technique where an instance x is given a prediction based on the expected y value (mean or majority vote) of its k nearest instances in the feature space, relying on a distance metric. The resemblance with the human mechanism of decision making based on past events makes this technique intuitively interpretable. However, kNN can be seen as a special case in the sense that no actual model is built through learning, i.e. it uses lazy learning. Alternatively seeing the metric and the examples as the model, then its transparency heavily depends on the number of features, the number of neighbours and the distance

metric. The distance function might not be completely transparent, especially if the number of features is large, affecting its decomposability.

The simulatability can be affected by a large value of k. kNN cannot be thought about as a whole, but rather as a generator of example-based explanations (see next section). The type of interpretation obtained through kNN is "the prediction for y is y_i because x is similar to these other k examples". In this case, the interpretations are at different granularity level compared to the previously described models. They express relations among instances instead of features of the problem. This is the case of the explanations that can be obtained by other granular models [43].

2.3.2 Generating Explanations with Post-hoc Methods

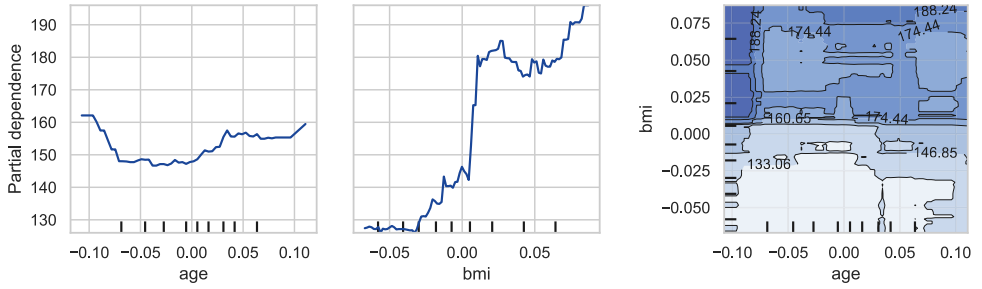
Post-hoc techniques focus on generating explanations (i.e. explainability) for already learned black-box models. The scope of the explanations can be *local*, i.e. for one instance or a subset of instances, or *global*, that is for the entire model [112]. Post-hoc techniques can produce different kinds of explanations such as visualization of features dependency, feature importance metrics, explanations with (counter) examples, etc. A wide review covering model-specific post-hoc approaches, including those tailored to deep learning techniques, is presented in [5]. Next subsections focus on some representative model-agnostic approaches.

Dependency Plots

Dependency plots are post-hoc methods that can be applied to a previously learned black-box model. They attempt to generate explanations by visualizing the relationship between features and the prediction. These visualizations can be either global or local, but only relying on pairs of input and outputs of the black-box model.

Partial dependency plots (PDP) [51] show the marginal effect of up to two features on the prediction generated by a black-box. It computes a partial function that represents the variation in the prediction for one or two features considering the average of the rest of the features. Assuming independence of the features, the explanation generated by the plot is "how the average prediction of y changes when the feature a changes", see Figure 2.1 for an example. A major limitation of PDP is the assumption of independence in the features, which leads to include average values of other features that are unrealistic, given the value of the feature being analyzed. Besides, the fact that the contribution is being averaged from the entire dataset could hide heterogeneous contributions from different instances which are canceling each other [112].

Individual Conditional Expectation plots (ICE) [58] aim to correct the drawbacks of the global character of PDP plots. This method visualizes the dependency of the



(a) One-dimensional PDP (b) Two-dimensional PDP

Figure 2.1: Partial dependency plots showing the individual (a) and joint (b) influence of *age* and *body mass index (bmi)* in diabetes progression. The plots are generated using a toy dataset from [42]. Features are standardized and the prediction variable is a numerical indicator of the progression of the disease compared to a baseline. It can be easily seen that the average prediction of the progression of the disease increases with the body mass index.

prediction variable for individual instances. Similarly to PDP, each instance line can be computed by fixing the rest of the features to a reference value and then varying the value of the feature being studied. The reference values are possible combinations of values of the fixed features, thus creating synthetic data for computing the prediction of the underlying black-box model. The explanations of this plot are similar to PDP, however, by looking at individual predictions instead of average lines, the heterogeneous contributions from different instances can be unveiled. It should be noticed that ICE could show unrealistic data points which do not take place in the real joint feature distribution.

Feature Importance

Feature importance –also known as variable importance– was first proposed by Breiman [18] for random forests and was later extended by Fisher et al. [48] to make it model agnostic. The underlying idea is to calculate the influence of a feature in the prediction error of the model by randomizing its values along with the instances. In that way, the generated explanation has the form "feature a is more (less) important since the black-box incurs in more (equal or less) prediction error when a is shuffled". An alternative method consists in learning two models with and without the feature and compare their results [55]. However, in the latter way, the measure does not reflect how much either individual model relies on the feature.

The extension of variable importance to model class reliance (MCR) by [48] aims to mitigate the Rashomon effect [18], i.e. the fact that different models can achieve attractive performance while relying on different features. MCR explains then "the highest and lowest degree to which any well-performing model within a given class may rely on a variable of interest for prediction accuracy". This method can be especially useful when models are private and cannot be inspected directly, but there are records of their outputs. Therefore, by approximating the performance of the black-box model with a class of other models, the likely reliance on certain features can be estimated.

When permuting the values of a feature along with the instances, feature importance measures also quantify the interactions with other features. However, similarly to PDP, unlikely instances can appear when a feature is permuted and two or more features are correlated. Furthermore, a highly correlated feature can lead to a decrease in the relative importance of both features (the original one and the correlated one) by "splitting" the importance between them [112].

Local Surrogates

Local surrogate models aim to explain the prediction of individual instances by locally approximating an interpretable model. Therefore, it profits from the intrinsic interpretability of other machine learning models. Here, the model acting as a surrogate must be a good approximation of the black-box model for a subset of instances (and possibly a subset of features), but not necessarily of the entire black-box. Local interpretable model-agnostic explanations (LIME) [141] is a specific implementation of this idea, where the instances are sampled with normal distribution and weighted according to their distance to the instance to be explained. Later on, a linear regression model is built on the synthetic dataset, from which feature contributions can be examined as seen with linear regression (see Section 2.3.1). However, a major difference with linear regression relies on the fact that the regression coefficients are not based on real data points, but on synthetic weighted data points, therefore the interpretation is not valid for the entire domain.

A weak point of LIME is the need to define the distance metric or the size of the neighborhood to be deemed relevant for the instance that is being explained. The resulting explanations are very sensitive to the choice of these metrics [4]. Similar to other methods, the sampling of instances without taking into account the correlation among features can result in unrealistic data points being used for building the explanations. A strong point of local surrogates is that they provide the explanations via another model instead of feature summarization measures, which can lead to a local replacement of the black-box if needed.

LIME can be also used for text and image data, where words or groups of pixels are perturbed by removing them from the instance, respectively. This method is espe-

cially fitting for the scenario where a black-box is trained on derived non-interpretable features (e.g. word embeddings) and the interpretable local surrogate is trained on the original interpretable ones (e.g. words). This facilitates to build understandable explanations while the prediction is approximated in a more complex feature space.

Shapley Values

Shapley values [151] is a post-hoc approach that comes from the field of coalitional game theory and shows how much a feature brings in for a prediction, in addition to a given subset of features. This can be calculated individually for an instance or globally as an average over all instances. This average needs to be computed over all possible combinations of features subsets (or coalitions), therefore it can be computationally expensive.

More formally, a Shapley value ϕ_a represents the importance of the feature a when included in the model f:

$$\varphi_a = \sum_{B \subseteq A \setminus \{a\}} \frac{|B|!(|A| - |B| - 1)!}{|A|!} \left(f_{B \cup \{a\}}(x_{B \cup \{a\}}) - f_B(x_B) \right) \tag{2.5}$$

where A is the original feature set, B are possible subsets of $A \setminus \{a\}$. Shapley values comply with the property of efficiency, i.e. the sum of all feature contributions equals the difference of the prediction for x and the average prediction. This method is also the only one that complies with symmetry, linearity and null-player properties simultaneously with efficiency, which implies that the feature contributions to the difference of a prediction with the average prediction are being fairly distributed [112]. However, similarly to feature importance methods, Shapley values can generate unrealistic data instances when features are correlated due to the marginalization of the missing features on each coalition. Another drawback is that the explanation consists of the feature contributions for all features, which could be less clear if the number of features is high.

An alternative that also allows computing Shapley values for a subset of features is the Shapley additive explanations (SHAP) method [104]. Interestingly, SHAP is also applicable for groups of features, e.g. a group of pixels in an image. Lundberg and Lee [104] build upon Shapley values theory and rewrite them as an additive feature attribution method, i.e. a linear model. They propose a model-agnostic method for estimation of the Shapley values called Kernel-SHAP using local linear regression inspired in LIME for estimating the values. They also propose model-specific variants (e.g. for deep learning). Recently, in [106] they propose Tree-SHAP, a more efficient variant of SHAP for decision trees, random forests and gradient boosted trees which uses the number of training examples traversing the tree to each leaf to represent the

background distributions. A software library[1] for the use of this method containing a pool of attractive plots of the results supports the explainability by adding the visualization resource. For example, the force plot portrays the "force" that each feature executes (in both directions) for moving the actual prediction of an instance away from the average prediction of the dataset (see Figure 2.2).

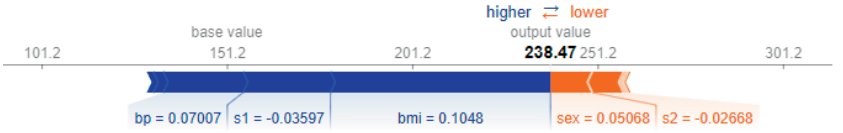


Figure 2.2: Force plot of SHAP values of different features predicting the diabetes progression for a specific patient with linear regression. The plot is generated using a toy dataset taken from [42]. Features are standardized and the prediction variable is a numerical indicator of the progression of the disease compared to a baseline. The average prediction of the dataset is represented as the base value. The bars represent the contribution of each feature in obtaining the prediction. Blue bars move the prediction to a higher value and orange bars move the prediction to a smaller value.

In general, SHAP is a very flexible model computing individual and fair feature contributions for specific instances. It can be easily extended to the entire dataset by averaging the absolute Shapley values per feature across the data, obtaining a feature importance measure.

Example-based Explanations

Example-based explanations are another family of post-hoc methods which build explanations with instances (either synthetic, real or prototypes), thus leveraging the intuitiveness of human case-based reasoning. The first of these methods, k-nearest neighbors, was already covered as a special case of intrinsically interpretable models in Subsection 2.3.1.

Counterfactual explanations [163] is another approach which uses alternative instances for formulating counterfactual *if-then* rules, e.g. "if feature $a = v_2$ instead of v_1 the prediction changes to the desired value y_i". Another way of reading this kind of explanations is "if feature $a = v_2$ had not occurred, then y_i would not have been predicted". Finding a counterfactual explanation for an instance involves optimizing a loss function with two components: the distance of the counterfactual instance to the instance to be explained and the distance of the counterfactual prediction to the

[1] https://github.com/slundberg/shap

alternative desired prediction. In this way, a counterfactual instance that is close to the original instance with minimal alterations will lead to the alternative desired prediction. It should be noticed that equally good counterfactual instances proposing different changes to obtain the desired prediction can be found. This Rashomon effect (see Subsection 2.3.2) reflects the complexity of the real domain and therefore the explanations should be evaluated or selected by domain experts. Recently, tools for obtaining counterfactual explanations have been proposed in [59, 114].

2.3.3 Global Surrogates

The aforementioned post-hoc techniques while generating justifications of the predictions do not necessarily enhance the transparency of the black-box model itself. An idea in between is to use intrinsic interpretable models as global surrogates by learning from the predictions of the already trained black box. The prediction function from the black-box will be approximated by another function obtained with a more interpretable model. The goal is to enhance the transparency of the final model as a whole while trying to mimic the performance of the original one as much as possible. By building an entire model view instead of providing statistics or examples, we consider that this strategy is more in line with interpretability rather than explainability. Here, a trade-off in preserving interpretability and accuracy naturally arises.

The choice of surrogate white-box could be any intrinsically interpretable model (see Section 2.3.1) which can be trained using the data points and the black-box predictions and satisfies the type of interpretation desired for the task at hand. The surrogate model can be trained on all (weighted) data points, subsets or prototypes. However, this could have an implication in the global character of the interpretability, since a selection of rather local instances could lead to a local surrogate. During the course of this research, the global surrogate models have being concurrently published under different names: model simplification, model distillation, proxy model, mimic model, hybrid models, twin-system approach, glass-box models or grey-box models.

There are multiple emerging model-specific implementations of global surrogates. For example, in [22] the authors propose to mimic the learning of two types of deep neural networks (variational autoencoders and long short term memories) by training a gradient boosting tree, which they claim to be interpretable. However, the ensemble character of the chosen white-box component limits the interpretability to the observation of the most important features in one of the trees that make up the ensemble. In contrast, in [69] the authors use gradient boosting trees as a black-box model and try to obtain a decision model that mimics the decision regions found by the black-box. They create binary features for representing the decision regions and try to approximate the new decisions boundaries by optimizing the prediction and the regions using an expectation-maximization algorithm. The final model is expressed

as rules representing each region. Although this could be a promising approach for improving the interpretability of tree ensembles, the results of the paper are based on only two datasets. A more general implementation is portrayed in [7] where random forests, neural networks and the control policy of a reinforcement learning problem are approximated with a CART decision tree using a sampling function for selecting instances to be learned by the surrogate model.

More on the hybridization direction, another approach on combining neural networks and decision trees is proposed by Frosst and Hinton in [53], where the inner nodes of a soft decision tree represent binary decisions made by a logistic regression model. The parameters of the nodes (called filters) are then learned using a loss function that minimizes the cross-entropy between each leaf using mini-batch gradient descent. For explaining the prediction of an instance the learned filters can be examined through the path to the leaf. However, the hierarchy of nodes in the tree represents a hierarchy of filters combining features rather than features alone, which makes it difficult to inspect further than two levels.

Other approaches are linked with security, fairness[2] and other fields of XAI. For example, [126] is oriented to the field of adversarial machine learning. This approach computes k-nearest neighbors on each layer of a deep neural network for estimating the lack of support of a prediction in the neural network. It relies on the principle that neighboring instances with the same prediction offer confidence in the current prediction. At the same time, they leverage the explanation-by-example ability of k-nearest neighbors with image data. Linked to the field of fairness, the authors in [156] propose to train a surrogate model based on the predictions of a black box from which the underlying model is not known (i.e. private model). By comparing the surrogate with an alternative transparent model of the same class trained on available data with potentially different features from the black-box model (audit data), they compute the likelihood of the audit data is missing features that the black-box model used for prediction. This serves as a way of auditing the features used by the private black box.

In general, an extra advantage of global surrogates is the ability to compare different black-boxes for the same task and gain insight into why some black boxes perform better than others by using the holistic view of an interpretable model instead of feature summaries alone.

[2]Fairness in AI studies individual and group fairness notions based on protected attributes. It attempts to detect and mitigate the bias coming from data using fairness metrics and pre- or post-processing algorithms.

2.4 Evaluation and Measures

While the prediction performance of a machine learning algorithm is a well-defined concept with several measures for comparing different techniques, the interpretability measures remain as a less standardised and formalized terminology. This section compiles the first steps of the community into proposing forms of evaluation, explanations desiderata and quantitative measures for comparing the interpretability of machine learning techniques or the quality of individual explanations from different perspectives.

2.4.1 Interpretability Evaluation Frameworks

Doshi-Velez and Kim [40] formalize three evaluation frameworks for interpretability.

The first one, called *function level evaluation*, proposes to use a class of model that is known to be interpretable, i.e. any method from Section 2.3. The evaluation measure would be a characteristic of that model that serves as a proxy, e.g. sparseness of a linear regression model or the number of rules in a decision list. This type of evaluation is generally easier to perform in scenarios where involving humans is costly or unethical.

The second type of evaluation called *human level evaluation* as its name indicates involves humans, but not necessarily experts in the domain that could be expensive to recruit. For this type of evaluation, a simplified task is defined and lay humans are provided with pairs of explanations to choose the one they prefer. Another alternative for testing the understanding of the user about a model is providing an explanation and an input and request the user to find the output.

Finally, the most challenging type for evaluation is *application level evaluation*, which is conceived for the specific application task and carried out with human experts in the domain. Here, the explanations are evaluated by contrasting them to human-generated explanations on the same task. The quality of the explanation should be measured in terms of its end purpose, for example, identifying new relationships or errors. This definition of interpretability evaluation in incremental stages can help deploy an interpretable machine learning solution for a particular application task while testing its interpretability from the beginning.

2.4.2 Desired Properties of Explanations

Another advance in the topic of evaluating interpretability is defining what constitutes a good explanation. In this direction, the work of Robnik-Sikonja and Bohanec in [143] gathers the desired properties of individual explanations:

Accuracy: How well the explanations generalize to unseen test data. This is especially useful when explanations are intended to replace the underlying black boxes for future predictions.

Fidelity: How well the explanations approximate the prediction of the underlying black-box model.

Consistency: How much explanations built from different black-boxes differ, provided they are trained on the same task and their predictions are similar.

Stability: How much explanations for similar instances differ, provided they are built with the same underlying black-box. This property is also extendable to explanations generated by intrinsically interpretable models.

Comprehensibility: How well do humans understand the explanations. This property can be addressed from different angles and there is no consensus on a model-agnostic measure to date.

Certainty: How the explanations reproduce the certainty of the predictions of the underlying black-box. This property can be measured when the black-box also provides confidence measures of its predictions.

Degree of importance: Whether the explanations provide a degree of importance of the features or the parts that compose the explanations.

Novelty: Related to certainty, how the explanations cover a data point coming from a region of data not covered in the training data. With high novelty comes a high risk of the model to be inaccurate and the certainty to be low.

Representativeness: How many instances the explanation covers, or whether the explanation is local or global.

The properties accuracy, fidelity, consistency, stability, certainty and novelty are measurable based on accuracy, support or confidence measures that are already established in machine learning. The degrees of importance and representativeness depend on the type of the explanation model being used, e.g. feature importance or SHAP will provide importance measures while LIME has local representativeness. However, comprehensibility (also referred to as understandability, readability, etc) is, in our opinion, strongly dependent on the cognitive units and the structure that forms the explanation. For example, it could be measured as the number of coefficients in a linear model or the number of rules in a decision list, but these measures are still limited to the class of the model and no agnostic measures are proposed to date.

In addition, in [143] define *translucency* and *portability* properties of interpretable methods. Translucency describes to what extent the generated explanations rely on

the parameters of the model being explained in contrast with manipulating inputs and observing outputs (model-agnostic). Therefore, translucency aligns with the definition of transparency in Section 2.2. Portability is also related to the difference between intrinsically interpretable models and model-agnostic explanations since methods with low translucency tend to have more portability. For example, global surrogates can be applied in general to any class of black-box model.

The study published in [110] reviews how people create, perceive and evaluate explanations from the philosophy, psychology, and cognitive science points of view. The author argues that explanations are mostly contrastive, i.e. humans prefer counterfactual explanations that contrast their situation with the desired one, instead of the entire set of conditions that led to a prediction. Therefore, explanations are also selected (with the consequent bias) since humans prefer to see one or two possible causes of the outcome than the entire set of possible ones. This oversimplification of the explanations must be handled with care and fairness should be specially taken into account if those guidelines are followed. Finally, the author also emphasizes that explanations are a social process and are presented as part of a conversation or interaction, therefore the users and the purpose of the model should be taken into account when selecting the method of generating explanations.

2.4.3 Emerging Measures

Establishing a suitable trade-off between accuracy and interpretability is a challenging task since generally, more complex models are able to approximate more complex functions, but simplicity is a strong proxy for interpretability. Figure 2.3, adapted from [66, 5] shows a fictional plot representing this conception. It should be noticed that the ordering of classes of models concerning the spectrum of interpretability or performance is not absolute nor necessarily true for individual models. For example, a huge decision tree would not be necessarily more interpretable than a random forest model.

Inspired by this known trade-off Bersimas et al. [14] propose a general framework of interpretable paths for decomposing intrinsically interpretable models to building blocks. This decomposition allows defining a generic interpretability loss that makes one path more interpretable than others. In addition, using the interpretability loss, they formulate the optimization problem of computing models that are on the Pareto front of interpretability and predictive accuracy. However, this framework is only applicable when the interpretable model can be built incrementally.

There are a few attempts in the literature to use metrics for measuring specific properties of explanations. For example, in [141] the authors use local fidelity to the underlying black-box as one of the components to optimize by LIME. The work in [4] studies the robustness of explanations generated by LIME and SHAP model-agnostic

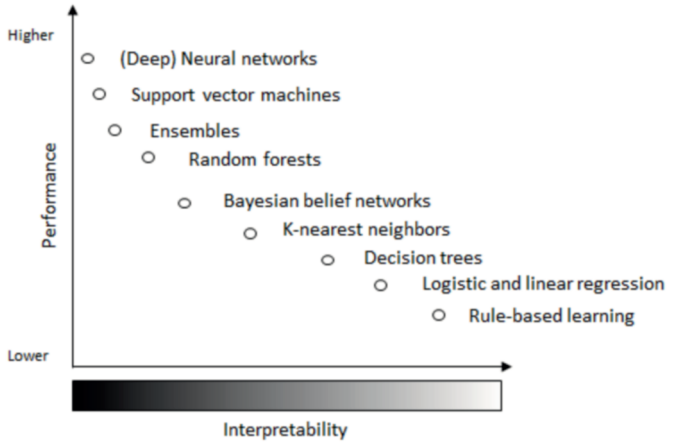


Figure 2.3: Representation of the trade-off between accuracy and interpretability for most known machine learning families of models. Adapted from figures published at [66, 5].

methods and five other techniques tailored to deep learning classification of images. The robustness notion here is related to the consistency and stability properties, since it measures how locally similar are explanations inside a model and compare them between different models for the same data points (anchor instances). The authors propose a measure based on local Lipschitz continuity, which defines a local Lipschitz estimate by optimizing:

$$\mathcal{L}_X(x_i) = \underset{x_j \in N_\epsilon(xi) \leq \epsilon}{\arg\max} \frac{\|ex(x_i) - ex(x_j)\|_2}{\|x_i - x_j\|_2} \tag{2.6}$$

where instance x_j is the ϵ-neighbor of instance x_i with the most dissimilar explanation $ex(x_j)$ to the one of x_i. The Lipschitz estimates can be then aggregated for a sample of anchor instances x_i and compare models relative to each other for the same task. However, this measure has no previously known ideal value or defined thresholds and its values strongly depend on the data domain.

Current work on explainable AI measurements has a strong dependence on the human-computer interaction field. Hoffman et al. [73] revises questionnaires and interview methods for qualitatively measuring trust, satisfaction and curiosity in the context of explainable AI. Similarly, Mohseni et al. [111] see users' mental models as a way of studying the understanding of the human about the intelligent system. These techniques measure the *a posteriori* effect of explanations on the performance of the

user in the given task. In general, more work is needed to define quantitative measures for the *a priori* quality of the explanation coming from the method.

2.5 Concluding Note

Explainable artificial intelligence is a rapidly growing topic in the field[3], although the need for interpretability in machine learning has been always present. This chapter covered different methods for obtaining interpretability and generating explanations for data-driven approaches. We made a clear distinction between interpretability and generating explanations.

On the one hand, intrinsically interpretable methods rely on their levels of transparency for generating explanations by themselves but are prone to be less accurate than black-box models. In this group, linear regression is perhaps the simplest interpretable classifier allowing explanations involving the role of features, but it is limited to approximating linear functions. Logistic regression is more flexible but its interpretation of coefficients is more obscure. Decision trees and decision lists offer a powerful prediction method that is entirely transparent, globally and for specific instances. Decision trees are more prone to do overfitting especially if no pruning is used, but decision lists provide more compact sets of rules. Naive Bayes is also a powerful transparent classifier, but its interpretation is subject to the understanding of the user of conditional probabilities. Finally, k-nearest neighbors resemble the human reasoning based on past events, although explanations are built at the level of instances instead of based on the role of features. Intrinsically interpretable models can be recommended when a transparent model that can be inspected as a whole is needed and the prediction problem does not require a very powerful technique. Decision trees and decision lists seem, in general, the most accurate predictors that can provide intrinsic interpretability when the structure is kept on a simulatable size.

On the other hand, model-agnostic post-hoc explanation methods compute explanations from black-box models to preserve accuracy. For example, partial dependency plots can provide a quick but limited view on how a given feature influences the prediction, assuming independence from other features. The feature importance measure does take into account the relationship with other features but tends to split the importance of two correlated features. Shapley values correct this problem and distribute the feature contribution fairly and efficiently. In particular, SHAP implementations for particular black-box models seem to be very useful for explaining the prediction of a given instance or obtaining a feature contribution measure for the entire dataset. Local surrogate models approximate a prediction with local interpretable models such

[3]See Figure 1 in [5] for publication statistics on December 10th, 2019

as linear regression, which leverages the intrinsic interpretability, but is limited to local explanations.

To sum up, model-agnostic post-hoc methods generate explanations that are often local or limited to feature attribution rather than a holistic view of the model. Explanations provided by intrinsically interpretable models are more derived from their structure and easily mappable to the problem domain. Global surrogates or grey-box models take the best of both worlds while trying to find a suitable trade-off between accuracy and interpretability. Being a black-box or a white-box model can be seen as a spectrum, even between families of models. The next chapter studies the interpretability potential of the state-of-the-art for semi-supervised classification based on the terminology defined in this chapter.

3 | Semi-Supervised Classification

This chapter formalizes the semi-supervised learning problem with an emphasis on pattern classification. The state-of-the-art for semi-supervised classification (SSC) techniques is explored by grouping the methods with similar assumptions in families. In addition, this chapter outlines the advantages and drawbacks of SSC methods, with a special focus on their interpretability potential.

3.1 Semi-supervised Classification Problem

In several domains, gathering data examples for training a classifier is often simple, but the process of assigning them labels can be costly in terms of money, time or effort. For example, the scenario where labeled data is scarce is very common in medical applications such as computer-aided diagnosis or prognosis analysis. This is especially true in the context of rare diseases, where labeled data is scarce but unlabeled data coming from other patients (not related to that disease) could help with the classification task at hand. In this case, it may be difficult to build a reliable supervised classifier based only on the labeled data, but an SSC method may be useful.

In supervised learning, the goal is to learn a function that maps inputs (described as vectors) into outputs based on examples of input-output observations. These observations (hereinafter called instances) are often described by a set of numerical and/or nominal attributes. More specifically, solving a supervised machine learning task im-

plies to learn a mapping $f : X \to Y$ that assigns a label $y \in Y$ to each instance $x \in X$, described by an attribute set $A = \{a_1 \ldots, a_p\}$. The mapping is learned from data in a supervised manner, i.e., by relying on a set of examples that are already labeled. Once the learning process is done, the obtained model can be used to infer the label of an unseen instance with a given certainty. On the other hand, in unsupervised learning, the task is to learn the underlying characteristics of the instances that allow us to group them, without information about a target output.

In general, semi-supervised learning techniques attempt to use both labeled and unlabeled instances during the learning process to improve on the performance of supervised methods that use only labeled data or unsupervised ones that use only unlabeled data. In the case of SSC, unlabeled points could help better elucidate the classification boundaries. While for semi-supervised clustering, information about the labels could help group the instances in a better way. The scope of this research focuses on SSC, but other related machine learning tasks are outlined in Section 3.4.

More formally, in a SSC scenario there exists a set of m instances $L = \{l_1, \ldots, l_m\}$ which are associated with their respective class labels in Y, and a set of additional n unlabeled instances $U = \{u_1, \ldots, u_n\}$, where usually $n > m$. Following above notation, L and U are disjoint and $L, U \subset X$. SSC models can operate in two different settings. In *transductive learning*, the classifier only attempts to predict the labels for the given unlabeled instances in U, but not for unseen data. In *inductive learning* on the other hand, the classifier tries to infer a mapping $g : X \to Y$, with $L, U \subset X$, in order to predict the class label of any instance associated with the classification problem [188].

SSC relies on assumptions about the data for leveraging the extra unlabeled data when compared to supervised classification baselines. The following section describes these assumptions and reviews the main families of SSC methods analyzing their interpretability potential.

3.2 State-of-the-art Review

Several families of methods have been proposed in the last decades, from the early graph-based methods to the current use of deep learning techniques. The interest on the application of semi-supervised approaches for unstructured data domains such as image classification [119, 180], text classification [148], sentiment analysis [127, 173] and video object segmentation [138, 176] has been on the rise since the success of deep learning techniques often relies on the availability of a large number of labeled data. However, semi-supervised learning continues to prove its relevance in other machine learning tasks with structured (tabular) data. Particularly, in the context of bioinformatics and medical informatics, semi-supervised techniques have proved in several tasks such as predicting disease outcome from clinical data [24, 79], disease

co-occurrence prediction [85], predicting active enhancers from gene expression data [109], protein interaction sites predictions [168] or protein topology prediction [155].

This section first describes the assumptions about the data that need to be made when using SSC methods. Different assumptions led to the emergence of different families of methods. Next, the section provides a critical review of the most representative techniques in SSC, starting from graph-based methods, through semi-supervised support vector machines, generative mixture models, the extension of the field to deep learning and finalizes with self-labeling. However, it is not the intention to provide the reader with an exhaustive review of all available methods. For an up-to-date review on general semi-supervised learning the reader can refer to [159] and [188]. Additionally, in [154] the authors provide a survey specific for graph-based methods, in [158] the authors widely cover self-labeling techniques while in [36] the authors make an overview of semi-supervised support vector machines. Lastly, this section adds our vision on the potential interpretability of these families of methods, which is a topic frequently neglected in semi-supervised learning review papers.

3.2.1 Semi-supervised Classification Assumptions

It is not evident how the semi-supervised prediction model $g : X \rightarrow Y$ can be improved using unlabeled instances in X since g represents the mapping between X and Y and the unlabeled instances $U \subset X$ do not have information about this mapping. The key lies in the assumptions that need to be made regarding the relationship between the distribution of the unlabeled data and the label. In general, the main assumption of SSC is that the underlying marginal distribution $p(x)$ provides information on the conditional distribution $p(y|x)$, from where the labeled instances were sampled. When this condition is met, it is possible to use the unlabeled data for gaining information about $p(x)$ and hence $p(y|x)$ [188]. However, the different interactions that $p(x)$ and $p(y|x)$ have in real-world problems lead to specific assumptions that need to be made when working with different methods.

A first assumption, also commonly found in supervised classification, is the *smoothness assumption*, i.e. for two instances $x_i, x_j \in X$ which are similar in the feature space, the corresponding labels y_i and y_j should be the same. This assumption is generally used with transitivity, which allows propagating the label to similar unlabeled instances in a number of steps. A complementary supposition is the *low-density assumption*, which states that the classification boundaries should be in regions of the feature space that are not crowded with observed points. When placing decision boundaries in low-density areas, at the same time the smoothness assumption is respected since populations of similar data points will be assigned to the same label. A third assumption refers to the *manifold assumption*, which states that the data points described in a high-dimensional input space come from low-dimensional sub-spaces

called manifolds. The data points that belong to the same manifold should share their label. In that way identifying the manifolds and the points belonging to them allows transferring the label from labeled instances to unlabeled ones. Finally, the *cluster assumption* states that, when data points belong to the same cluster, they also share the same label, based on a similarity measure. This assumption is often interchanged with the other assumptions since it can be seen as a generalization of them [159]. Next subsections review the main families of SSC methods while describing their connection to the SSC assumptions.

3.2.2 Graph-based Methods

Graph-based methods [16] represent the data space as a graph, where each node denotes a training instance (both labeled or unlabeled) and edges describe relations between them. The edges are weighted based on some similarity measure. Thus, they assume label *smoothness* over the graph, i.e., if two instances are strongly connected, then they likely belong to the same class. Graph-based methods also rely on the *manifold* assumption, since they potentially provide a lower-dimensional representation of the high-dimensional input data [159]. These methods were originally focusing on transductive learning, i.e. predicting the label for the given set of unlabeled data rather than finding a model capable of predicting the classification of unseen instances. Traditionally, computing the label for a new instance would require to relearn the model including the new instance as a node.

Graph-based methods estimate a continuous function closely enough to the label values, with the ultimate goal of propagating labels between similar instances. The function is usually expressed with two terms by using a loss function and a regularizer. The first term is a supervised loss function that keeps the predicted labels close to the known labeled data and the regularizer term minimizes the difference of the predicted labels to those of its neighbors. The goal of the regularizer is to keep the function smooth through the graph. The main differences among graph-based approaches are the choice of the functions for the two terms [188]. Reviews published in [159, 186, 188] cover several regularization techniques for graph-based methods.

Label propagation [187] and its modification label spreading [182] are perhaps the most used graph-based SSC methods. In these methods, each label is iteratively computed from the weighted average of the neighbors' labels until convergence. Label propagation uses the graph Laplacian for regularizing the smoothness of the function and a nearest neighbor kernel for spreading the labels while keeping the original labels fixed. Label spreading modification uses the normalized graph Laplacian and the original labels are allowed to change, thus it is more robust to noise. An analogy exists between supervised kNN and label propagation. While kNN predicts the new labels based on the labels of similar instances, graph-based methods extend the similarity

to unlabeled data points. Figure 3.1 shows how label spreading works in a synthetic dataset by using the unlabeled instances to separate classes, compared to a baseline supervised kNN.

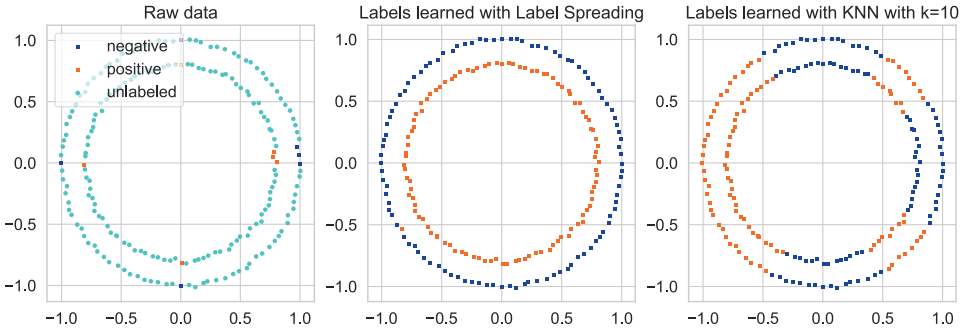


Figure 3.1: Label spreading compared to kNN method (using only labeled data and Euclidean distance), for the circles dataset. The inner circle is known to be positive and the outer circle is known to be negative. Label spreading is able to leverage the proximity of unlabeled instances and propagates the correct label. This figure is an extension of an example available in the documentation of *scikit-learn* library [129].

Recent works on label propagation methods are focused on the construction of an effective graph over data with complex distribution [44] and in the reduction of the risk of error propagation through outliers [60]. An interesting extension of label propagation to the data stream scenario can be found in [164]. Although label propagation variants are known to be computationally complex (with the worst-case scenario of $O(n^3)$) they are proven to converge to the approximate solution [12]. Graph-based methods tend to be sensitive to class imbalance [188], thus leading to multiple works using label proportions for regulating the influence of labeled instances [159].

In terms of interpretability, this family of methods tends to be more on the dark grey part of the spectrum. For the specific methods were the prediction of an unlabeled data point is equal to the average prediction of its neighbors, the algorithm transparency is similar to that of kNN (see Section 2.3.1). The parallelism with the human reasoning based on past experiences makes it look as intuitively interpretable, thus allowing the generation of example-based explanations. However, its components such as the function to be minimized, are not easily mappable to the domain. Regarding the simulatability, the graph acting as a model could be only considered transparent at the global level when the instances can be visualized in a low-dimensional space. In this setting, visually examining the structure of the obtained graph could give some holistic view of the model and the relations among data points, but it would become

too complex for even medium-size graphs. The first work toward this direction can be found in [147], where the authors propose a flow sub-graph framework which visualizes the path along the information flow from a source labeled instance to a target unlabeled instance. These sub-graphs can be seen as rather local explanations in the form of visualizations of the model. Their usability is limited to data that can be represented in the graph replacing the abstract representation of the node (e.g. images). A more general option is to obtain kNN-like explanations with examples by leveraging the graph structure, e.g. "the predicted label of instance x_i was propagated from instances x_1, x_2 and x_3".

3.2.3 Semi-supervised Support Vector Machines

An alternative SSC approach is to assume *low-density* separation of the data, i.e. the decision boundary should be in a low-density region. A well-known method in this family is transductive support vector machines (tSVM) [80, 13], which uses unlabeled data for maximizing the margin between the different classes by placing the decision boundaries in sparse regions (see Figure 3.2). As a natural extension of support vector machines (SVM), the algorithm enumerates all possible labeling of the unlabeled points, builds one standard SVM for each labeling and chooses the SVM with the largest margin. It is important to notice that, despite its name, tSVM are inductive learners since the model is built over the entire space.

Given the fact that the complexity of the optimization problem increases in the semi-supervised setting, the computational cost of this technique is very high and it does not scale well for large-scale data. Recent studies [20, 100] try to overcome this limitation by using a concave-convex procedure and variations of stochastic gradient descent to solve the optimization problem.

Another interesting approach is connected to graph-based models and their assumptions. Belkin et al. [8] proposed Laplacian support vector machines (LapSVM), which extends the regularization framework of SVM with a manifold regularization term. The manifold regularization is added as a Laplace operator, taking into account the geometry of the distribution of the unlabeled data. In contrast to some graph-based methods, LapSVM is able to deal with transductive and inductive semi-supervised learning. Compared to tSVM, LapSVM performs better in time complexity [36]. A review on modified variants of these techniques for reducing time complexity and supporting cost-sensitive classification can be found in [36].

Although SVMs are a powerful technique with a strong mathematical framework for building classifiers, it has the drawback of working as a black box from the interpretability point of view. The lack of transparency of SVMs does not allow them to produce explanations or interpretations of the obtained model. Some scattered works can be found on providing some degree of interpretability to supervised SVMs

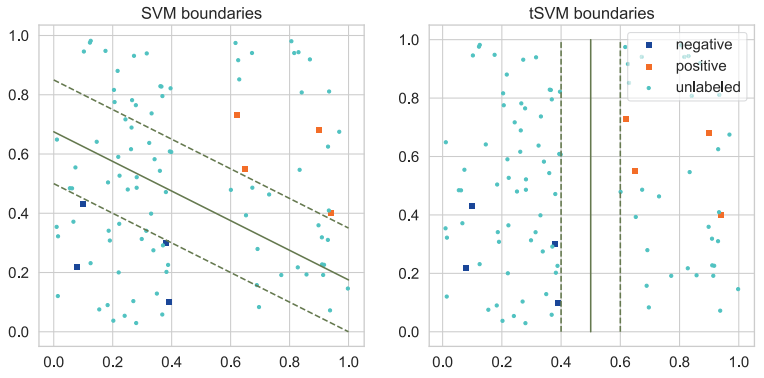


Figure 3.2: Supervised SVM vs. transductive SVM boundaries in a randomly generated binary classification problem. The solid green line represents the boundary found by each method and the dashed lines are the geometric margins of the boundaries. For SVM the boundary is guided by the labeled data only. While for tSVM, unlabeled data guide the decision boundaries to more sparse regions of the space. This figure is inspired by a similar figure in [188].

[137, 107], but no works were found for its semi-supervised flavor. In this case, the use of post-hoc methods for generating explanations is necessary when requiring explanations over the obtained predictions.

3.2.4 Generative Mixture Models

Generative models focus on learning a joint distribution $p(x, y) = p(x|y)p(y)$ from which the instances can be generated. This methods can be also used for classification by assigning an instance x_i the label y_j that maximizes the conditional probability $p(y|x)$. The conditional probability $p(y|x)$ can be computed using the Bayes rule as shown below:

$$p(y|x) = \frac{p(x|y)p(y)}{\sum_{y_j} p(x|y_j)p(y_j)} \tag{3.1}$$

where $p(x|y)$ is the class conditional probability of the instance, $p(y)$ is the prior probability of class label y and the denominator is the prior probability $p(x)$ expressed as the law of total probabilities using each class value y_j. Related to the *cluster* assumption, the idea behind generative mixture models (GMM) is to assume that the data follows a mixture of identifiable distributions (e.g. Gaussian distributions), where each distribution represents a class label [188], as depicted in Figure 3.3.

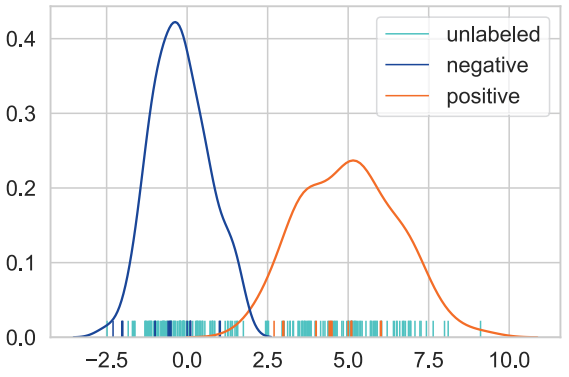


Figure 3.3: Gaussian mixture distributions of 1-dimensional instances in a randomly generated binary classification problem. Each blue curve is the distribution $p(x|y = -1)$ and the orange one is $p(x|y = +1)$. The instances are plotted as short bars in the x axis, where the unlabeled ones are represented on light blue and the labeled ones in the corresponding colors of each class label. The unlabeled instances help estimating the parameters of the distribution of each class. This figure is inspired by a similar figure in [188].

GMMs estimate the joint probability by assuming a type of distribution while learning its parameters using information from the labeled and the unlabeled data. In this way, the information from unlabeled instances can help estimate the true mean of the Gaussian distributions, for example. The expectation-maximization algorithm is used for inferring the parameters that maximize the probability of generating such training data from the model. The theoretical foundations of expectation-maximization ensure that with sufficiently large amounts of unlabeled data, a more probable model (and therefore a more accurate classifier) can be found when compared to just using the labeled data alone [81]. For example, in Figure 3.4 it can be observed how the estimated model changes by using the distribution of unlabeled data as well.

This approach may be convenient when the available data produce well-separated clusters, but in real-world applications, the joint distribution is not easily identifiable [186]. The unlabeled data could actually harm if the distribution assumption is wrong, therefore this technique should be used when there is evidence from the knowledge domain that supports the chosen distribution.

From the interpretability point of view, the classification of a new instance can leverage the Bayes rule for building a (rather abstract) explanation: "y_j is the most probable value of y for x_i since the probability $p(x_i)$ is high when $y = y_i$". Moreover, the estimated mixture distribution could only be visualized in a low-dimensional fea-

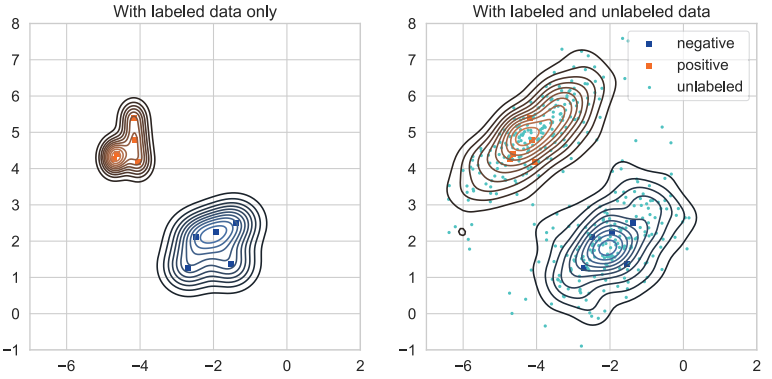


Figure 3.4: Gaussian mixture distributions obtained when (a) using only labeled data or (b) with labeled and unlabeled data combined. Each distribution from the mixture is associated with a class label (positive or negative). The data points are randomly generated in a binary classification problem for illustration purposes.

ture space for gaining insights into the clusters found by the model. In our opinion, GMMs require the use of post-hoc methods or global surrogates for gaining in interpretability of their results. An interesting work in this direction includes generating rectangular regions from the clusters and transforming them into rules [23].

3.2.5 Deep Semi-supervised Learning

Deep architectures have also been introduced in the field of semi-supervised learning by extending the generative models family. For example, generative adversarial networks (GAN) [61] are based on the idea of learning a generative model and a discriminative model at the same time. Implemented with neural networks, the generative model tries to produce data points adjusted to the real distribution of data, while the discriminator tries to predict whether a data point is real or generated. GANs are originally unsupervised but easily extensible to a supervised classification scenario by training the discriminator to distinguish different classes instead of real or fake instances. The authors in [122, 149] simultaneously proposed to use a discriminator for $c+1$ labels instead of the binary "real/fake" distinction, where the first c ones are the class labels of the problem while $c+1$ corresponds with the generated instances. The authors in [31] theoretically analyze whether a good generator and a good discriminator for semi-supervised learning can be obtained at the same time. The study concludes that the generator should be "bad" in the sense of assigning high probabilities to low-density

regions of the input space according to the true distribution, in order to complement the true data distribution and improve the semi-supervised performance.

Variational autoencoders (VAE) [35] are another line of work in extending generative models to deep learning architectures. VAEs suppose that each data point x is generated from a vector of latent variables z and tries to find a distribution $p(z)$ from which z can be easily sampled. During the training process, an encoder network learns the parameters for obtaining the distribution $p(z|x)$ based on a data point x, and the decoder learns the parameters for the correct reconstruction of x after sampling $p(z)$. The authors in [86] extend this model to the semi-supervised setting with a two-step procedure. In the first step, one VAE is trained on labeled and unlabeled data in an unsupervised way. Afterward, another VAE is trained such that it contains the label information in the latent representation. For labeled data, this extra variable represents the true labels and for unlabeled data, it is an unconstrained latent variable. Finally, a separate classification network which uses the latent features is used to infer the predictions.

For completeness, we mention some methods that use unlabeled information for adding a regularization component to the loss function in neural networks, as a natural extension of deep neural networks to semi-supervised settings. Weston et al. [169] propose to learn unsupervised encodings at multiple layers of a deep architecture jointly with a supervised task. Rasmus et al. [136] propose ladder networks as autoencoder networks with skip connections for the semi-supervised setting achieving outperforming results in the semi-supervised MNIST task [95]. This family of methods continues to outperform state-of-the-art results mainly on semi-supervised versions of image datasets such as SHVN [117] or CIFAR-10 [89], which have become the standard for the empirical evaluation of deep classifiers [123].

Regarding interpretability, deep neural networks are black-box models that need post-hoc procedures for generating explanations of their predictions. The majority of contributions are focused on local surrogate models or feature importance methods (see Subsection 2.3.2) specially designed for deep multilayer, convolutional or recurrent neural networks. For a wide review on post-hocs for generating explanations in deep learning the reader is referred to [21, 159]. Interesting works connected to the semi-supervised setting include learning disentangled latent representations in a variational autoencoder, i.e. latent variables with an interpretable meaning coming from labeled data are added to the latent representation [115]. These latent interpretable variables can be used later on for inspecting their influence in the prediction.

3.2.6 Self-labeling Techniques

While the above families of methods rely on specific assumptions about the distribution of unlabeled and labeled data, a less constrained strategy for SSC exists. Self-

labeling methods refer to a wide family of versatile techniques which are extensions of supervised ones. They are known also as wrapper methods since they encapsulate supervised classifiers which are assumed to be good predictors. This means they use one or more base classifiers for enlarging the available labeled dataset, assuming the predictions they produce on the unlabeled data are correct. Self-labeling techniques are commonly grouped in self-training and co-training methods based on whether they need one or multiple classifiers for learning.

Self-training

Self-training methods are also found in the literature referred to as self-labeling and pseudo-labeling. Self-training methods [177] rely on the prediction of one base classifier. They repeatedly increase the size of the labeled dataset by predicting unlabeled instances and retraining the base classifier. The criteria for including instances in the next enlarged labeled dataset are generally based on the confidence of the prediction.

The instances can be added incrementally, in batch [68] or in an amending procedure [98]. When adding instances incrementally, a fixed number of the most confident instances is added to the enlarged dataset. This is the simplest of the approaches but its main disadvantage is that the strictly incremental addition of instances can propagate and reinforce self-labeling errors to the next iteration. Adding the instances in batches is slightly different since, instead of a fixed number of instances, only those that meet the criteria will be added on each iteration. The amending alternative allows adding, removing or weighting the self-labeled instances to be used for enlarged labeled datasets according to a given criterion. The flexibility of an amending procedure contributes to avoiding error reinforcement.

The standard self-training scheme has been explored with a variety of base classifiers such as naive Bayes, kNN, C4.5 decision tree or support vector machines [158]. For example, the methods SETRED [98] and SNNRCE [167] use a graph cut edge statistic as a measure of confidence in the classification for the amending procedure. Logistic model trees [45] and rotation forests [46] have also been evaluated in benchmark datasets with competitive results. Other works have explored the potential of fuzzy-rough sets models in self-training, suggesting them as good base classifiers based on a preliminary study of prediction quality and stability [162]. Pseudo-label [96] has been also used with deep neural networks in combination with dropout and a denoising autoencoder, showing promising results in the semi-supervised version of MNIST dataset.

Co-training

Co-training is often used as an umbrella term that refers to self-labeling methods that use an ensemble of classifiers. However, the standard co-training approach [17] is a

multi-view method which means it needs more than one description of the dataset for its workings. The multi-view methods assume that the data space can be described from two or more different viewpoints. These different views normally correspond to distinct sets of attributes describing the same instances. In standard co-training, the base classifiers are trained separately using each attribute subset. Thereafter, the prediction of each classifier over the unlabeled dataset is used for enlarging the training set of the other. The use of this method is advisable when the features split naturally into two sets, e.g. a patient described with clinical and genomic data.

Other alternatives using multiple base classifiers which do not need multi-view datasets are democratic co-learning [183], tri-training [184], co-training by committee [67] and co-forest [99]. For example, tri-training uses three base classifiers that collaborate in the learning process by labeling an unlabeled example if the other two classifiers agree. Co-training by committee explores different ensemble strategies with bagging as the best performing one. Similarly, co-forest can be seen as the implementation of co-training using random forests as the base classifier. A wide experiment conducted in [158] shows that co-training using support vector machines as a base classifier [67], tri-training using C4.5 decision tree [184], co-bagging using C4.5 [67] and democratic co-learning [183], are the best performing self-labeling classifiers for structured (tabular) datasets.

In general, self-labeling techniques are easy to implement and apply to almost all existing classifiers. This means that when a given supervised classifier is known to be good for solving a specific task, then it can be easily extended to the semi-supervised setting with the self-labeling strategy. The supervised base classifier is completely agnostic of the wrapper classifier when passing the self-labeled instances as regular labeled instances. Thus, the performance relies on the generalization ability of the base classifiers and the strategy for avoiding the propagation of misclassifications during self-labeling. Compared to other families of SSC, there is no need of assuming additional characteristics of the data, such as smoothness, clusters, manifolds, etc..

In terms of interpretability, a self-training scheme producing a simulatable model (e.g. relatively simple tree structure) as the final classifier can be considered a transparent model. However, in the same way, a self-training with a support vector machine as a base classifier would be a black box. More complex approaches such as the multi-classifier (e.g. tri-training, co-bagging, or co-forest) or multi-view ones (e.g. standard co-training) are more difficult to interpret at a global level due to the collaborative nature of the algorithms and the complexity of the resulting ensemble. Therefore, the interpretability of a self-labeling technique depends on its base classifiers and their connection.

It can be noticed that the self-labeling concept has a direct link with the global surrogate technique for providing interpretability since they both can be seen as wrapper strategies. A black-box model known to be good for a certain task can be taken as

the base classifier for the self-labeling process. This base classifier is independent of a second wrapper classifier that can be in turn a white-box model. The wrapper will act as a global surrogate trained on the self-labeled data, thus providing interpretability as long as the features of the task are interpretable. This research takes this idea as a starting point for proposing a general schema for solving SSC problems, where interpretability is a requirement.

3.3 Empirical Evaluation of Semi-supervised Classifiers

For empirically evaluating and comparing semi-supervised classifiers, the first decision should be to identify whether the scenario is transductive or inductive. For transductive settings, only the unlabeled instances should be correctly classified. For the inductive setting, the most commonly found in literature, the classifier should be able to generalize to unseen instances. When compared with other algorithms, supervised baselines should be included. Besides, when possible for the classification task, transfer learning alternatives should also be compared [123].

Regarding the datasets, common benchmarks are the UCI Machine Learning repository [41] for structured (i.e. tabular) data and CIFAR [89], MNIST [95] and SVHN [117] for unstructured image data. It is a generalized practice to partition the datasets in labeled and unlabeled instances by using different fixed ratios. The unlabeled instances are obtained from the training set by neglecting the label information when no real unlabeled data is available. Test and validation sets are kept aside similarly to supervised classification. Generally, only the ratio of labeled data is varied in benchmark experiments. However, Oliver et al. [123] argue that for a more realistic evaluation of the semi-supervised classifiers, both the labeled and unlabeled amount of data should be varied. Since it is not possible to guarantee that adding more unlabeled data will not degrade the performance, it is important to report the relative performance compared to the supervised baseline. Performance degradation is a phenomenon observed in practice, although, it is likely under-reported due to the lack of publication of negative results [188].

3.4 What Semi-supervised Classification is Not

As mentioned before, the contribution of this thesis falls within the SSC field, which gathers the majority of the contributions of semi-supervised learning. However, there exist other machine learning tasks within the umbrella term of *learning from weak*

labels which should not be confused with SSC. A compilation of these fields is outlined below:

- *Semi-supervised clustering*: the goal of these methods is to improve the discovering of the clusters using the labeled data for identifying pairwise constraints. These constraints establish whether the cluster labels of two samples must be in the same or not. A review on these techniques can be found in [6].

- *Semi-supervised regression*: similarly to SSC, this field focuses on improving the prediction performance with extra unlabeled data, but for regression problems instead. According to a recent review in [88] the most explored techniques are regression based on the co-training paradigm, kernel regression and regression via Laplacian regularization.

- *Learning with positive and unlabeled data*: in this type of problem, only positive instances have the label available. Assuming the unlabeled data has both positive and negative instances, these methods estimate the positive class conditional probability $p(x|+)$ and $p(x)$. If the probability $p(+)$ is known, they can estimate $p(x|-)$ and perform the classification via Bayes rule [33]. Other approaches heuristically estimate negative examples from the unlabeled data [97].

- *Learning from partial labels* [30] or *superset learning* [76, 103]: in this type of task, the label associated with a training instance is only characterized in terms of a subset of possible classes including the correct one. Despite the "distracting labels", the classifier needs to learn how to disambiguate data and identify the correct label.

- *Multi-instance learning*: for this kind of task, the instances are grouped into bags that are labeled as a whole. A bag is labeled "positive" when containing at least one positive instance, otherwise will be labeled "negative". Zhou and Xu [185] show how this problem can be reformulated as a special case of SSC and solved using an SVM-based classifier.

- *Active learning* [50]: similarly to SSC, this field also tackles the context where labeled data are difficult to obtain. The difference is that it relies on an oracle (human or another system) to label selected instances of unlabeled data. The wide variety of approaches differ in the strategy for selecting the query examples: maximum entropy, least confidence, the most disagreed by an ensemble of classifiers, among others. A review on this topic can be found in [150].

3.5 Concluding Note

This chapter elaborated on the state-of-the-art on SSC techniques, with an emphasis on the assumptions they make and their potential for interpretability. SSC techniques such as graph-based methods, transductive support vector machines, or generative mixture models rely on assumptions about the unlabeled data and include them directly in their objective functions or the learning algorithms. Self-labeling methods, on the other hand, are more flexible and applicable to any supervised base classifier, thus making it easier the extension from the supervised scenario to the semi-supervised one.

We discussed how to correctly evaluate an SSC technique, recommending to compare to baseline supervised alternatives and to vary the amount of labeled and unlabeled data. In general, SSC should not only be seen as a direct way of increasing performance by adding unlabeled data. It is rather a suitable alternative for the scenario where unlabeled data is available and can help obtain a classifier that adjusts better to the problem domain. This is especially important when interpretability is a requirement since the generated explanations will be better adjusted to the domain as well.

Through the chapter, we have argued that graph-based methods, transductive support vector machines, generative models, or their deep learning extensions do not provide clear room for incorporating interpretability in their design. It seems that, for the majority of them, the use of agnostic post-hoc explanation methods is necessary. In contrast, self-labeling methods exhibit a clear connection in structure with the global surrogate strategy for providing interpretability. Both techniques are wrapper strategies that encapsulate a base classifier for optimizing different objectives. The self-labeling attempts to improve the accuracy in a semi-supervised setting concerning its supervised baseline. The global surrogate trains a white-box model on mimicking the predictions of a base black box for improving interpretability. Combining an accurate black box that performs the self-labeling with a surrogate white box that learns from the self-labeled data seems to be a promising direction. In the next chapter, we start from this idea to develop an interpretable semi-supervised classifier that aims to achieve a good trade-off between accuracy and interpretability.

4 | Self-labeling Grey-box

In this chapter, we develop the main contribution of this research, namely an interpretable semi-supervised classifier called *self-labeling grey-box* (SlGb). We combine the self-labeling approach for semi-supervised learning with the global surrogate strategy from the interpretability field. The aim is to find a good trade-off between accuracy and interpretability in the semi-supervised setting. To mitigate the propagation of mistakes in the self-labeling process, we propose two amending procedures. The first one is based on the class membership probabilities estimated by the base classifier doing the self-labeling. However, this strategy considers the labeled data as absolutely correct while inconsistency in the class labels can be present due to the limited or diverse sources of labeled data. Intending to tackle inconsistency using both labeled and unlabeled data, we propose a second amending procedure based on rough sets theory (RST). Additionally, through the chapter, we discuss the requirements of the base classifiers that are part of the SlGb.

4.1 Self-labeling Grey-box Scheme

As mentioned in Chapter 2, we refer to a *grey box* as the combination of a black-box model with a white-box one. Black boxes are normally more accurate techniques that learn exclusively from data but are not interpretable at a global level. Some classic examples of black-box models are (deep) neural networks, support vector machines or ensemble classifiers. On the other hand, white boxes refer to models that are built based on laws or principles of the problem domain. More frequently, they also refer to those who are built from data but their structure allows for explanations or

interpretation since pure white boxes rarely exist [116]. These are known in machine learning as intrinsically interpretable methods (e.g. decision trees or lists) which were covered in Section 2.3.1.

We propose SlGb as a self-labeling ensemble of two classifiers which combines a base black-box predictor with a surrogate white box. The base black-box component is the base supervised classifier in the self-labeling strategy (see Section 3.2.6). The black box performs the self-labeling of unlabeled instances based on its training on the available labeled data. The self-labeling assumes that the predictions are correct to some extent. Then, instead of re-training the same black box with the enlarged dataset, as it usually happens in self-labeling, a second white-box classifier is trained. The wrapper white-box classifier acts as a global surrogate which tries to mimic the predictions of the black box while keeping the inherent interpretability. The surrogate white box can be later used for predicting the class of unseen instances and explain (locally or globally) the predictions. Regarding performance, SlGb aims to achieve a suitable trade-off between accuracy and interpretability by outperforming the white-box base classifier while keeping a similar complexity in structure.

4.2 Architecture and Learning Algorithm

The learning process of SlGb is performed in sequential order. In a first step (self-labeling), we provide the available labeled dataset (L, Y) to a black-box classifier for training. The black box estimates a function $f : L \rightarrow Y$, where $f \in F$, being F the hypothesis space that associates each instance with a class label. The function f can be computed from the scoring function $h : L \times Y \rightarrow [0, 1]$ such that $f(l) = argmax_{y \in Y}\{h(l, y)\}, l \in L$. For example, the function h can be the class membership probability, whose usefulness in this context will be explained in the coming sections. Thereafter, the trained black-box component is used for generating new tuples (u, y) by mapping all or a subset of unlabeled instances $u \in U$ to a class label $y \in Y$. This mapping is possible with the function f as $y = f(u)$, thus adding a self-labeling character to the model. From this step, we obtain an enlarged training set $(L \cup U, Y)$ comprising the originally labeled instances and the extra labeled ones. The process of enlarging the labeled dataset can be performed iteratively, in batch or using an amending strategy (see Section 3.2.6). In our proposal, we add all instances at once with different amending procedures that will be detailed later in Sections 4.3.1 and 4.3.2.

In the second step, we want to approximate the function f with a function g subject to the restriction that g is intrinsically interpretable. The enlarged training set $(L \cup U, Y)$ is used to learn the surrogate white-box classifier $g : (L \cup U) \rightarrow Y$ with $L \cup U \subset X$, with X being the set of instances (see Section 3.1). This results

in a classifier which is more likely to have better generalization capabilities than the supervised white-box base component. Figure 4.1 summarizes the learning process.

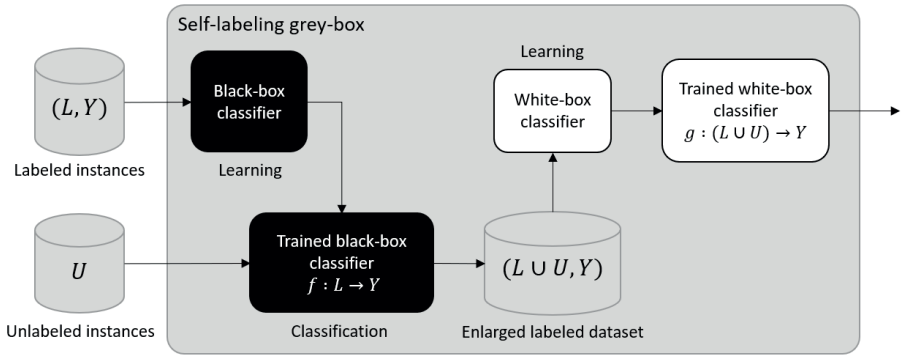


Figure 4.1: Blueprint of the SlGb architecture. During the first step, labeled data is used for training a black-box model, which assigns labels to the unlabeled data. Later on, a white-box surrogate model is trained on the enlarged dataset, thus resulting in an interpretable model.

When applying self-labeling, we should be aware of the risk of having imbalanced data concerning the class labels. It might be easier to obtain unlabeled data of a certain class, for example, in the context of rare diseases classification. To overcome this problem, our approach additionally incorporates a simple strategy for balancing instances as a preprocessing step. This weight is computed as:

$$w_{(l_j,y_i)} = |L_{[y_{min}]}|/|L_{[y_i]}| \tag{4.1}$$

where $L_{[y_i]}, L_{[y_{min}]} \subset L$ denote the sets of labeled instances that are mapped to the class label y_i and the minority class y_{min}, respectively. This preprocessing step assigns a weight that is proportional to the size of the instances' class. The minority class receives the highest weight. It must be noticed that the balancing is performed on the labeled data only, before training the black-box component. The aim is that the black box learns, in the same proportion, from labels with different availability.

Being a self-labeling method, the SlGb classifier is only based on the general assumption made by SSC: the distribution of unlabeled instances helps elucidate the distribution of all examples. According to the taxonomy proposed in [158], our model can be categorized as follows:

single-view: the SlGb classifier does not need different attribute sets for describing the instances, thus adding simplicity to the model;

multi-classifier: two different base classifiers are used, connected in a sequential process;

multi-learning: the learning process comprises two steps, where two different learning algorithms are used, depending on the base classifiers.

It can be noticed that the performance of the SlGb classifier depends on two factors. Firstly, the generalization capability of the underlying black box that performs the self-labeling part. Errors in the classification of unlabeled instances can reinforce themselves when the self-labeling is made in an iterative incremental way. Secondly, the ability of the surrogate white box for approximating the predictions of the black box as accurately as possible while keeping interpretability. For the second factor, we will propose measures for empirically testing the behavior of the white box in Chapter 5. In the next section, we address the first factor by proposing two amending strategies for weighting instances that are suspected to be misclassified, to prevent the errors from propagating through the model.

4.3 Amending Strategies

Since self-labeling assumes that its classifications are correct, it comes with the potential drawback of propagating errors in the enlarged dataset. Amending procedures are used for removing, adding or weighting instances of the enlarged dataset in the self-labeling process to avoid this problem. This process is guided by a heuristic that tries to identify the possible misclassifications of unlabeled instances. In this section, we describe two strategies for weighting the instances of the enlarged dataset. The goal is to improve the performance of the SlGb model either in terms of accuracy or interpretability. The first strategy uses the class membership probability estimated by the baseline black box and the second one focuses on detecting class inconsistency of the enlarged dataset. Therefore, both procedures assign more importance to more reliable instances in the second learning step (i.e. the training of the white box), thus avoiding the propagation of errors or superfluous information.

4.3.1 Amending based on Class Membership Probabilities

In the first strategy, the amending process is based on the class membership probability computed by the black box. The weights are assigned to the originally unlabeled instances after being labeled by the black box. They express a confidence degree associated with the label produced by the black box. By assigning the weights to self-labeled instances only, we assume that the ground truth labels are correct. The aim is to induce the surrogate model to learn from the most confident instances after the

self-labeling. Equation 4.2 shows how to compute the weight $w_{(u_k,y_i)}$ for the unlabeled instances. This equation uses the scoring function of the black-box model $h(u_k, y_i)$ that expresses the probability of u_k being correctly assigned to the y_i class,

$$w_{(u_k,y_i)} = h(u_k, y_i). \tag{4.2}$$

It is important to mention that the black-box classifier should be able to measure calibrated probabilities to correctly interpret them as the confidence of its predictions. By calibrated probabilities, we mean that the probabilities predicted by the model match with the expected distribution of probabilities for each class. For example, a binary prediction model is well-calibrated when, from those instances to which the model assigns a probability near to 0.9, the 90% is actually from the positive class. Not all machine learning models are able to provide well-calibrated probabilities. According to a study on probability estimation for different supervised classifiers [118], maximum margin methods such as boosted trees and support vector machines produce distorted probabilities. Other methods such as logistic regression, multilayer perceptrons, and bagged trees naturally provide well-calibrated probabilities.

Interpreting probabilities from models without calibration could constitute a source of bias. An interesting example published in [181] shows that datasets for multi-label object classification contains significant gender bias, which is further amplified when learning a conditional random fields model [91]. In one of the analyzed datasets, 67% of *cooking* images are associated with *women* and the rest with *men*, while after learning, the model labels *women* in 84% of *cooking* images.

When the calibration of probabilities is needed, two main options are available: Platt's scaling [132] and isotonic regression [178]. Platt's scaling calibrates the probabilities by fitting a logistic regression to the predicted scores. The logistic regression predicts the calibrated probabilities from the output of the non-calibrated model. This alternative is recommended especially for SVM and other techniques that describe a sigmoid shape in the distortion of the probabilities. Alternatively, isotonic regression is recommended for boosted naive Bayes, SVM or decision trees. This method fits an isotonic (i.e. monotonic) free-form line that adjusts better to the data points. However, it requires a significant amount of data for avoiding overfitting. The choice of calibration algorithm therefore heavily depends on the base classifier and the size of the dataset available.

The proposed amending strategy constitutes an alternative to the use of incremental or batch procedures. Our amending does not need several iterations, thus reducing the computational burden of the self-labeling process. The pseudo-code in Algorithm 1 formalizes the method and incorporates the amending step in the general scheme.

The amending based on class membership probabilities assumes the ground truth labels are correct and induces the white box to focus its learning on instances that are certain according to that. However, when dealing with limited labeled data we

Data: Labeled instances (L, Y), Unlabeled instances U
Result: $g : (L \cup U) \rightarrow Y$
begin
  /* Preprocessing: Weight the labeled instances according to Eq. 4.1 */
  forall $(l_j, y_i) \in (L, Y)$ **do**
    $w_{(l_j,y_i)} \longleftarrow |L_{min}|/|L_i|$
  end
  /* Train the black-box component with the weighted labeled data. Include a calibration procedure if needed for the black box. */
  $f, h \longleftarrow blackboxClassifier.fit(L, Y, w)$
  /* Self-labeling process: Assign a label to the unlabeled instances using the black-box inference */
  forall $u_k \in U$ **do**
    $y_i \longleftarrow f(u_k)$
    /* Compute the weight of the unlabeled instance according to Eq. 4.2 */
    $w_{(u_k,y_i)} \longleftarrow h(u_k, y_i)$
    /* Add the instance to the enlarged dataset */
    $(L \cup U, Y) \cup \{(u_k, y_i)\}$
  end
  /* Train the white-box component with the weighted $(L \cup U, Y)$ dataset */
  $g \longleftarrow whiteboxClassifier.fit(L \cup U, Y, w)$
  return g
end

Algorithm 1: SlGb learning algorithm with confidence amending. The learning is performed in two steps and the self-labeled instances are weighted by the confidence-based amending procedure.

should not discard the existence of noise in the class labels. This can generate class inconsistency, especially when unlabeled data is added from different sources.

4.3.2 Amending based on Inclusion Degree from Rough Sets Theory

We refer to a dataset as inconsistent, when identical or very similar instances have different labels. Inconsistency can arise during the self-labeling process when adding unlabeled data. This can happen because the black box predicted wrong labels for unlabeled instances or because there is noise in the class labels of the originally labeled

data. To address both issues, we propose another amending method which will be applied to the entire enlarged dataset, instead of only the self-labeled instances.

This amending is based on the principles of RST [128]. This formalism allows handling inconsistency through the computation of the lower and upper approximations for any set in the decision space. Next, we describe how the rough regions associated with these approximations can be used for weighting the instances after performing the self-labeling process.

Rough Sets Theory

RST [128] is a mathematical formalism for handling uncertainty in the form of inconsistency. This theory is part of the *granular computing* paradigm [130], which aims to recognize and exploit the knowledge present in data at various scales or levels of resolution. An information granule can be defined as a collection of objects sharing a specific property. Therefore, RST attempts to build information granules by assuming that every pair of instances in a universe of data that have the same (or similar) description are inseparable, according to a (sub)set of attributes.

Let $DS = (\mathcal{U}, A \cup \{d\})$ be a decision system where the universe of objects $\mathcal{U}$ is described by a non-empty finite set of attributes A and its respective decision class d, any concept (subset of objects) $X \in \mathcal{U}$ can be approximated by two crisp sets. These sets are called lower and upper approximations of X ($\underline{B}X$ and $\overline{B}X$, respectively) and can be computed taking into account an equivalence relation, as follows:

$$\underline{B}X = \{x \in \mathcal{U} \mid [x]_B \subseteq X\} \tag{4.3}$$

$$\overline{B}X = \{x \in \mathcal{U} \mid [x]_B \cap X \neq \emptyset\} \tag{4.4}$$

The equivalence class $[x]_B$ gathers the objects in the universe $\mathcal{U}$ which are inseparable according to a subset of attributes $B \subseteq A$. From the formulations of upper and lower approximation, we can derive the positive, negative and boundary regions of any subset $X \in \mathcal{U}$. The *positive region* $\mathcal{P}(X) = \underline{B}X$ includes those objects that are surely contained in X; the *negative region* $\mathcal{N}(X) = \mathcal{U} - \overline{B}X$ denotes those objects that are surely not contained in X, while the *boundary region* $\mathcal{B}(X) = \overline{B}X - \underline{B}X$ captures the objects whose membership to the set X is uncertain, i.e., they might or might be not members of X.

The classic RST is regularly defined over a subset of discrete attributes, which produces a partition of $\mathcal{U}$. A more relaxed formulation of RST establishes the inseparability between objects based on a weak binary relation. Equation 4.5 formalizes the

similarity relation used in this research, which defines whether two objects x_i and x_j can be considered similar or not,

$$\mathcal{R} : x_i \mathcal{R} x_j \rightarrow \psi(x_i, x_j) \geq \varepsilon \tag{4.5}$$

where $0 < \psi(x_i, x_j) < 1$ computes the extent to which x_i and x_j are deemed inseparable as indicated by the similarity threshold ε. Under this assumption, the universe is arranged in similarity classes that are no longer disjoint but overlapping. Through this work, $\varepsilon = 0.98$ and the similarity function $\psi(x_i, x_j) = 1 - \delta(x_i, x_j)$, i.e. is defined as the complement of a distance function, such as the *Heterogeneous Euclidean-Overlap Metric* [171]. This distance function computes the normalized Euclidean distance between numerical attributes and an overlap metric for nominal attributes. Equations 4.6 and 4.7 define the distance function,

$$\delta(x_i, x_j) = \sqrt{\frac{\sum_{t=1}^{|B|} \omega_t \rho_t(x_i, x_j)}{\sum_{t=1}^{|B|} \omega_t}} \tag{4.6}$$

with,

$$\rho_t(x_i, x_j) = \begin{cases} 0 & \text{if } b_t \text{ is nominal } \wedge\, x_i(b_t) = x_j(b_t) \\ 1 & \text{if } b_t \text{ is nominal } \wedge\, x_i(b_t) \neq x_j(b_t) \\ (x_i(b_t) - x_j(b_t))^2 & \text{if } b_t \text{ is numerical} \end{cases} \tag{4.7}$$

where $x_i(b_t)$ and $x_j(b_t)$ denote the normalized values of the t-th attribute $b \in B$ for heterogeneous instances x_i and x_j, respectively, and ω_t is the information gain[172] of the b_t attribute.

Once the covering of the decision space is generated according to the similarity function, several RST-based measures can be computed for quantifying the uncertainty contained in a dataset [9]. In the following subsection, we adopt one of these measures to weight the instances belonging to the enlarged training set obtained after performing the self-labeling process.

Inclusion Degree

The second amending strategy is based on the *inclusion degree* of both labeled and self-labeled instances into the RST regions. Thus, let $X = L \cup U$ represent all instances in the enlarged dataset and $d = y$, i.e. the decisions of DS be the class labels of the semi-supervised problem. Each concept $X_{[y_i]}$ to be approximated with RST represents the subset of instances in X that have class y_i. Each information granule, i.e. the positive $\mathcal{P}(X_{[y_i]})$, negative $\mathcal{N}(X_{[y_i]})$ and boundary $\mathcal{B}(X_{[y_i]})$ regions for each decision class are

computed from the enlarged dataset containing labeled and self-labeled instances. Let $\mu^{\mathcal{R}}_{\mathcal{P}_{[y_i]}}(x)$, $\mu^{\mathcal{R}}_{\mathcal{B}_{(y_i)}}(x)$ and $\mu^{\mathcal{R}}_{\mathcal{N}_{(y_i)}}(x)$ denote the membership degrees of any instance x to the positive, boundary and negative region of its decision class y_i, respectively. These membership degrees are computed from the inclusion degree [175] of the similarity class of x into each information granule,

$$\mu^{\mathcal{R}}_{\mathcal{P}_{(y_i)}}(x) = \frac{|\bar{\mathcal{R}}(x) \cap \mathcal{P}(X_{[y_i]})|}{|\mathcal{P}(X_{[y_i]})|} \tag{4.8}$$

$$\mu^{\mathcal{R}}_{\mathcal{B}(y_i)}(x) = \frac{|\bar{\mathcal{R}}(x) \cap \mathcal{B}(X_{[y_i]})|}{|\mathcal{B}(X_{[y_i]})|} \tag{4.9}$$

$$\mu^{\mathcal{R}}_{\mathcal{N}(y_i)}(x) = \frac{|\bar{\mathcal{R}}(x) \cap \mathcal{N}(X_{[y_i]})|}{|\mathcal{N}(X_{[y_i]})|} \tag{4.10}$$

where $\bar{\mathcal{R}}(x)$ is the similarity class associated with the instance x. The similarity class of an instance x groups all instances that are similar to x according to the subset of attributes taken into account. By computing how much x and its similar instances are included in the positive region of its class y_i, we are estimating how sure we are of this classification. On the other hand, a high inclusion degree of $\bar{\mathcal{R}}(x)$ on a negative region of a class y_i hints a misclassification by the black-box (when x is from the self-labeled subset) or a class noise (when x is from the labeled instances). A high membership to the boundary region can be considered positive evidence about the class label to some extent, but less certain than the positive evidence.

Equation 4.11 computes the weight for the instance x belonging to the enlarged dataset, given its label y_i and a similarity relation $\mathcal{R}$. The sigmoid function $\phi(.)$ is used to maintain the weight in the $(0, 1)$ range.

$$w_{(x,y_i)} = \phi\left(\mu^{\mathcal{R}}_{\mathcal{P}(y_i)}(x) + 0.5 * \mu^{\mathcal{R}}_{\mathcal{B}(y_i)}(x) - \mu^{\mathcal{R}}_{\mathcal{N}(y_i)}(x)\right) \tag{4.11}$$

with

$$\phi(x) = \frac{1}{1 + e^{-x}} \tag{4.12}$$

The intuition of this weight is aggregating all evidence from positive, negative and even boundary regions. A high weight near one would indicate high confidence, while a low weight near zero would identify the less reliable instances. Observe that the boundary information is also interesting since a high inclusion degree of an instance in the boundary region is, to some extent, positive evidence as well (see Equation 4.4). Therefore, we assign an importance of 0.5 to the evidence coming from the boundary region. The boundary region role can be reinforced or diluted according to

the evidence coming from the inclusion degrees in the other two regions. Ignoring the boundary information by setting its importance to 0.0 decreases the weight towards zero when the evidence from the positive region is not strong enough to counterfeit the negative evidence. Considering that negative evidence tends to be strong, i.e. the negative regions tend to be bigger, this will trigger that the training of the white box heavily relies on very confident instances only. In contrast, when the positive evidence is very weak and the boundary is taken into account with a positive weight, there is still a chance for the instance to be assigned a weight larger than zero.

When using the RST-based amending, Equation 4.11 replaces Equation 4.2 in the pseudo-code but the weighting is performed on the entire enlarged dataset. Algorithm 2 reflects this difference.

Finally, Figure 4.2 illustrates the inclusion of the amending procedures into the learning scheme of the SlGb algorithm. When using RST-based amending, the enlarged dataset is modified taking into account the class noise in the originally labeled data as well.

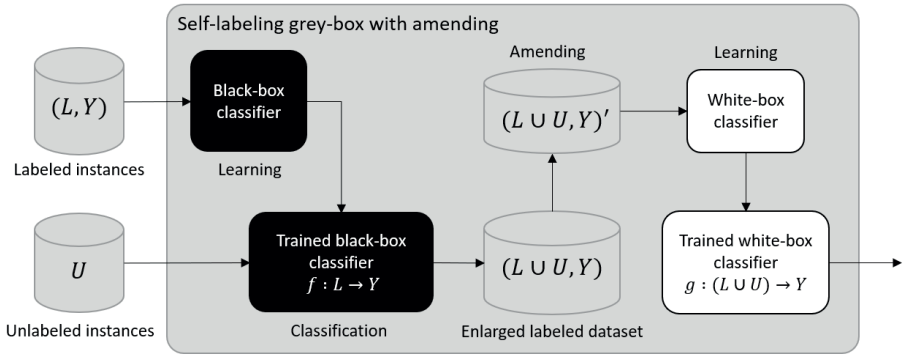


Figure 4.2: Blueprint of the SlGb architecture using amending procedures for correcting the influence of the misclassifications from the self-labeling process. When RST-based amending is used, it also tackles class inconsistency coming from noise in the labeled data.

4.4 Other Amending Alternatives

An alternative way of applying amending when using class membership probabilities could be resetting the imbalance weight of labeled instances. This value could be reset to 1.0, assuming that ground truth labels are 100% confident, or to its class

Data: Labeled instances (L, Y), Unlabeled instances U
Result: $g : (L \cup U) \rightarrow Y$
begin
  /* Preprocessing: Weight the labeled instances according to Eq. 4.1 */
  forall $(l_j, y_i) \in (L, Y)$ **do**
    $w_{(l_j,y_i)} \longleftarrow |L_{min}|/|L_i|$
  end
  /* Train the black-box component with weighted labeled data. Include a calibration procedure if needed for the black box. */
  $f, h \longleftarrow blackboxClassifier.fit(L, Y, w)$
  /* Self-labeling process: Assign a label to unlabeled instances using the black-box inference */
  forall $u_k \in U$ **do**
    $y_i \longleftarrow f(u_k)$
    /* Add the instance to the enlarged dataset */
    $(L \cup U, Y) \cup \{(u_k, y_i)\}$
  end
  forall $x \in L \cup U$ **do**
    /* Compute the weights of the enlarged dataset according to Eq. 4.11 */
    $w_{(x,y_i)} = \phi\left(\mu^{\mathcal{R}}_{\mathcal{P}(y_i)}(x) + 0.5 * \mu^{\mathcal{R}}_{\mathcal{B}(y_i)}(x) - \mu^{\mathcal{R}}_{\mathcal{N}(y_i)}(x)\right)$
  end
  /* Train the white-box component with the weighted $(L \cup U, Y)$ dataset */
  $g \longleftarrow whiteboxClassifier.fit(L \cup U, Y, w)$
  return g
end

Algorithm 2: SlGb learning algorithm with RST-based amending. The learning is performed in two steps and the enlarged dataset is weighted based on the inclusion degrees.

membership probability, which is expected to be near 1.0 since it is part of the training data. This modification would imply that all labeled instances would have similar and high importance for the white box independently of their class label. This alternative would be worth exploring in a scenario where the labeled data is very trustful, the inherent imbalance in the explanations obtained is not an issue, and the interpretable model must strongly reflect this knowledge.

In the RST-based amending, by using a similarity relation instead of an equivalence relation for building the information granules, we create an overlap instead of a

partition of the decision space. However, even with this flexible approach, an instance with a noisy class can affect the positive region of a class. For example, when there is a negative instance surrounded by positive ones, all similarity classes that include the negative instance will be taken out of the class's positive region and placed in the boundary region. This situation can lead to small positive regions, and it could be corrected by tuning the similarity relation threshold to leave the negative instance out of the similarity class. A suitable alternative is to use more flexible flavors of RST, such as Fuzzy RST [29], which allows the objects to belong to a concept with different degrees. Fuzzy RST would remove the similarity threshold and allow each instance to belong to the positive, negative, and boundary regions of each class with a certain degree.

Another alternative applicable for both amending procedures is to exclude instances with low confidence (either probability or RST-based). This option would bias the white box to explain the most confident data points, gaining certainty but reducing novelty, i.e., the ability to explain more rare data points (see Section 2.4). As another option, low-confidence instances can be grouped in a new class label "unknown" and either explore an active learning approach with them or see whether the white box can identify common patterns.

4.5 Concluding Note

This chapter presented *self-labeling grey-box*, a semi-supervised classifier aiming to provide a good balance between accuracy and interpretability. The SlGb uses an accurate black-box component for labeling unlabeled data. A white-box classifier is then used as a global surrogate model for building an interpretable model. To avoid the propagation of errors in the self-labeling process, two amending procedures are proposed. Both amending strategies aim to correct the misclassifications by weighting the instances before the learning process of the white-box surrogate occurs. The first strategy is based on class membership probabilities provided by the black box in the self-labeling. The second strategy aims to also correct the inconsistency in the labels in the enlarged dataset by computing the certainty of the classification based on the RST inclusion degree. RST-based amending covers the two sources of class noise commonly found in machine learning [189]: i) inconsistency: very similar examples are labeled with different classifications, and ii) misclassifications: instances are labeled with wrong classes. By considering the class inconsistency in the ground truth labels together with the self-labeled ones, RST-based amending reduces the impact of the class noise. Furthermore, the RST-based amending could have a positive influence on reducing the number of explanations produced by the white box.

The use of amending by weighting can have implications for the interpretability. Assigning high weights to a small subset of instances transforms the global surrogate model towards a more local one. In other words, the weighting of instances makes the white box biased towards learning from the most confident ones, thus providing explanations for some sub-spaces of the domain. Although, contrary to local surrogates, these explanations are based on real data points and not synthetic ones. In addition, it makes sense to provide interpretability or explanations over the predictions that are most certain in the problem domain.

The next chapters study the effect of combining different black-box and white-box base classifiers, as well as the influence of the two amending procedures in terms of accuracy and interpretability. In Chapter 5, we perform this evaluation on extensive benchmark data, while Chapters 6 and 7 focus on real application problems.

5 | Evaluation on Benchmark Datasets

In this chapter, we evaluate the predictive ability of the proposed SlGb classifier through a three-step methodology using standard benchmark datasets. We evaluate different settings of the SlGb in terms of performance and interpretability while having different percentages of labeled instances. Additionally, we propose two new evaluation measures related to interpretability and one that connects interpretability and accuracy.

The first step of our empirical study is devoted to determining which black-box classifier produces the best results in terms of prediction performance. This step is quite important since the overall performance will depend on the discriminatory ability of the black box. The second step is dedicated to determining which combination of white box and amending approach reaches the best trade-off between prediction rates and interpretability. As a third step, we further explore the impact of having different percentages of labeled and unlabeled instances on the algorithm's performance.

For completeness, in the last part of this chapter, we compare the proposed SlGb against the best-performing state-of-the-art methods for structured data. For that section, the evaluation is limited to prediction rates as the state-of-the-art methods used for comparison are black boxes and cannot be interpreted. We show that SlGb is not just simple and transparent, but also able to outperform other self-labeling methods reported in the literature for classification tasks with structured tabular datasets.

5.1 Benchmark Dataset Description

Our experimental design includes 55 challenging and diverse datasets for classification tasks where features are structured (i.e. the dataset has tabular form) and therefore are potentially interpretable. Four ratios of labeled instances in the training set (from 10% to 40%) allow studying the influence of the number of labeled examples on the overall performance (see Section 3.3). Testing with a 10% ratio means that the training set contains only a 10% of labeled instances and the rest of are unlabeled, the instances in the test set are all labeled but set apart. These datasets comprise different characteristics: the number of attributes ranges from 2 to 90, the number of decision classes from 2 to 28, and the number of instances from 100 to 19,000. Moreover, we have 25 datasets with different degrees of class imbalance and roughly half of the datasets are multiclass problems. Table A.1 in Appendix A show the detailed list of benchmark datasets that were used.

These datasets are partitioned into training and test sets as done in a 10-fold cross-validation process, but each training set consists of labeled and unlabeled instances. The subset of unlabeled instances is obtained by performing a random selection without replacement and neglecting the class label of such instances. The ratio (10% to 40%) determines the number of labeled instances that are kept in this process for each training set. These datasets (including the cross-validation fold partitions) were provided as supplementary material in [158] and constitute an standard in the evaluation of shallow SSC techniques. We use these datasets, including the partitions as a form of guaranteeing a fair comparison against state-of-the-art SSC methods (see Section 5.7).

5.2 Base Classifiers and Parameter Settings

There are several algorithms that can be adopted as base classifiers. On one hand, the selected classifier for the base black box should exhibit a strong predictive capability as it is used to determine the decision class of unlabeled instances. Next, we describe three mainstream supervised classifiers that will be used in the experiments for instantiating the black-box component. Our choice is motivated by experimental evidence of their superior performance in a wide range of classification problems [179, 47, 165] and their ability to produce calibrated probabilities (except for support vector machines where a calibration post-hoc is needed).

Black-box classifiers

- Random Forests (RF) [18]: Ensemble of decision trees that uses bagging technique for aggregating the results in order to reduce the high variance of indi-

vidual decision trees. Individual decision trees are built with a random subset of attributes and a random sample with replacement of instances. In our implementation 100 trees are aggregated and the number of random attributes to consider for each tree equals $\log_2(|A|)$.

- Multilayer Perceptron (MLP) [72]: Feed-forward neural network using backpropagation algorithm for adjusting its weights. Our implementation uses learning rate equals to 0.3, momentum equals to 0.2, 500 epochs for learning and one hidden layer with $(|A| + |Y|)/2$ as the number of neurons.

- Support Vector Machine (SVM) [131, 83]: Support vector machine classifier using sequential minimal optimization algorithm for training. Our implementation uses a polynomial kernel with Platt's scaling (logistic) calibration of probabilities (see Section 4.3.1).

On the other hand, for the white-box component any intrinsically interpretable classifier can be used as a surrogate model (see Section 2.3.1). Therefore, the choice of a white box must be driven by the type of explanations that are desired, e.g. rules, feature coefficients, probabilities, examples, etc. We decide to explore decision trees and decision lists alternatives as they provide both intuitive individual explanations in the form of *if-then* rules and a view of the model as a whole. For decision trees, the hierarchical structure provides this view and it can be considered transparent as long as the size of the tree remains manageable. For the case of the decision lists, rules sets are generally more concise than the ones extracted from decision trees. Additionally, these algorithms are able to handle weighted instances in the learning process. Next, we describe three classifiers explored in the scope of this experiment.

White-box classifiers

- Decision Tree (C45) [134]: For a general description of decision trees and their interpretability see Section 2.3.1. Our implementation uses C4.5 algorithm for inducing the decision tree. We allow two instances as the minimum number of instances per leaf. The confidence factor for pruning is 0.25, where a lower value incurs in more pruning. When pruning the sub-tree raising operation is used.

- PART Decision List (PART) [49]: For a general description of decision lists and their interpretability see Section 2.3.1. PART uses the separate-and-conquer strategy for building a rule set by generating a partial C4.5 decision tree and making the most confident leaf into a rule. In the next iteration, all covered instances are removed from the dataset and the process is repeated. Thus, decision lists must be interpreted in order. Our implementation uses the same

hyper-parameters of the decision tree described above for generating the partial C4.5 decision trees.

- RIPPER Decision List (RIP) [28]: This method is a propositional rule learner with a separate-and-conquer strategy, as described for PART. Additionally, the training data is split into a growing set and a pruning set for performing reduced error pruning. The rule set formed from the growing set is simplified with pruning operations optimizing the error on the pruning set. For our implementation, the minimum allowed support of a rule is two and the data is split in three folds where one is used for pruning. Besides, two optimization iterations are performed.

For completeness, we repeat the amending procedures proposed in Chapter 4 that will be tested in combination with the previous base classifiers.

Amending procedures

- No amending (NONE): The first option is not using amending. All self-labeled instances are provided as extra data to the surrogate white box. This is used as a baseline for evaluating the contribution of the two amending procedures proposed.
- Amending based on class membership probabilities (CONF): Amending procedure based on calibrated class membership probabilities obtained from the black-box base classifier, as proposed in Section 4.3.1.
- Amending based on RST inclusion degree measure (RST): Amending procedure based on RST aiming to correct the inconsistency in the classifications, as described in Subsection 4.3.2.

Hereinafter, when referring to a particular configuration of SlGb we denote it as "*bb-wb-am*" where *bb* represents the base black box, *wb* represents the surrogate white box and *am* represents the amending procedure. The code, datasets and results using different measures (e.g., kappa, accuracy, number of rules) are available as supplementary material for reproducibility purposes[1]. All SlGb configurations were implemented using *weka* library [172] and its default parameters listed above, yet obtaining competitive results against state-of-the-art methods (see details in Section 5.7).

[1] gitlab.ai.vub.ac.be/igraugar/slgb_scripts/tree/paper

5.3 Impact of the Black-Box Base Classifiers on the Performance

This section focuses on evaluating the influence of the base black box on the performance of the algorithm. Here no amending procedure is taken into account yet since it does not directly affect the ability of the black box to produce correct classifications.

To measure the configurations in terms of prediction rates we report the Cohen's kappa coefficient [27]. This measure estimates the inter-rater agreement for categorical items and ranges in $[-1, 1]$, where -1 indicates no agreement between the prediction and the actual values, 0 means no learning (i.e., random prediction), and 1 total agreement or perfect performance. Additionally, other measures such as accuracy were also computed and can be found the in aforementioned repository for reproducibility purposes. While accuracy is considered mainstream when measuring classification rates, the kappa is a more robust measure since this coefficient takes into account the agreement occurring by chance, which is especially relevant for datasets with class imbalance [78, 10].

Table 5.1 gives the mean and the standard deviation of the kappa coefficient over the 10-fold cross-validation, achieved for each configuration of SlGb. We group the results for different percentages of labeled instances. The numerical simulations indicate that using RF as the black-box component leads to higher prediction rates. In particular, the RF-PART-NONE configuration stands as the best performing one for varying amounts of labeled instances, very closely followed by RF-C45-NONE and RF-RIP-NONE.

To provide a rigorous statistical analysis of the differences, we compute the Friedman two-way analysis of variances by ranks [52], per ratio. The test suggests rejecting the null hypotheses for all labeled ratios based on a confidence interval of 95% (see Table B.1 in Appendix B[2]). This means that there exist significant differences between at least two configurations on each ratio.

The next step is focused on determining whether RF black box is truly superior compared to other configurations. To do so, we adopt the Wilcoxon signed-rank test [170] and Holm's post-hoc procedure [75] to correct the p-values, as suggested by Benavoli *et al.* [11]. Table B.2 reports the unadjusted p-value computed by the Wilcoxon test and the corrected p-value associated with each pairwise comparison. To discover the influence of the black box we compare the pairs of configurations using the same surrogate white box. Each section of the table represents the ratio of labeled instances. The null hypothesis states that there is no significant difference between the performance of each pair of configurations. All null hypotheses are rejected, except for

[2]All tables related to statistical tests are included in Appendix B.

Table 5.1: Prediction rates (kappa) achieved by different combinations of black-box and white-box algorithms without using amending. Results are grouped by ratio and best results are highlighted in bold. Random forest as black-box component leads to higher prediction rates.

	Ratio	10%	20%	30%	40%
MLP-C45-NONE	mean	0.50	0.53	0.55	0.56
	stdev	0.28	0.28	0.28	0.28
MLP-PART-NONE	mean	0.50	0.54	0.56	0.57
	stdev	0.29	0.28	0.28	0.28
MLP-RIP-NONE	mean	0.51	0.54	0.55	0.57
	stdev	0.29	0.28	0.28	0.28
RF-C45-NONE	mean	**0.56**	**0.60**	**0.61**	0.61
	stdev	0.28	0.27	0.27	0.27
RF-PART-NONE	mean	0.56	**0.60**	**0.61**	**0.62**
	stdev	0.29	0.27	0.27	0.27
RF-RIP-NONE	mean	0.55	**0.60**	**0.61**	**0.62**
	stdev	0.28	0.27	0.27	0.27
SVM-C45-NONE	mean	0.49	0.53	0.55	0.56
	stdev	0.28	0.27	0.27	0.26
SVM-PART-NONE	mean	0.50	0.53	0.56	0.57
	stdev	0.28	0.28	0.28	0.27
SVM-RIP-NONE	mean	0.50	0.53	0.55	0.57
	stdev	0.28	0.28	0.27	0.27

RF-RIP-NONE vs. MLP-RIP-NONE in the 40% ratio (however RF still has higher prediction rates).

This suggests that RF is clearly the best-performing base black box for the self-learning grey-box. This result is not surprising since RF has proven to be a very competent classifier in different experimental studies [179, 47, 165, 157]. Furthermore, RF generally produces calibrated probabilities [118], which is a requirement for the later use of the amending based on class membership probabilities.

Figure 5.1 compares the SlGb against the RF black-box and different white-box baselines. Clearly, the SlGb outperforms the white box in terms of kappa, while it is not as accurate as the black box. The less labeled data is available, the more attractive results obtains the SlGb. It is important to remark that the goal is to outperform the white box while keeping interpretability to some extent.

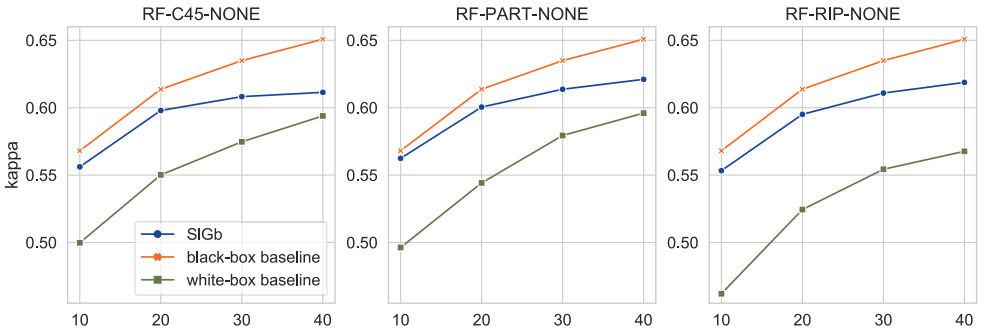


Figure 5.1: Comparison of SlGb performance in terms of kappa against the RF baseline and different white-box baselines. For the three combinations, the SlGb achieves less performance than the RF black-box baseline, however it outperforms all the white-box baselines.

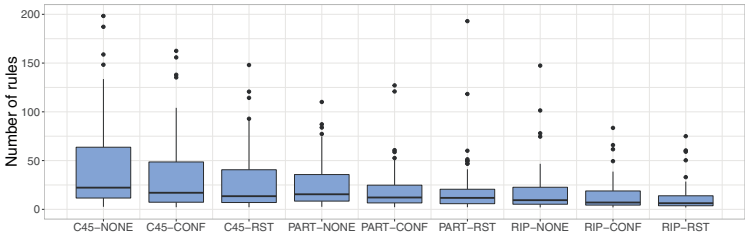
5.4 Impact of Using Different White Boxes and Amending Configurations

In this section, we study how different choices of the amending processes and white-box surrogates impact the overall results. We propose some measures for evaluating performance taking both accuracy and interpretability into account.

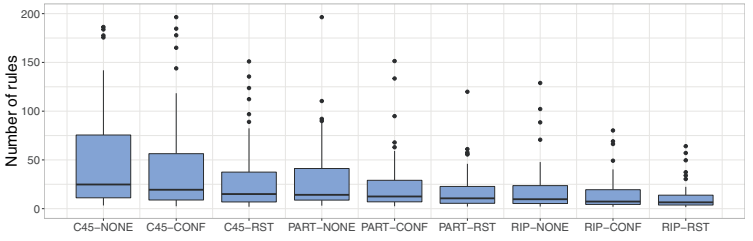
We first explore the influence on the prediction rates. Based on the selection of RF as the black-box component, Table 5.2 shows very similar results across each ratio. Going deeper with the statistical analysis, we apply Friedman and Wilcoxon tests with post-hoc correction. Although the Friedman test finds significant differences in the four groups (Table B.3), examining Wilcoxon corrected tests we ascertain that the null hypothesis cannot be rejected for the vast majority of pairs compared (see Tables B.4 and B.5 for details). This means that there are no statistically significant differences in the prediction rates when comparing different amending procedures with a fixed white box and vice versa. This behavior suggests that the overall prediction rates of the approach mostly rely on the correct choice of the black-box algorithm.

However, when examining the number of rules obtained, the difference is significantly visible. Figure 5.2 plots of the number of rules produced by each combination, per ratio of labeled data. Two results are consistent across ratios: both amending strategies (especially the RST-based one) reduce the number of rules while RIP as a surrogate white box produces the lowest number of rules for all possible combinations.

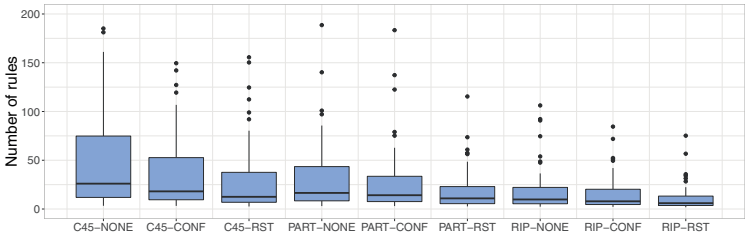
Toward exploring this result further, we also propose two new measures to evaluate models' interpretability via a quantifiable proxy. The first measure can be used in the



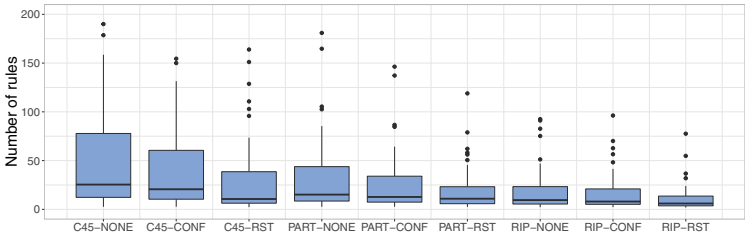
(a) Using 10% of labeled instances.



(b) Using 20% of labeled instances.



(c) Using 30% of labeled instances.



(d) Using 40% of labeled instances.

Figure 5.2: Number of rules produced by each combination of white box and amending, using random forests as black box. Both amending strategies (specially RST) reduce the number of rules while RIP white box produces the lowest number of rules.

Table 5.2: Prediction rates (kappa) achieved by different combinations of white boxes and amending strategies while using RF as black box. Results are grouped by ratio and best results are highlighted in bold. No significant differences were found in the kappa value when varying the choice of white box and amending.

	Ratio	10%	20%	30%	40%
RF-C45-NONE	mean	**0.56**	0.60	0.61	0.61
	stdev	0.28	0.27	0.27	0.27
RF-PART-NONE	mean	**0.56**	0.60	0.61	**0.62**
	stdev	0.28	0.27	0.27	0.27
RF-RIP-NONE	mean	0.55	0.60	0.61	**0.62**
	stdev	0.28	0.27	0.27	0.27
RF-C45-CONF	mean	0.55	0.59	0.61	0.61
	stdev	0.29	0.28	0.28	0.28
RF-PART-CONF	mean	0.56	0.60	0.61	**0.62**
	stdev	0.29	0.27	0.27	0.27
RF-RIP-CONF	mean	0.54	0.59	0.61	0.60
	stdev	0.29	0.27	0.27	0.28
RF-C45-RST	mean	0.56	0.60	**0.62**	0.62
	stdev	0.29	0.27	0.27	0.28
RF-PART-RST	mean	**0.56**	**0.61**	**0.62**	**0.62**
	stdev	0.28	0.27	0.27	0.27
RF-RIP-RST	mean	0.53	0.57	0.58	0.59
	stdev	0.28	0.28	0.28	0.28

context of self-labeling and the second one applies to any model containing explanation units. As mentioned in Section 2.4, there are three main forms of evaluating interpretability: application-grounded, human-grounded and functionally-grounded metrics. The functionally-grounded approach is the only form not requiring the involvement of humans. As an alternative, it uses desiderata for interpretability (e.g. transparency) as a proxy for assessing the interpretability of the model. Since we are working with benchmark datasets, we use the functionally-grounded approach for creating measures based on simplicity as a means to gain transparency and simulatability (see definitions in section 2.2). The first measure can be used in the context of self-labeling for base methods that produce tree structures, rules or decision lists. It involves the number of rules in the decision lists (or equivalently the number of leaves in a decision tree) and expresses the *relative growth* in structure as:

$$\Gamma = |E^g|/|E^w| \tag{5.1}$$

where E^g is the set of rules produced by the self-labeling method (here the grey box) and E^w is the set of rules produced by the baseline white box when using only labeled data. For this measure, a number much greater than one indicates that major growth in the structure of the self-labeling method is needed when using the extra unlabeled data. In that case, the balance between interpretability and performance must be taken into account for further evaluation.

The second measure is more general and applicable to any model whose structure is formed by quantifiable explanation units (e.g. rules, prototypes, features, derived features, etc.). For our case, this measure estimates the *simplicity* of the model according to the size of the structure in terms of the number of rules. Although the first notion would be that the smaller the rule set the better, this is not necessarily a linear relation. The desired simplicity in terms of the number of rules has a smooth behavior which can drop quickly. Therefore, we propose to measure simplicity through a generalized sigmoid function which has been historically used for fitting growth curves [15], since it allows representing this relation with enough flexibility. The simplicity can be formalized as the following equation:

$$\Upsilon(|E^g|) = \theta_1 + \frac{\theta_2 - \theta_1}{(1 + e^{-\lambda(|E^g| - \eta)})^{1/\nu}} \tag{5.2}$$

where $\theta_1 = 1$ and $\theta_2 = 0$ represent the upper and lower asymptotes of the function respectively, λ is the slope of the curve, η regulates the shift over the x-axis and ν affects near which asymptote maximum growth occurs. In this way, a result value of one indicates high simplicity and it decreases smoothly towards zero. A bigger λ would make the function less smooth, generating a drastic drop in simplicity after a threshold in the number of rules is surpassed. The value of η defines where the middle value of the function is obtained. While a value of $\nu = 1$ makes no change in the curve, $\nu < 1$ moves the growth toward the upper asymptote and $\nu > 1$ toward the lower one. Observe that both η and ν influence where 0.5 simplicity is obtained. Given the diversity of our benchmark data, we take $\lambda = 0.1, \eta = 30, \nu = 0.5$ for illustrating a general setting (see Figure 5.3).

With these values, the function produces medium evaluations (around 0.5) when the number of rules is around 40. Similarly, it obtains rather high simplicity (higher than 0.8) when the number of rules goes below 30. In real application scenarios, these parameters can be adjusted based on the feedback of domain experts. This highly flexible function allows customizing the value of simplicity according to the specifics of a given case study. The simplicity measure can be used as a proxy for transparency, especially as an indicator of simulatability which is the most subjective property, as seen in the definitions of Section 2.2.

Table 5.3 shows the average relative growth and simplicity achieved by the model, over the 55 datasets tested for the four ratios. Regarding the relative growth, the

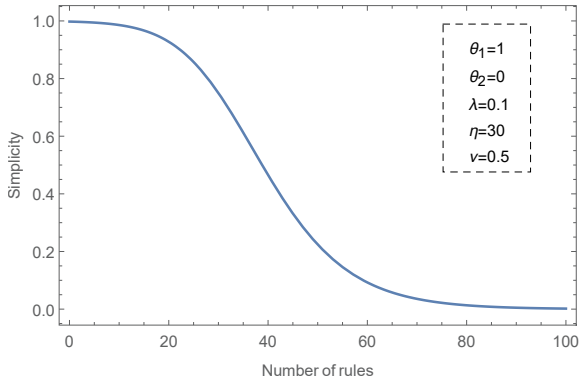


Figure 5.3: Simplicity function with default parameters used for the benchmark datasets. For specific applications these parameters are domain dependent.

increase in the structure of the grey-box is on average larger when using small amounts of labeled data, while for bigger ratios this difference decreases. This growth in the structure is an expected consequence of providing more unlabeled data to the white-box surrogate in the grey-box scheme. However, the use of amending procedures mitigates this effect by giving more importance to relevant unlabeled instances and less weight to less relevant instances. In general, a smaller growth is observed when using RST amending, especially in combination with PART as the white box, thus resulting in the winning combination for all ratios.

Besides, the simplicity measure (the closer the value to one the better) indicates that the use of amending is convenient for obtaining more concise sets of rules. It is also evident that using RIP as a surrogate generates the least number of rules, followed by PART. For this measure the absolute winner is RF-RIP-RST combination, exhibiting the highest values of simplicity for all ratio values used for experimentation. Similar statistical validation support this statement (see Tables B.6 and B.7), finding significant statistical differences when comparing RF-RIP-RST with other configurations using simplicity as interpretability measure.

It is important to remark that the simplicity measure solely quantifies to what extent it would be considered a simulatable model (see definition in Section 2.2). Of course, a very simple model with only one rule and poor prediction rates is not desirable, whereas for a very simple dataset three or four rules might be enough to reach accurate results. That is why taking into account the prediction performance is fundamental for a proper assessment. To measure algorithms' quality based on the balance between the prediction rates and the simplicity of the learned model, we

Table 5.3: Mean (and standard deviation) of the relative growth and simplicity achieved by different combinations of white boxes and amending strategies while using RF as black box. Results are grouped by ratio and best results are highlighted in bold.

	Ratio	10%	20%	30%	40%
RF-C45-NONE	growth	4.24 (2.98)	3.22 (3.81)	3.00 (6.61)	3.95 (15.12)
	simplicity	0.57 (0.44)	0.56 (0.45)	0.56 (0.45)	0.56 (0.45)
RF-PART-NONE	growth	3.07 (0.92)	2.19 (0.62)	1.78 (0.51)	1.55 (0.47)
	simplicity	0.70 (0.39)	0.70 (0.39)	0.69 (0.40)	0.69 (0.40)
RF-RIP-NONE	growth	3.93 (4.78)	3.19 (4.13)	2.93 (4.48)	2.67 (4.37)
	simplicity	0.85 (0.28)	0.84 (0.29)	0.84 (0.30)	0.84 (0.30)
RF-C45-CONF	growth	2.74 (2.38)	2.34 (3.35)	2.43 (5.96)	3.39 (13.75)
	simplicity	0.67 (0.42)	0.63 (0.44)	0.61 (0.45)	0.60 (0.45)
RF-PART-CONF	growth	2.11 (0.59)	1.66 (0.47)	1.45 (0.43)	1.30 (0.40)
	simplicity	0.81 (0.32)	0.78 (0.34)	0.75 (0.35)	0.74 (0.36)
RF-RIP-CONF	growth	2.96 (2.94)	2.52 (3.17)	2.41 (3.94)	2.54 (5.87)
	simplicity	0.89 (0.39)	0.88 (0.25)	0.87 (0.26)	0.86 (0.27)
RF-C45-RST	growth	2.26 (0.93)	1.53 (0.61)	1.20 (0.59)	1.00 (0.33)
	simplicity	0.71 (0.39)	0.71 (0.39)	0.71 (0.39)	0.71 (0.40)
RF-PART-RST	growth	**1.99 (0.49)**	**1.38 (0.31)**	**1.13 (0.24)**	**0.98 (0.21)**
	simplicity	0.82 (0.32)	0.81 (0.33)	0.81 (0.33)	0.81 (0.34)
RF-RIP-RST	growth	2.42 (2.26)	1.69 (1.06)	1.39 (0.63)	1.20 (0.42)
	simplicity	**0.91 (0.23)**	**0.92 (0.21)**	**0.93 (0.19)**	**0.94 (0.18)**

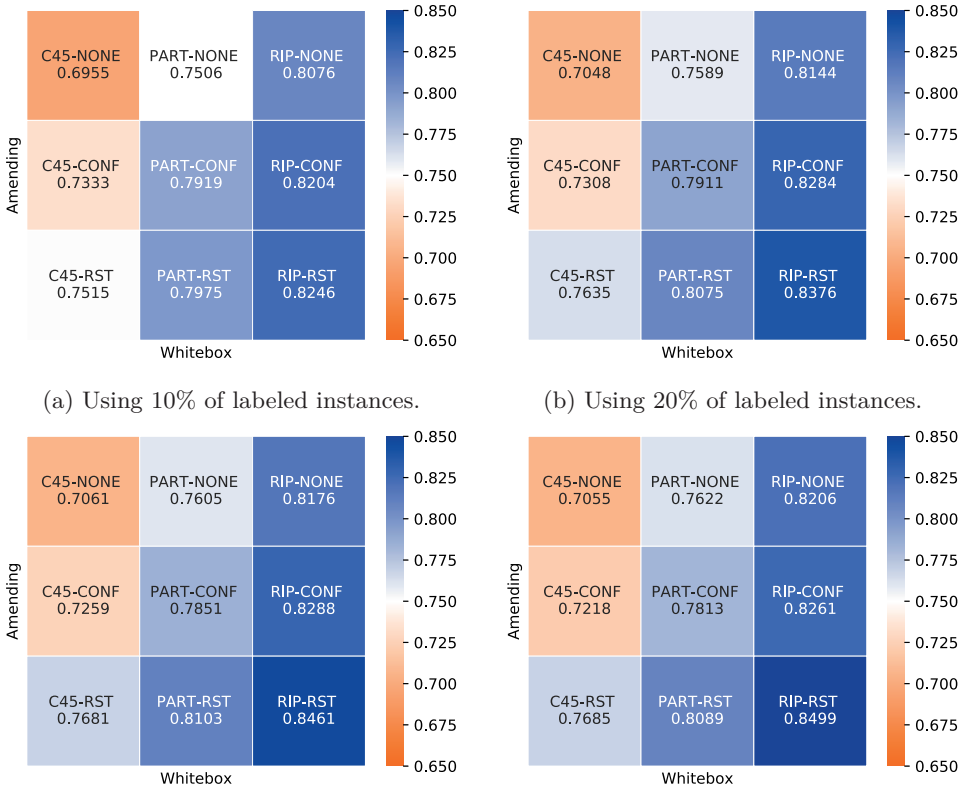
propose a third measure, called *utility*, combining the kappa (re-scaled to (0,1)) and the simplicity values with a weighting parameter α,

$$\Psi(E^g) = \alpha * \kappa(E^g)' + (1 - \alpha) * \Upsilon(|E^g|) \tag{5.3}$$

where α is set to 0.6 in our experimental setting, representing a scenario where the accuracy and the interpretability have almost the same preference. Utility functions are commonly used in multi-objective optimization for mapping a vector of pay-offs to a single scalar value [145]. In this case, the utility function is a linear combination of two terms parameterized by the weight α. This weighting parameter allows adjusting the preference of the user for prioritizing the accuracy or the interpretability objectives. Here, the two objectives are measured based on kappa and simplicity, respectively. It would be interesting to extend the proposed utility to involve more objectives where the parameters should be obtained from the preferences of a panel of domain experts [190, 144].

As a partial summary, Figure 5.4 visualizes the utility values in a heat-map plot. From this figure, it is easy to see that the RIP algorithm, as a white-box surrogate, positively contributes to the overall performance of the approach when taking both

kappa and simplicity into account. Additionally, RST amending also increases the value of utility when compared with CONF amending or not using amending at all. This measure reflects that, in general, the best trade-off is reached when using the RF-RIP-RST combination and the highest values are achieved when more labeled data is available.



(a) Using 10% of labeled instances.

(b) Using 20% of labeled instances.

(c) Using 30% of labeled instances.

(d) Using 40% of labeled instances.

Figure 5.4: Mean utility values of each combination of white box and amending, using random forests as the black-box base classifier. The use of RIP as a white-box component in combination with the RST-based amending achieves the best trade-off between accuracy and interpretability, for all explored ratios.

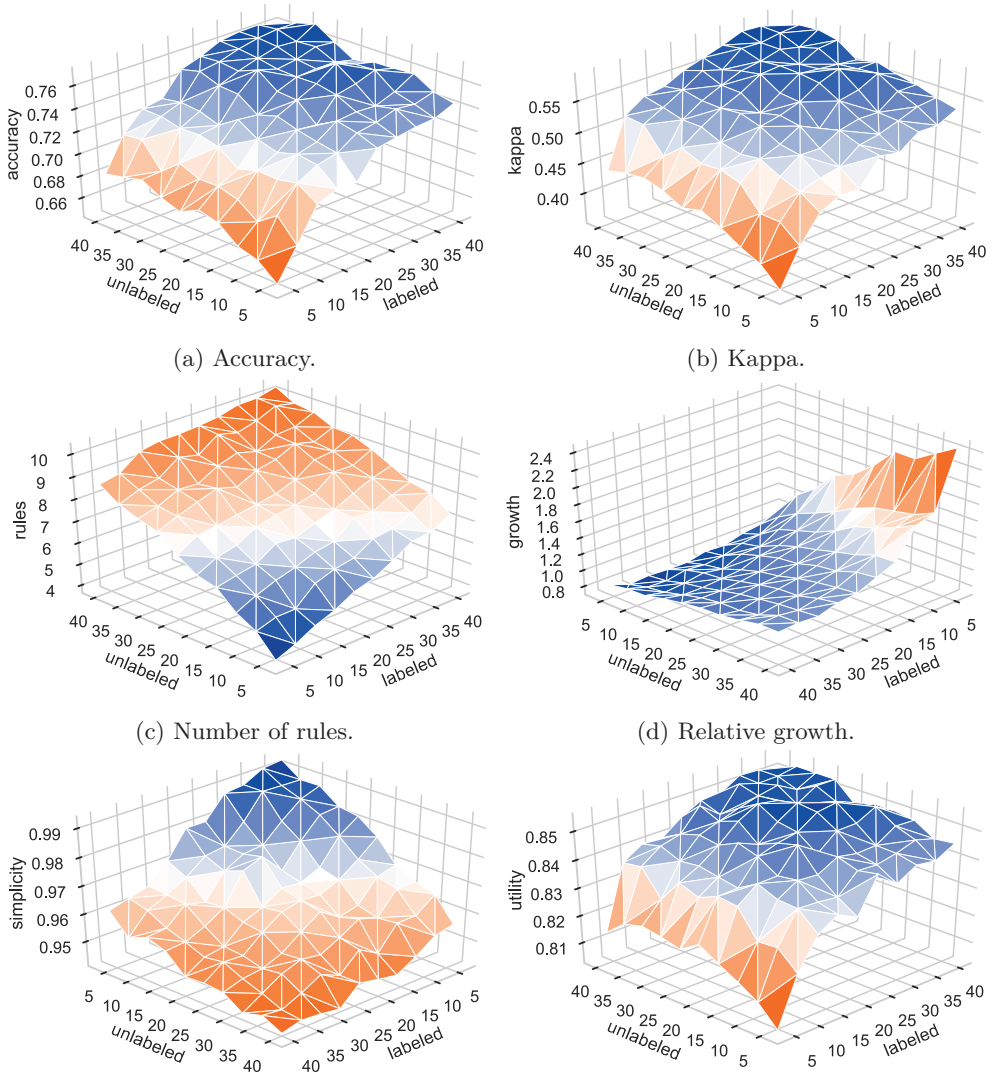
5.5 Influence of the Number of Labeled and Unlabeled Instances

In this section, we use RF-RIP-RST to explore the impact of having different amounts of labeled and unlabeled instances on the algorithm's results. In the evaluation of semi-supervised techniques, it is a common strategy to vary the size of L by systematically neglecting the label of different amounts of instances and adding them to U. But this procedure does not explore the scenario where also the unlabeled instances could be hard to obtain [123]. Observe that since this is a controlled experiment we can safely assume that the unlabeled instances follow the same class distribution as the labeled ones. In reality, one might need to re-balance the dataset after self-labeling if the unlabeled instances per-class distribution significantly differs.

Because we do not have truly unlabeled instances, we use the same datasets from the previous experiment. First, a test set with 20% of instances is kept aside for evaluation. Then, we divide the train set into two equally sized and disjoint subsets (each with 40% of the total instances). Each subset is a source for labeled and unlabeled instances, respectively, from where we vary the number of instances we use for training. Figure 5.5 shows the surfaces resulting from the average of different measures over the 55 datasets.

From the first two surfaces (Figures 5.5a and 5.5b), it can be observed that the prediction rates (accuracy and kappa) have a pronounced increment when adding more labeled and unlabeled instances. The most dramatic change is observed when adding labeled data to a few unlabeled instances (5%), which is an expected result as it tends to be a more supervised setting. However, when labeled instances are very limited (5% of the dataset), adding unlabeled instances clearly increases the overall performance. In addition, even when more labeled data is available (40%), an increase in performance is observed by adding more unlabeled data. This result confirms that our approach fulfills the main aim of SSC approaches.

The number of rules (Figure 5.5c) increases almost linearly with the number of training instances, either labeled or unlabeled. However, the relative growth (Figure 5.5d) is more sensitive to adding unlabeled data when labeled data is very scarce, i.e. a bigger amount of unlabeled instances rapidly increases the structure and loses in interpretability, compared with the baseline white box. However, the base white boxes generally perform very poor when the labeled data is scarce. When the labeled data is not so scarce, then the growth is more robust to adding more unlabeled data. This means that even when a base white box can achieve good performance with some labeled data, adding unlabeled data does not generate too much growth in structure and can benefit the performance of the grey-box (Figures 5.5a and 5.5b).



(a) Accuracy.

(b) Kappa.

(c) Number of rules.

(d) Relative growth.

(e) Simplicity.

(f) Utility.

Figure 5.5: Performance of RF-RIP-RST when varying the number of labeled and unlabeled instances, for different measures. Axes x and y are expressed in percentage of instances taken for training from each subset. Sub-figures (d) and (e) are rotated for visualization purposes.

The simplicity (Figure 5.5e) shows the expected behavior: the best values of this measure are observed with the least number of instances and it decreases uniformly in both directions. This means that adding more unlabeled instances does not generate a greater number of extra rules compared to adding more labeled instances. This is a consequence of using amending procedures for adjusting the confidence of the unlabeled instances, thus avoiding that the white box learns from inconsistent instances. Finally, the utility surface (Figure 5.5f) summarizes all results reflecting the increase in the overall performance when adding both labeled and unlabeled instances.

5.6 When the self-labeling grey-box works best?

In this section, we illustrate through an example when the use of SlGb can be advantageous and when it does not represent a significant gain in performance compared to baselines.

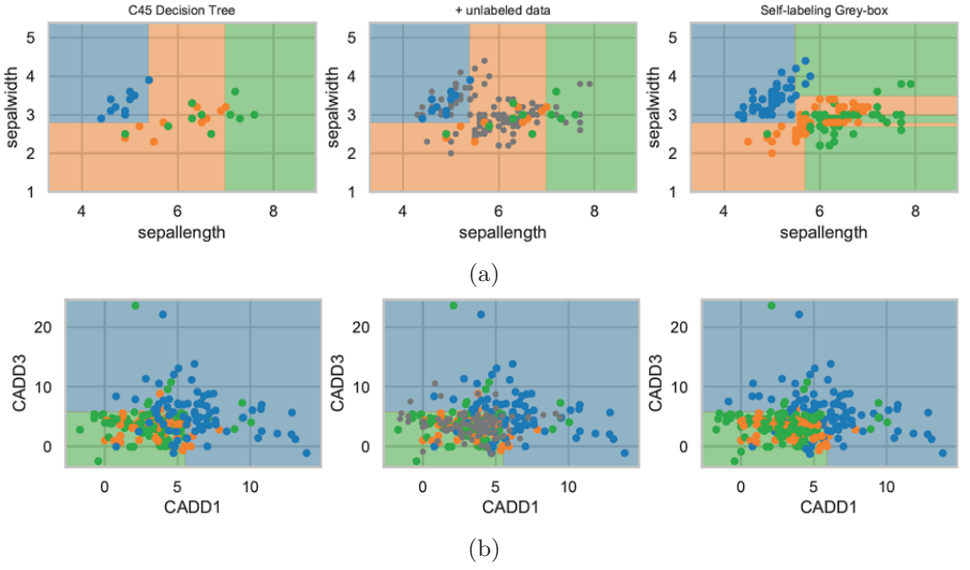


Figure 5.6: Decision boundaries for two datasets where SlGb significantly improves the classification (upper row) and does not report a big gain in performance (lower row). The first column portrays the boundaries computed by the white box baseline, the middle grid cell adds the unlabeled data points in grey, and the third cell shows how the boundaries are significantly changed (a) or left similar (b) by the SlGb.

Figure 5.6 represents two datasets: (a) the well-known iris classification problem and (b) the prediction of digenic effects of bi-locus genomic variants combinations [161]. Both classification problems have three classes that are represented by different colors in the space. For visualization purposes, we represent the boundaries drawn in a two-dimensional space using two features. In the first case (a), we can see that the boundaries drawn by the baseline white box, a C45 decision tree, are significantly altered by the SlGb after adding the unlabeled data. Here, the unlabeled instances provide useful information for adjusting these boundaries to fit all available data better. The performance of the SlGb is superior, and new explanations come out of the rules. On the other side (b), we can observe that the unlabeled data is not abundant compared to the labeled data, and it is practically surrounded by the labeled data as well. Consequently, the boundaries drawn by the SlGb are not significantly different from the white box baseline. Here, there is no big gain in performance, and therefore the use of a semi-supervised approach is not justified based on the available labeled and unlabeled data. The incorporation of more informative unlabeled data could change these results.

5.7 Comparing against State-of-the-Art

In this section, we compare the predictive capability of SlGb against the four best self-labeling techniques reported in the review paper in [158]: co-training using support vector machine [67] (CT(SMO)), tri-training using C45 decision tree [184] (TT(C45)), co-bagging using C45 decision tree [67] (CB(C45)) and democratic co-learning [183] (DCT). These four algorithms were evaluated against a pool of other 35 self-labeling techniques, using the standard benchmark of datasets that have been used through this chapter. These methods reported the best performance in inductive settings. However, since they are not inherently interpretable we focus our comparison on the prediction rates only. For this section, SlGb is instantiated with the RF-PART-RST combination, which exhibits the best results in terms of kappa, as shown in Section 5.4.

Table 5.4 reports the mean and standard deviation of the Kappa coefficient for each classifier, taking into account the four studied ratios. The results reveal that SlGb has the highest mean for all ratios. In order to support this assertion, we compute the Friedman p-value per ratio. The test suggests rejecting the null hypotheses for all labeled ratios based on a confidence interval of 95% (see Table B.8). This means that there is an indication that there exist significant differences between at least two algorithms in each comparison.

The next step is focused on determining whether the superiority of the SlGb classifier is responsible for the significant difference reported by the Friedman test. Similar

Table 5.4: Mean and standard deviation of kappa coefficient obtained by SlGb and four self-labeling methods from the state-of-the-art. SlGb outperforms the rest of the algorithms with statistical significance.

	10%		20%		30%		40%	
	mean	stdev	mean	stdev	mean	stdev	mean	stdev
SlGb	**0.56**	**0.29**	**0.61**	**0.27**	**0.62**	**0.27**	**0.62**	**0.27**
TT(C45)	0.51	0.29	0.55	0.29	0.57	0.29	0.59	0.29
CB(C45)	0.51	0.29	0.55	0.29	0.57	0.29	0.56	0.28
DCT	0.49	0.32	0.54	0.30	0.58	0.28	0.59	0.28
CT(SMO)	0.48	0.31	0.55	0.29	0.58	0.29	0.60	0.29

to previous sections we use the Wilcoxon signed-rank test and the Holm's post-hoc procedure for computing the corrected p-values associated with each pairwise comparison. Each section of the Table B.9 represents a ratio of labeled instances. The null hypothesis states that there is no significant difference between the performance of each pair of algorithms, taking SlGb as the control one.

From the statistical tests, we can draw the following conclusions. First, there is no doubt about the superiority of the SlGb classifier when tested with datasets with ratios of 10% and 20% of labeled instances, as all the null hypotheses were rejected. This result, in combination with the first place in the Friedman ranking, demonstrates that our algorithm significantly outperforms the other four algorithms in these settings. In the case of datasets comprising 30% and 40% of labeled instances, the results show that SlGb is the best-performing classifier, but with no significant differences observed between the pairs SlGb vs. DCT (for 30%), and SlGb vs. CT(SMO) (for both ratios), as these null hypotheses could not be rejected. However, DCT and CT(SMO) cannot be considered transparent due to their complex structure involving support vector machines and collaboration between base classifiers. Although our main goal was not to outperform the SSC methods in terms of classification rates, the analysis reported above supports our claim that we obtain a favorable balance between performance and interpretability by using the SlGb approach for solving SSC problems.

5.8 Concluding Note

In this chapter, we have evaluated the performance of SlGb on several benchmark datasets which are the standard for structured SSC tasks. The experiments showed that using random forests as the base black box for the self-labeling process is the best choice in terms of prediction rates. The choice of a white box and amending does not significantly affect the prediction rates but it is relevant for the size of the structure.

Three measures based on the number of rules were proposed for estimating the relative growth, simplicity, and utility of the SlGb. SlGb produces simpler models when using decision lists instead of a C4.5 decision tree as surrogate white boxes, even when no amending is performed. However, the amending procedures help further increase the simplicity (and therefore transparency) without affecting the prediction rates by giving more importance to confident instances in the self-labeling. Especially RST based amending looks more promising since it does not need the black-box base classifier to provide calibrated probabilities. Furthermore, RST based amending could be a good choice for a given case study where the uncertainty coming from inconsistency is high, even on the available labeled data. Therefore, we strongly advise the use of random forests as a base black box and RST for amending the self-labeling, while the choice of white box is more flexible to the desired interpretability, either a decision tree with rules or a decision list. Although, the best trade-off between accuracy and interpretability (utility) is reached when using the RF-RIP-RST combination.

The study varying the number of unlabeled instances and labeled instances together shows that even when the number of labeled instances is not that scarce, the SlGb is able to leverage unlabeled instances for increasing the performance. Another conclusion is that adding unlabeled instances does not make the interpretability worse compared to adding more labeled instances. This evidences that the amending procedure (in this case RST-based amending) avoids that the SlGb generates more rules from inconsistent instances. Finally, the experimental comparison shows that our SlGb method outperforms the state-of-the-art self-labeling approaches, yet being far more simple in structure than these techniques.

6 | Semi-supervised Classification of Genomic Variants

In this chapter, we illustrate the usability of SlGb in a SSC task from the medical informatics field. The classification problem at hand is the prediction of disease-causing variants in patients with a rare genetic disease. Since the labeling of the genomic variants according to their pathogenicity is a process that is usually performed manually, there exist a lot of unlabeled data available. Through the chapter, we describe the characteristics of this particular application and show the results that SlGb achieves.

6.1 Problem Description

The understanding of diseases has leaped forward since the introduction of the first draft of the human genome sequence [92, 160]. With the emergence of high-throughput sequencing technologies [139], targeted or whole exome and genome screening are becoming standard diagnostic resources in clinical settings to identify the variants associated with a genetic disease. The amount of data that is currently available has made the creation of computational tools possible for this prediction problem [87, 125], which have the potential to be an aid for personalized medicine.

Brugada syndrome (BrS) [57] is a heart condition that causes alterations in the normal rhythm of the heart (i.e. arrhythmia) which can lead to sudden cardiac arrest and death. BrS is known to be a rare genetic disease [63] which is associated with variants (i.e. differences in the DNA compared to a reference genome) in the SCN5A gene and other 25 genes [113, 174], although their role is still not completely defined [113]. The classification problem at hand aims to predict the pathogenicity of variants in BrS patients. The pathogenicity indicates whether a genomic variant is considered disease-causing. The pathogenicity can be described with five standard categories: pathogenic, likely pathogenic, uncertain significance, likely benign, and benign [142].

The process for classifying a BrS variant in one of these categories follows a set of recommendations formalized in a scheme published in [74]. This process is normally carried out manually by a clinical geneticist, which results in a tedious and time-consuming process. Thus, it makes the process of shaping the data to a curated labeled dataset for supervised learning a significant effort.

6.2 Knowledge Acquisition for Semi-Automatic Labeling

In an effort for generating more labeled data for this classification problem, we developed a semi-automated tool for knowledge acquisition [84]. GeVaCT, which stands for Genomic Variant Classification Tool, implements a variant classification schema for cardiac arrhythmia syndromes based on criteria from clinical geneticist from the Centre of Medical Genetics at UZ Brussel. This approach is supported by a yield of DNA testing over 15 years [74], between probands (i.e. a person serving as the starting point for the genetic study of a family) with isolated/familial cases, and also between probands with or without clear disease-specific clinical characteristics.

GeVaCT algorithm is implemented in two phases: pre-processing and labeling. The details of the implementation of the following steps are described further in the Appendix C. In the pre-processing phase, an annotated tab-delimited variant call file [32] generated from the Alamut Batch software [77], is refined based on a gene list for the disease of interest. This first step allows reducing the number of variants for the analysis. Secondly, we filter the data based on variants that have been already reported in the Human Genome Mutation Database [153] and in ClinVar [93, 94], or those which have been previously detected and classified in an internal patient population. And lastly, the variants are filtered based on their location in the genome and their coding effect, followed by the check for minor allele frequency of the variant in a control population [152]. These preprocessing steps select the variants of interest of the expert for performing the pathogenicity classification.

Thereafter, in the labeling phase, the filtered variants are grouped according to their type in *missense* or *nonsense and frameshift* variants. Missense variants are those that change a single nucleotide that results in a codon that codes for a different amino acid, altering the protein. Nonsense and frameshift variants can cause the premature termination of a protein. Altered proteins may be partially or completely inactivated, resulting in a change or loss of the protein function [120].

For missense variants the labeling process is based on the values of following parameters: amino acid substitution and its impact on protein function [90, 2], biochemical variation [108], conservation [133], frequency of variant alleles in a control population [82], effects on splicing [34], family and phenotype information and functional analysis. Whereas for the nonsense and frameshift variants, the label is based on effects on splicing, frequency of variant alleles in a control population, family and phenotype information, and functional analysis.

The calculation of each parameter mentioned above is detailed in Appendix C. For each computed parameter in the labeling process, a score is assigned to the variant, which is subsequently accumulated. Finally, based on the cumulative score, each variant is thereby labeled into one of the five classes of pathogenicity. A total of 65 variants were tested and validated as correct, by classifying them with GeVaCT and comparing them to their actual label. For a detailed description of the algorithm implemented in GeVaCT software as well as some images of the graphical and console user interfaces, the reader is referred to Appendix C.

As a result of the execution of GeVaCT, we can obtain some labeled annotated variants of BrS regarding its pathogenicity. However, some of the steps in GeVaCT still require the manual input of experts' criteria, thus making the labeling a semi-automated process. In consequence, the limitation of the manual labeling process is not removed and a large amount of data remains unlabeled for this classification task. In the next section, we describe the characteristics of the dataset that result from joining the labeled and unlabeled data.

6.3 Dataset Characterization

After preprocessing with GeVaCT, a dataset of 1181 samples was obtained. Extra preprocessing steps included removing non-informative or redundant attributes generated by Alamut Batch software (e.g. with no variance, too much variance, related with the location of the variant, etc.). This criterion was supported by interviews with clinical geneticists. A total of 30 attributes were taken into account for classification. From the resulting instances, the 69% are unlabeled instances, i.e. pathogenicity is unknown. Figure 6.1 depicts the highly imbalanced class labels distributions.

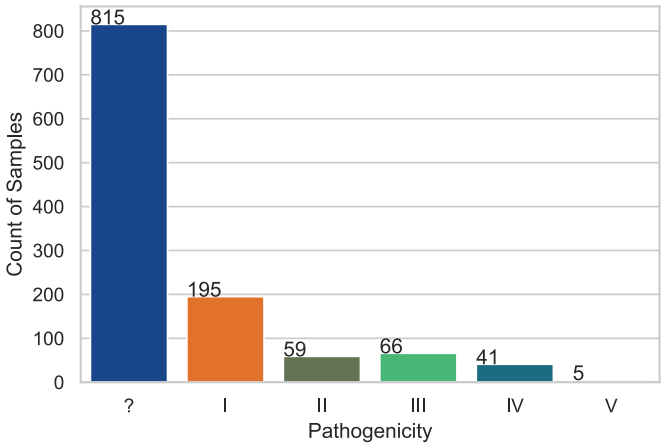


Figure 6.1: Distribution of variants according to their pathogenicity, where I means *non-pathogenic*, II means *unlikely pathogenic*, III means *unclear*, IV means *likely pathogenic* and V means *pathogenic*. The blue bar represents 815 variants with unknown class which represents a 69% of unlabeled data with respect to the entire dataset. In addition, from the 31% of labeled data (362 samples), a high imbalance is observed, with class I having the majority of instances and class V having only 5 samples.

Table 6.1: Cost matrix for misclassifications based on expert criteria which reflects the ordinal character of the classification problem.

	True values	I	II	III	IV	V
Classified as	I	0.0	0.2	0.8	1.0	1.0
	II	0.5	0.0	0.6	0.8	1.0
	III	0.8	0.6	0.0	0.6	0.8
	IV	1.0	1.0	0.5	0.0	0.2
	V	1.0	1.0	1.0	0.5	0.0

Additionally, from the conception of the default approach implemented in GeVaCT (see Appendix C), we realized that the cost of misclassifications in this problem is not symmetric or equal. The last step of the labeling in GeVaCT (see Figure C.2) is a manual input of a score based on the judgment of the expert about literature at the moment, and information about the family of the patient. As a result of this input, it is possible that the score that determines the class increases, e.g. from *likely pathogenic*

to *pathogenic*, but it is less probable to change from *unlikely pathogenic* to *pathogenic*. However, SlGb could predict a variant as *pathogenic* that the expert considered *likely pathogenic*, because there was not enough information in the literature available at the moment. After consultation with clinical geneticists, a cost matrix was conceived for penalizing the errors of SlGb taking into account this characteristic (see Table 6.1). This cost matrix is used for weighting the calculation of the confusion matrix and the performance measures derived from it. During the learning process, it penalizes the classifier harder when committing more relevant errors. The matrix reflects the ordinal character of the classification problem.

6.4 Experimental Results and Discussion

In this section, we explore the SlGb performance through 10-fold stratified cross-validation. On each iteration, an SlGb predictor is evaluated on a different 10% of data (only labeled instances) while the remaining 90% (containing labeled and unlabeled instances) is used for training. Following the recommendations for the evaluation of semi-supervised classifiers (see Section 3.3) we also evaluate the supervised baselines for comparison. In this case, we use only labeled instances and the same test data for each fold. Following the results found in the experimentation with benchmark data, RF was the best performing classifier for acting as the self-labeling base. In the same way, we decide to use RST-based amending since it demonstrated to improve interpretability in terms of simplicity while maintaining accuracy. Decision trees and decision lists alternatives for white-boxes are explored. Tables 6.2 and 6.3 show the results achieved in terms of accuracy and interpretability, respectively.

First, from Table 6.2 we can conclude that SlGb outperforms its white-box baselines for the three configurations tested. Overall using PART as a white box provides the best results in terms of accuracy and kappa values. The sensitivity shows that it is easier to obtain true positives from class I and more difficult for class V, although the specificity for class V is high, which means that almost no samples are misclassified as pathogenic. Both kappa and Mathew's correlation coefficient (Mcc.) values support the performance of SlGb despite the high imbalance in the distribution of the classes.

Secondly, from Table 6.3 we can observe that PART and RST generated the least number of rules. The resulting decision list generated by PART was especially concise compared to its white-box baseline, even reducing in 9% the number of rules needed for achieving better accuracy. Since PART is a sequential covering algorithm, this indicates that the unlabeled data allowed PART to generate rules with more support which were also more accurate. Having the least number of rules, the simplicity of SlGb using RIP is very high with 0.96. Therefore, the utility, which gives a measure of the trade-off between accuracy and interpretability, is best valued also with RIP.

Table 6.2: Mean performance achieved by three configurations of SlGb, using different white boxes. The performance is measured in accuracy (Acc.), Cohen's kappa (Kap.), Sensitivity (Sen.), Specificity (Spe.), Precision (Pre.) and Mathew's Correlation Coefficient (Mcc.). The last five measures are shown by class and as a weighted average. All SlGb configurations outperform baseline white boxes, with SlGb using PART decision list achieving the best results.

Algorithm	Acc.	Kap.	Per Class	Sen.	Spe.	Pre.	Mcc.
SlGb (RF-C45-RST)	0.76	0.62	W.Ave.	0.76	0.88	0.75	0.65
			I	0.92	0.83	0.85	0.75
			II	0.54	0.95	0.64	0.51
			III	0.68	0.92	0.65	0.59
			IV	0.53	0.97	0.66	0.55
			V	0.20	0.99	0.25	0.21
SlGb (RF-PART-RST)	**0.78**	**0.66**	W.Ave.	**0.78**	**0.92**	**0.78**	**0.69**
			I	0.89	0.91	0.91	0.79
			II	0.64	0.94	0.64	0.57
			III	0.68	0.90	0.59	0.54
			IV	0.65	0.98	0.75	0.66
			V	0.20	0.99	0.33	0.25
SlGb (RF-RIP-RST)	0.75	0.60	W.Ave.	0.75	0.86	0.74	0.63
			I	0.92	0.78	0.82	0.71
			II	0.47	0.96	0.66	0.49
			III	0.66	0.91	0.61	0.55
			IV	0.53	0.97	0.68	0.56
			V	0.40	1.0	1.0	0.63
RF	0.82	0.72	W.Ave.	0.82	0.91	0.82	0.75
C45	0.68	0.50	W.Ave.	0.68	0.87	0.68	0.55
PART	0.69	0.52	W.Ave.	0.69	0.89	0.69	0.57
RIP	0.68	0.47	W.Ave.	0.68	0.78	0.66	0.50

It should be noticed that RIP produces a decision list with rules covering from the most rare class labels to the more common ones (see Figure 6.2). This means that this model can be very useful for interpretation when a short set of rules uncovering the rarest patterns is needed. On the contrary, PART (see Figure 6.3) builds a decision list starting with the best supported and most confident rules of any class label.

Table 6.3: Performance in terms of interpretability for each configuration of the SlGb. The number of rules, relative growth (see Equation 5.1) and simplicity (see Equation 5.2) measure transparency as a proxy for interpretability. Utility (see Equation 5.3) measures the trade-off between accuracy and interpretability. The most concise list of rules is produced by SlGb using RIP as white box.

Algorithm	Rules	R.Growth	Simplicity	Utility
SlGb (RF-C45-RST)	41	1.10	0.43	0.66
SlGb (RF-PART-RST)	**31**	**0.91**	0.72	0.78
SlGb (RF-RIP-RST)	**15**	**1.0**	**0.96**	**0.86**

```
(phyloP >= 0.74) and (rsMAF >= 0.23) and (varAApolarity <= 0.45) =>
class=V (3.52/0.0)

(phyloP >= 0.92) and (granthamDist >= 0.29) and (phyloP <= 0.94) =>
class=V (2.35/0.0)

(AGVGDgd >= 0.64) => class=IV (13.61/0.0)

(rsMAF <= 0.001) and (AGVGDgd >= 0.40) and (phastCons >= 0.98) =>
class=IV (5.44/0.0)

(varCodonFreq >= 0.56) and (posAA >= 0.41) and (varCodonFreq <= 0.56)
=> class=IV (6.35/0.0)

(wtCodonFreq >= 1) and (wtAAcomposition >= 0.047) => class=IV (2.72/0.0)

(exacAllFreq <= 0.003) and (granthamDist >= 0.42) and (SIFTweight >=
0.02) and (posAA >= 0.28) => class=II (23.32/1.88)

(exacAllFreq <= 0.00) and (wtAApolarity >= 0.79) => class=II (14.51/4.77)

(posAA <= 0.02) and (phyloP >= 0.72) => class=II (9.71/2.89)

(rsMAF <= 0.0004) and (AGVGDgv <= 0.24) => class=III (18.04/2.99)

(rsValidations = Cluster/Frequency/HapMap/1000G) and (phyloP >= 0.64) =>
class=III (20.80/3.86)

(exacAllFreq <= 0.00007) => class=III (9.47/2.89)

(rsMAF <= 0.004) and (BLOSUM62 <= 0.16) and (varCodonFreq >= 0.36) =>
class=III (6.58/0.0)

(posAA <= 0.01) and (wtAAcomposition >= 0.25) => class=III (3.76/0.0)

otherwise => class=I (219.05/28.49)
```

Figure 6.2: Decision list built by SlGb using RIP as a white box. It contains 15 rules which need to be interpreted in order, from more rare patterns corresponding to class V, to more common ones. The default rule assigns class I to the instances that were not covered by any of the previous rules.

```
AGVGDgd <= 0.64 AND rsValidated = no AND nOrthos <= 0.55 AND wtCodonFreq >
0.14 AND nOrthos > 0.11: III (16.03/0.97)
AGVGDgd > 0.64: IV (14.79/1.18)
phyloP <= 0.70 AND distNearestSS <= 0.23 AND phastCons <= 0.008: II
(10.75/1.01)
phyloP <= 0.70 AND distNearestSS > 0.23 AND rsValidated = yes AND BLOSUM62 >
0 AND granthamDist <= 0.56 AND rsValidations = Cluster/Frequency/1000G AND
rsMAF > 0.004: I (79.65)
nearestSSChange <= 0.569008: III (2.82)
phyloP <= 0.64 AND rsValidated = yes AND varAAvolume <= 0.79 AND BLOSUM62 > 0
AND distNearestSS > 0.23 AND granthamDist <= 0.45 AND AGVGDclass = C0 AND
exacAllFreq > 0.00009 AND wtAApolarity <= 0.91 AND substType = transversion
AND posAA > 0.039: I (33.27)
SIFTweight > 0.20 AND granthamDist <= 0.32 AND phastCons <= 0.99: I
(32.23/0.97)
AGVGDgd > 0.37 AND phastCons <= 0.97 AND AGVGDgv <= 0.50: III (7.53)
AGVGDgd > 0.37 AND phastCons > 0.97: IV (12.2/3.13)
phyloP <= 0.82 AND varAApolarity <= 0 AND phyloP > 0.49: II (9.75)
phyloP <= 0.82 AND varAApolarity <= 0.06 AND BLOSUM62 <= 0.16: III (7.53)
phyloP <= 0.82 AND substType = transversion AND nearestSSType = 3 AND
varAAvolume <= 0.17: II (6.82)
phyloP <= 0.82 AND varCodonFreq > 0.07 AND substType = transversion AND
conservedOrthos <= 0.53 AND exacAllFreq > 0.0003: III (5.71/1.01)
phyloP <= 0.82 AND varCodonFreq <= 0.07: III (5.58/1.81)
phyloP > 0.87 AND SIFTweight <= 0.15 AND rsMAF > 0: III (8.5/0.97)
phyloP <= 0.87 AND phyloP > 0.63 AND BLOSUM62 <= 0.66 AND exacAllFreq <=
0.003: II (13.61/0.94)
phyloP <= 0.82 AND rsValidated = no AND distNearestSS > 0.23: II (2.92)
rsValidated = yes AND exacAllFreq > 0.00007 AND phyloP <= 0.63 AND
varAApolarity <= 0.82 AND varCodonFreq > 0.50: I (21.17)
rsValidated = yes AND phyloP <= 0.61 AND varCodonFreq <= 0.43 AND BLOSUM62 >
0.16 AND varAAcomposition > 0.23: I (12.1)
rsValidated = yes AND wtAAcomposition > 0.33: II (5.81/0.94)
rsValidated = no: IV (4.64/1.01)
SIFTmedian <= 0.5 AND SIFTweight <= 0.16: IV (4.54)
rsValidations = Cluster/1000G: I (7.19/2.15)
conservedOrthos > 0.30 AND varCodonFreq <= 0.15: II (6.79/2.89)
conservedOrthos > 0.30 AND varAApolarity <= 0.45 AND varAAcomposition <=
0.07: I (4.03)
conservedOrthos > 0.30 AND AGVGDgd <= 0.18 AND nOrthos > 0.11 AND
varCodonFreq > 0.39: IV (5.55/1.01)
conservedOrthos > 0.30 AND rsValidations = Cluster: III (5.65)
wtAApolarity > 0.50 AND distNearestSS > 0.237235: I (4.03)
distNearestSS > 0.23: II (2.92)
distNearestSS <= 0.23: III (2.82)
default: V (2.35)
```

Figure 6.3: Decision list built by SlGb using PART as a white box. It contains 31 rules which need to be interpreted in order, from more confident patterns to a default rule. The default rule assigns class V to the instances that were not covered by any of the previous rules.

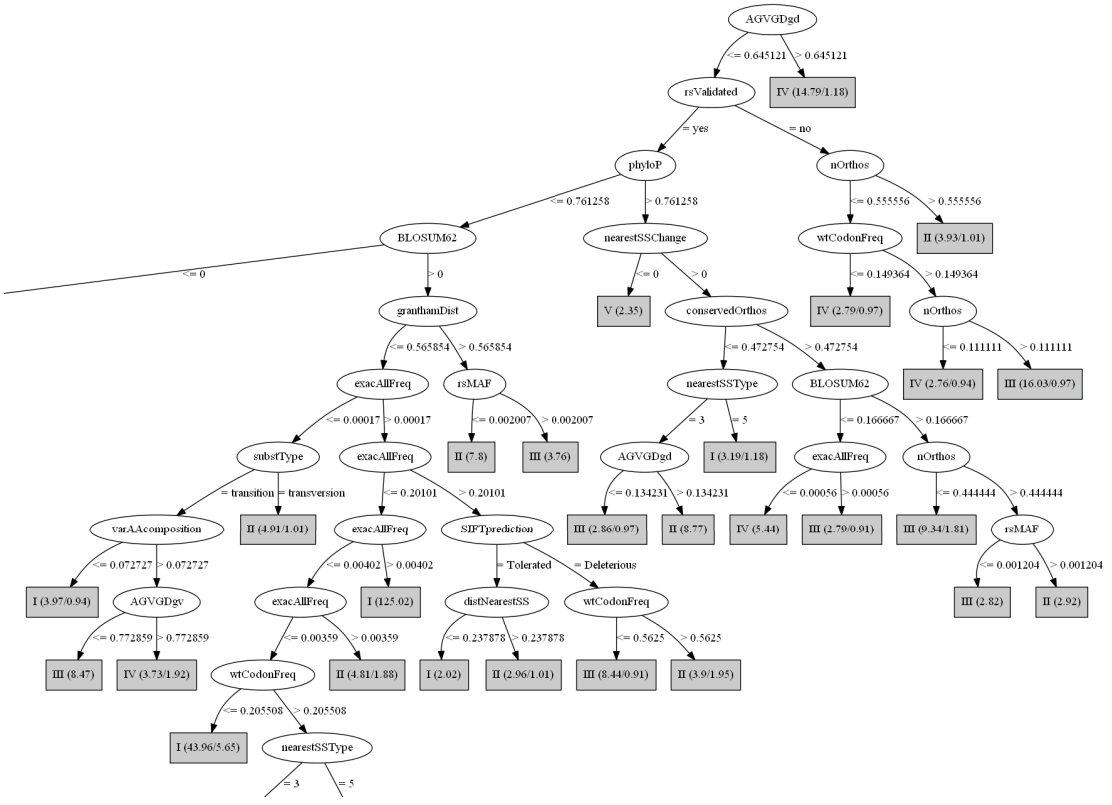


Figure 6.4: Decision tree built by SlGb using the C45 decision tree as a white box. The decision tree contains 41 rules when traversed from the root to each of its leaves. Three branches of the tree were omitted for visualization purposes. The omitted branch on the left leads to nine leafs, the majority classifying the instances as *non-pathogenic* by evaluating two more conditions. The two omitted branches in the bottom lead to five leafs by evaluating two more conditions.

Figures 6.2, 6.3, and 6.4 are shown to illustrate the interpretability that is possible to obtain from SlGb in the form of *if-then* rules. For all models, the first number in parenthesis after the conclusion of the rule denotes the weighted number of instances that the rule covers. The second number, if present, represents the weighted number of instances that were misclassified in that rule. By subtracting these two numbers we obtain the weighted number of true positives of the class label that is being predicted in that rule. In addition, since the values of the attributes are normalized due to implementation details of the RST-amending, the conditions should be re-scaled to their original values before literal interpretation. However, using the re-scaled values gives a more qualitative interpretation of the interaction between small or large values of different attributes in the antecedents of the rules.

Some patterns observed in the rules show relations with the expert criteria implemented in GeVaCT (see Appendix C). For example, high values of *phyloP* attribute in combination with high values of *granthamDist* attribute led to identify the variant as pathogenic. Further experiments and feedback from experts are needed to improve the fitness of the model and validate the rules that can be obtained.

Finally, although less impressive in performance, the SlGb using decision trees offers further possibilities for the integration of expert knowledge. The hierarchical structure of the tree (see Figure 6.4) allows replacing one attribute by another one that is related, for example in a superclass or subclass relation. For this particular case study, we could take instead of the allele frequencies for specific populations, the global value, or vice versa. These replacements could be supported by the use of an ontology that describes the relations between the attributes considered in the prediction problem. The involvement of the human experts in the loop is an interesting future research direction.

6.5 Concluding Note

This chapter illustrated the usability of SlGb as a semi-supervised classifier for a particular case study. We proposed GeVaCT, a software that automatizes a pipeline of expert criteria for labeling variants according to their pathogenicity. Although the intention was to facilitate the labeling of genomic variants, this software is still a semi-automated solution. Some steps in GeVaCT depend on manual input based on the experts' experience and their knowledge of literature. Therefore, it does not remove the limitation of manual labeling and thus the lack of labeled data. A total of 815 remaining unlabeled instances were leveraged for addressing this prediction problem as SSC. We showed that SlGb is a good predictor of the pathogenicity of BrS variants compared to its white box baselines. It not only produces better results in terms of several measures of accuracy but also maintained or improved the interpretability

when compared with the white-box baseline. Although some patterns related to the GeVaCT pipeline were observed, further experiments and feedback from experts are needed to adjust and validate the obtained rules.

7 | Semi-supervised Prediction of Early Protein Folding

This chapter illustrates the application of SlGb in the prediction of the early folding in proteins. Proteins are chains of amino acid residues that fold into a three-dimensional structure that influences its function. Early folding mechanisms are crucial for the protein folding and for understanding the protein behavior. Although very few labeled residue-level data is available, a successful predictor named EFoldMine [135] is able to identify early folding residues using features derived from the protein sequence. Although EFoldMine is able to detect 75% of early folding residues, it is based on a support vector machine classifier, which does not allow for direct interpretation. A large amount of protein sequence data (without early folding labels) is available from curated biological databases. Through this chapter, we investigate the use of SlGb to gain mechanistic residue-level insights into the determinants of early folding regions in proteins.

7.1 Problem Description and Dataset Characterization

The understanding of how a protein folds in a three-dimensional structure is fundamental for the study of their functionality and behavior [124]. When the folding process of a protein is not successful it can lead to the appearance of diseases such as Alzheimer's, Parkinson's, or type II diabetes [140, 38]. The early folding process is led by amino acid residues that are close to each other in the protein sequence, thus leading to overall structure formation. However, studying protein folding is a difficult task that requires highly time-sensitive and complex experiments [124], therefore the available labeled data is limited. In this regard, a curated database named Start2Fold [124], collects data from those experiments and provides a source of labeled residues (as early folding or not).

Even though this information is limited, a successful machine learning predictor named EFoldMine [135] was proposed for detecting early folding residues. EFoldMine was trained with data from 30 proteins from Start2Fold. This predictor uses numerical features computed with DynaMine [25, 26] for representing the *backbone* dynamics of each protein. For EFoldMine, the authors also compute four other features with a similar approach to DynaMine (see [135] for details), which describe the *side-chain* dynamics and the secondary structure formation propensity (*alpha-helix*, *beta-strand* and *coil*).

For each numerical descriptor, the authors consider a window of -2 to +2 residues. In this way, the vector of attributes describing each residue has values of the five properties for the residue itself (denoted with suffix 0) and its neighboring residues (with suffixes +2, +1, -1, and -2). This results in a vector of 25 attributes describing each instance. There are a total of 3,398 labeled instances, from which 482 are labeled as positive (early folding residue) and 2,916 as negative.

The EFoldMine predictor uses an SVM with a radial basis function kernel, with hyper-parameters $C = 100$ and $\gamma = 0.04$ while applying Platt's scaling [132] for the estimated probabilities. SVMs are known to be highly accurate classifiers based on strong mathematical foundations, the resulting model is however a black box in terms of interpretability. This hinders the possibility of extracting further knowledge about the determinants of early folding in proteins. One alternative is the use of more intrinsically interpretable machine learning techniques such as decision trees or rule-based algorithms. However, as mentioned before, these algorithms tend to be less performant compared to black boxes.

In addition, the labeled residue-level data available for extending this experimentation to other machine learning approaches allowing interpretation is limited to the data in Start2Fold. Therefore, we propose to tackle this problem as a SSC task. We

leverage unlabeled data from other proteins without experimental information about early folding, thus enlarging the current training data. The goal is to obtain better performance compared to the use of a supervised interpretable classifier. Consequently, we aim to obtain a model that can be still considered interpretable, as opposed to a black-box one. In the next section, we explore the ability of the SlGb approach to achieve a good trade-off between performance and interpretability.

7.2 Experimental Results and Discussion

In the following experiments, SlGb uses the EFoldMine predictor as a black-box base classifier for labeling extra unlabeled data[1]. As white boxes, we explore the decision tree (C45) and decision lists (PART and RIP) classifiers evaluated in Chapter 5. The extra unlabeled data comes from the PISCES database [166], which contains only proteins with a known overall structure. The unlabeled data comprises a total of 665,323 residues from 3,050 proteins. Similarly to EFoldMine, the class weight of the labeled data is modified for taking into account the class imbalance (see Figure 7.1).

In order to avoid bias in the validation, a stratified cross-validation is performed. Similarly to the EFoldMine validation, we divide the training set in 27 folds, each representing labeled proteins (two folds contain data from more than one protein, see [135] for details). In this way, we guarantee that residues from the same protein are not split in the training and test set. For the semi-supervised setting, we add an equal number of proteins from the unlabeled data to each fold. This means that all folds have extra unlabeled residues coming from 113 proteins, except for the last one which has 112 proteins. The total number of residues on each fold varies according to the number of residues in the proteins included.

Table 7.1 summarizes the results in terms of the performance of the different configurations of SlGb tested and its baseline classifiers. First, we conclude that SlGb outperforms its white-box baselines for the three configurations tested, making it a suitable alternative to using the white boxes alone. From the table, we conclude that decision lists offer the best performance across different measures compared to the EFoldMine black-box baseline. PART and RIP both have the best true positive rates (sensitivity) with a value of 0.7 at a cost of a 0.31 and 0.28 false positive rate (1 - specificity), respectively. For both of these configurations the low false positive rate is also evidenced by the higher precision (ratio of true positive residues to the total predicted positive residues). The decision tree configuration (C45) shows the best true negative rate (specificity) but at a cost of being more "conservative" in detecting

[1]For code compatibility with the white boxes, we reproduced the EFoldMine predictor using *weka* library [172] for the support vector machine instead of the original implementation with *scikit-learn* library [129]

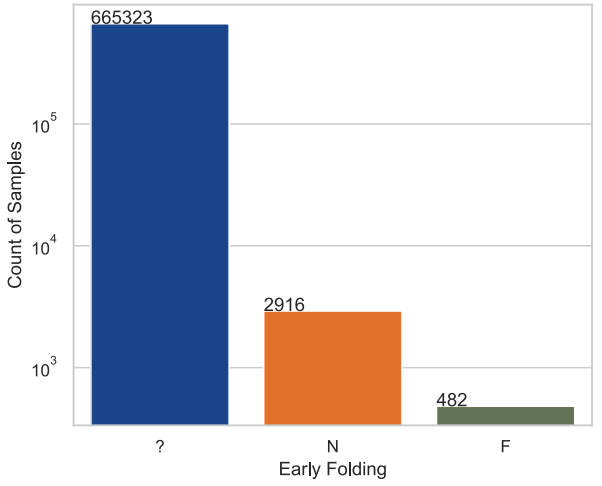


Figure 7.1: Distribution of residues according to their label, where '?' means *unknown*, 'N' means *not early folding residue* and 'F' means *early folding residue*. The y-axis is in log scale for better visualization. The blue bar represents 665,323 residues with unknown class which accounts for the 99.5% of data. In addition, from the 0.5% labeled data (3,398 residues), a high imbalance is observed, where 2,916 residues are labeled as negative and 482 are labeled as positive.

early folding residues. The kappa and MCC measures, which both account for imbalance in the dataset, support the conclusion that SlGb using PART or RIP are the best performing configurations. This is inline with the results of the experiments on benchmark datasets performed in Chapter 5.

Table 7.2 summarizes the results in terms of interpretability measures. SlGb using RIP or PART generates more concise models compared to using the decision tree. The relative growth of SlGb using RIP compared to its baseline white-box classifier is higher compared to SlGb using PART or C45. However, the SlGb using RIP has the highest simplicity and utility while SlGb using PART also achieves very competitive results. Therefore, for this study, we recommend interpreting the results using decision lists since they produce more transparent models (measured as simplicity). We discussed these results with an expert which is the principal investigator of EFoldMine [135]. The recommendation above was confirmed by the expert, which found the decision lists more concise than the decision tree, and in particular, the SlGb using RIP easier to understand. In addition, some of the rules obtained in the decision lists were expected

Table 7.1: Mean performance during 27-fold cross-validation achieved by three configurations of SlGb, using different white boxes. The performance is measured in sensitivity (Sen.), specificity (Spe.), accuracy (Acc.), balanced accuracy (Bac.), precision (Pre.), Mathew's correlation coefficient (Mcc.), area under the ROC curve (Auc.) and Cohen's kappa (Kap.). Sen., Spe., Prec., Mcc., and Auc. are measured with respect to the positive class. All SlGb configurations outperform baseline white boxes, with SlGb using RIP decision list achieving the best results for the majority of the measures.

Algorithm	Sen.	Spe.	Acc.	Bac.	Pre.	Mcc.	Auc.	Kap.
SlGb (SVM-C45-CONF)	0.59	**0.78**	0.66	0.68	**0.69**	0.35	0.68	0.32
SlGb (SVM-PART-CONF)	**0.70**	0.69	0.68	0.70	0.67	0.38	**0.73**	0.35
SlGb (SVM-RIP-CONF)	**0.70**	0.72	**0.69**	**0.71**	**0.69**	**0.40**	0.72	**0.37**
EFoldMine	0.73	0.76	0.73	0.74	0.36	0.35	0.81	0.40
C45	0.50	0.78	0.63	0.64	0.66	0.27	0.63	0.25
PART	0.57	0.66	0.61	0.62	0.61	0.23	0.65	0.21
RIP	0.68	0.68	0.66	0.68	0.66	0.35	0.70	0.32

Table 7.2: Performance in terms of interpretability for each configuration of the SlGb. The number of rules, relative growth (see Equation 5.1) and simplicity (see Equation 5.2) measure transparency as a proxy for interpretability. Utility (see Equation 5.3) measures the trade-off between accuracy and interpretability. The most concise list of rules is produced by SlGb while using RIP as the white box, with PART offering competitive results.

Algorithm	Rules	R.Growth	Simplicity	Utility
SlGb (SVM-C45-CONF)	215	1.23	1.8E-8	0.39
SlGb (SVM-PART-CONF)	58	1.38	0.11	0.44
SlGb (SVM-RIP-CONF)	55	4.23	0.14	0.46

and matched previous knowledge of the biophysical domain, while others exhibited meaningful new patterns that should be investigated further.

7.2.1 Further Interpretation of the Decision Lists

Decision lists are interpreted in descending order by testing each rule until obtaining a true antecedent or the default rule is reached. Figure 7.2 shows the structure and interpretation of an individual rule generated by PART or RIP. While PART prioritizes rules with the best confidence and support predicting any class label, RIP focuses on the patterns that predict the minority class first. For this application, this implies

that RIP decision lists will provide rules for identifying early folding residues, which is the class label of interest.

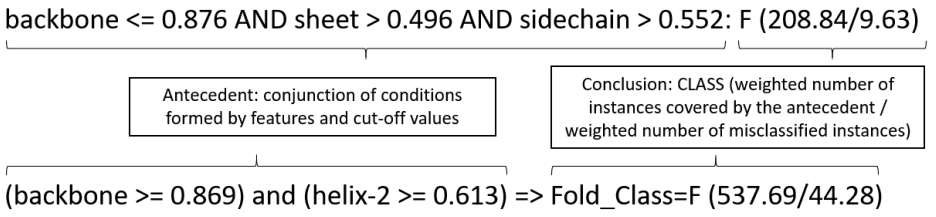


Figure 7.2: Interpretation of the rules generated by PART (above) and RIPPER (below) decision lists.

An additional layer of interpretability is used for extracting information from the rules generated by SlGb using the RIP decision list. In an attempt to detect paired relations between conditions (features and their values above given cut-offs) that determine early folding, we plot the true positive count (early folding residues detected) they yield accumulated over the set of rules. We performed this experiment for all paired combinations of features and extracted the pairs of relations that lead to the 70% of true positives, i.e. the bottom 30% weaker relations were not considered. (see Figure 7.3)[2].

In this way, we visualize how pairs of features with a certain behavior are correctly associated with early folding residues in the rules. For example, *backbone-1>=0.8* is associated with *helix-2*, *sheet0* and *sidechain0* in rules that detect a high number of early folding residues. The thickness of the relation in the figure represents how many true positives they detect together.

Figure 7.4 zooms in on the relations between *backbone0* and *backbone-1* with *helix-2* or *sheet0*, as well as *helix-2* with *sheet0*. From Figure 7.4 (a) we can observe that values of *backbone0* and *backbone-1* in the interval [0.8, 1.0] combined with values of *helix-2* in the interval [0.3,0.6] are associated with the majority of early folding residues for this two features. In the same way, Figure 7.4 (b) shows that values of *backbone0* and *backbone-1* between [0.7,1.0] and values of *sheet0* in [0.2,0.6] are associated with early folding residues. In addition, residues with *helix-2* in [0.3,0.6] and *sheet0* in [0.2,0.4], are also identified as early folding.

The use of this type of visualization is supported by the finding that, sometimes, humans prefer to see one or two possible causes of the outcome than the entire set of possible ones [110]. However, the limitation of showing only pairs of rules' conditions is a simplification that ignores the role of other conditions acting together. Although

[2]An animated version of this figure is available at https://codepen.io/igraugar/full/JjGmWMO.

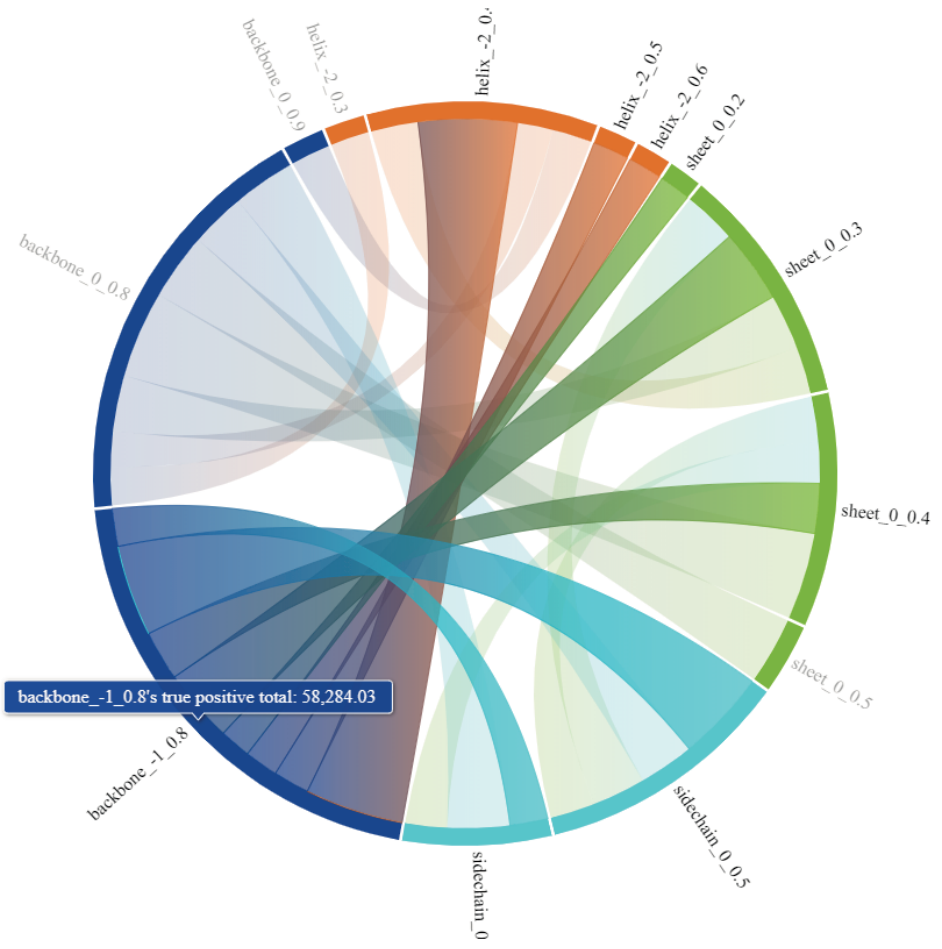


Figure 7.3: Pairwise combinations of attributes and cut-offs that lead to detecting early folding residues. Each color represents a group of features, e.g. dark blue for *backbone*. Each section in the chord diagram represents a condition in the rule, e.g. *backbone-1>=0.8* is highlighted. The thickness of the relations represents the total number of correct early folding residues detected by the pair of features connected, accumulated over the set of rules.

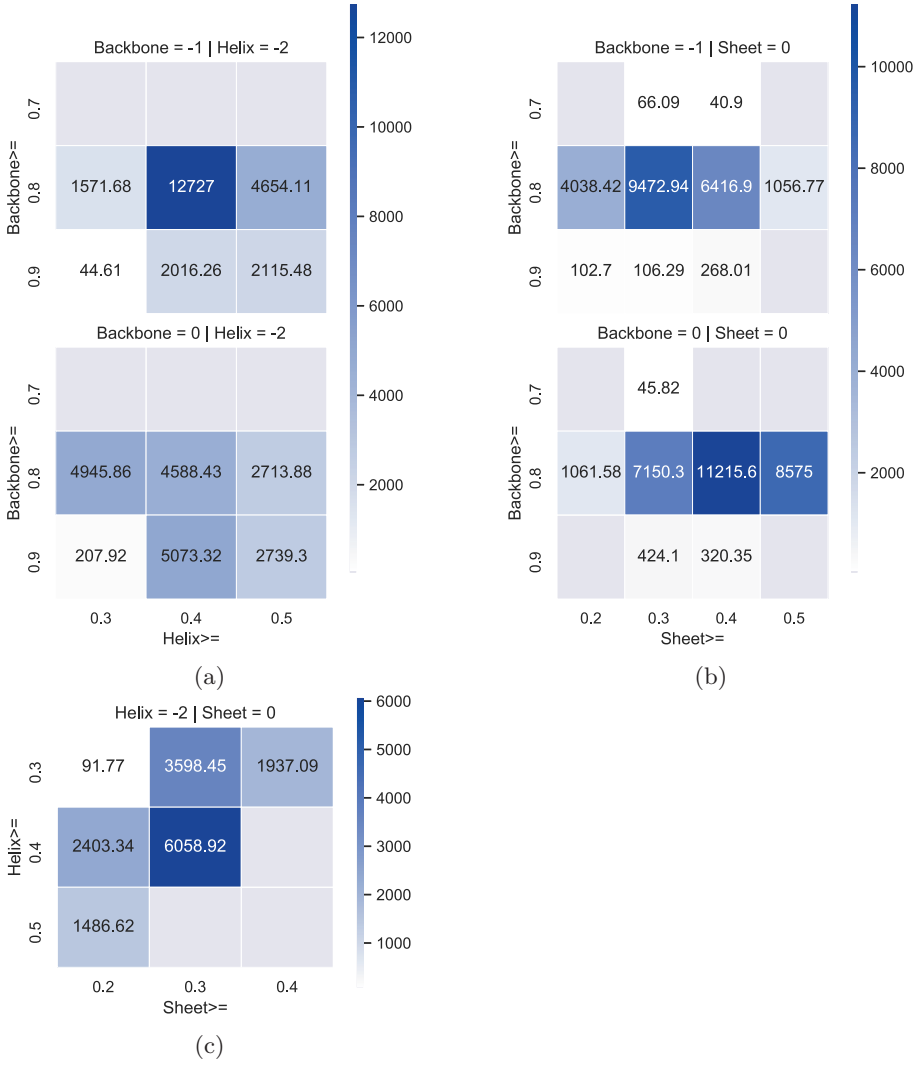


Figure 7.4: Pairwise combinations of attributes and cut-offs that lead to highest true positive counts: (a) *backbone0* and *backbone-1* paired with *helix-2*, (b) *backbone0* and *backbone-1* paired with *sheet0* and (c) *helix-2* paired with *sheet0*. Values in *backbone* greater or equal to 0.8 combined with *helix-2* or *sheet0* with values greater or equal to 0.4 are associated with the majority of early folding residues.

```
(backbone >= 0.873) and (sidechain >= 0.614) and (sheet >= 0.473)
     => Fold_Class=F (17081.55/48.89)
(backbone >= 0.87) and (sidechain >= 0.597) and (sheet >= 0.386) and
(helix-2 >= 0.44)
     => Fold_Class=F (6335.33/18.77)
(backbone-1 >= 0.873) and (sheet >= 0.232) and (helix-2 >= 0.619)
     => Fold_Class=F (5803.80/17.47)
(backbone-1 >= 0.911) and (sheet >= 0.292) and (helix-2 >= 0.513)
     => Fold_Class=F (3030.10/23.43)
(backbone >= 0.855) and (sheet >= 0.523) and (helix-2 >= 0.166)
     => Fold_Class=F (3210.38/8.47)
(backbone-1 >= 0.908) and (helix-2 >= 0.396) and (sidechain >= 0.623)
and (sidechain-1 >= 0.658)
     => Fold_Class=F (767.48/2.91)
(backbone >= 0.868) and (backbone-1 >= 0.908) and (helix-2 >= 0.633)
     => Fold_Class=F (1017.05/8.15)
```

(a) SlGb using RIP.

```
backbone<=0.889 AND sidechain>0.552 AND sheet>0.521
     : F (6104.15/9.17)
backbone>0.891 AND sidechain-1>0.401 AND sidechain>0.618 AND sheet>0.524
     : F (9804.29/11.65)
backbone>0.896 AND sidechain>0.417 AND helix-2>0.46 AND sheet>0.337
     : F (11096.67/37.57)
backbone>0.914 AND helix-2>0.379 AND sidechain>0.6 AND sidechain-1>0.608
     : F (3934.63/33.9)
sheet>0.449 AND backbone-1>0.841 AND backbone-2>0.829 AND helix-2>0.311
AND backbone<=0.926 AND sidechain>0.416
     : F (2897.97/4.82)
backbone>0.924 AND helix-2>0.512
     : F (2973.82/52.91)
sheet>0.483 AND backbone>0.852 AND sidechain>0.388 AND helix-2>0.16
     : F (2913.31/30.01)
```

(b) SlGb using PART.

Figure 7.5: First seven rules discovered by SlGb using (a) RIP and (b) PART (first seven that predict folding as positive). The paired conditions identified as relevant by the SlGb using RIP are also obtained in the decision list generated by SlGb using PART.

these visualizations were found interesting and understandable by the expert, a diagram that shows all conditions interacting together towards detecting early folding would provide a more detailed explanation.

Finally, we extract rules of interest from the two SlGb models. Figure 7.5 (a) shows the first seven rules obtained by SlGb using RIP as the white box. These rules are the most confident ones predicting early folding class. Figure 7.5 (b) shows the first seven rules for predicting early folding residues, obtained by SlGb using PART as the white box. We can observe in the antecedents that, the features appearing as relevant in Figure 7.3, are also obtained in the model built by SlGb using PART. This means that SlGb is able to find consistent patterns across different choices of white boxes. Further analysis of these patterns by associating the feature values with biophysical domain knowledge is needed for unveiling insights on the early folding regions of the proteins.

7.3 Concluding Note

In this chapter, we explored the use of SlGb in the prediction of early folding residues in proteins. We show that SlGb is able to leverage extra unlabeled data available for obtaining accurate prediction models that can be interpreted. We obtain the most accurate and interpretable results when using RIP as a white-box component. While SlGb using PART also achieves competitive performance and has low relative growth in structure compared to its base white box. The decision list generated by RIP is further interpreted by analyzing the pairs of features and their range of values that appear together in the most confident rules. We created visualizations of these paired interactions, which were found interesting and understandable by the expert. This analysis shows how high values of backbone features interact with other features in rules that detect early folding residues. Additionally, based on the interacting features, we show that SlGb is able to find consistent patterns across different choices of white boxes. The discussion of the results with an expert revealed that some rules obtained by the SlGb using decision lists were expected patterns matching the knowledge from the biophysical domain, while others were meaningful new patterns that should be further validated.

8 | Conclusions

This chapter outlines the main contributions of our research and the future research directions that were identified.

8.1 Contributions and Discussion

In multiple application domains for machine learning methods, annotating data according to a target characteristic when putting together a dataset for supervised classification is a costly or time-consuming process. This problem leads to datasets with a limited number of labeled data, while a higher number of unlabeled data is available. Semi-supervised classification (SSC) attempts to leverage both sources of data for learning models that reflect better the distribution of the data. Although the state-of-the-art is extensive, these methods are mostly complex ensembles or deep learning techniques that behave like black boxes. However, the interpretability requirement is an increasing need for the appropriate and responsible use of machine learning techniques in numerous applications.

In this research, we have proposed an interpretable semi-supervised classifier (called SlGb), aiming to find a balance between accuracy and interpretability. We build upon a self-labeling strategy for SSC, i.e., relying on the predictions of a base classifier for assigning labels to unlabeled data. An accurate black-box classifier is taken as the self-labeling component, but instead of re-training this base classifier in the enlarged dataset, we use a second white-box classifier. The white-box component can be any intrinsically interpretable classifier that acts as a surrogate model by mimicking the predictions of the black-box. The role of the black box is to ensure the accuracy of

the predictions of the unlabeled instances, while the white box builds an interpretable model that reflects the predictions made on the enlarged dataset.

As the self-labeling process's predictions on the unlabeled data might be incorrect, a mechanism for controlling the propagation of errors is needed. With this aim, we proposed two amending procedures that weight the instances in the enlarged dataset after the self-labeling. A first weighting strategy uses the class membership probability that the black box produces together with its predictions. This weighting guides the white-box component to learn from the most confident instances after the self-labeling. However, there could be inconsistency in the class labels that emerge when conforming the enlarged dataset. Therefore we propose a second amending procedure that computes the certainty of the labels of all instances, the originally labeled ones together with the self-labeled ones. For this purpose, we compute RST measures that use a similarity relation and the granulation of the decision space in regions of instances that are certainly from a class, certainly not from a class, or have boundary behavior. The result is an estimation of how confident is the classification of the instance, based not only on the positive evidence of the class but also on the negative and the doubtful information about the instance and its similar ones.

We evaluated our proposal through experiments conducted on a standard and comprehensive collection of structured benchmark datasets for SSC. We first evaluate three mainstream classifiers for their role as a black box: random forest, support vector machines, and multilayer perceptron neural networks. Random forests showed to be the best choice across the datasets in terms of prediction accuracy. Secondly, we evaluate the influence of using decision trees or two different decision lists as white boxes, combined with the two proposed amending strategies. The choice of a particular white box or amending strategy does not significantly affect the prediction rates, but it is rather relevant for the interpretability of the model.

We measured interpretability based on the size of the grey box structure, with the number of rules in the decision tree or the decision lists as an indicator of transparency. We evaluated three measures related to interpretability: the growth in structure compared to the base white box, the simplicity of the set of rules, and the balance between accuracy and interpretability in the form of a utility function. When using decision lists as surrogate white boxes, the grey box produces fewer rules than the C4.5 decision tree, even when no amending is used. However, the amending helps decrease the size of the structure without affecting the prediction rates by guiding the white box to learn from the most confident instances. Notably, the RST-based amending prevents the white box from focusing on learning inconsistencies in the labels that emerge from the enlarged dataset. Therefore, as a general configuration of SlGb, we advise using random forests as a base black box and amending the self-labeling with the RST-based strategy. The choice of white box is more flexible to the desired interpretability, either a decision tree with a set of rules or a decision list with ordered rules. The exper-

imental comparison against the state-of-the-art self-labeling approaches shows that this configuration outperforms four self-labeling methods, yet being far more simple in structure than these techniques.

As a final contribution, we illustrate the use of SlGb for two case studies from the field of bioinformatics. The first application involves the prediction of disease-causing genomic variants of a rare disease. In this case study, labeling genomic variants according to their pathogenicity was a manual process based on literature guidelines. In an attempt to generate more labeled data, we propose a tool named GeVaCT implementing these guidelines and automating the process as much as possible. However, some steps related to information about the family or the literature need the input from the expert and cannot be automatized, therefore GeVaCT is not a solution for classifying all data. This means that the limitations in the labeling process are not completely removed and SlGb is a valuable approach. We showed that SlGb is a good predictor of the pathogenicity of BrS variants compared to its white box baselines. It not only produces better results in terms of several measures of accuracy but also maintained or improved the interpretability when compared with the white-box baseline. When analyzing the rules generated by SlGb, some patterns related to the GeVaCT pipeline were observed. Further experiments and feedback from experts are needed to adjust and validate the rules obtained.

The second application concerns the prediction of early folding residues in proteins. In this domain, labeled data is limited due to it comes from controlled experiments. SlGb is able to leverage extra unlabeled data from other proteins to obtain accurate decision lists that can also be interpreted. We obtain the most accurate and interpretable results when using RIP as a white-box component. Since RIP focuses on finding rules for the minority class, it is more suitable for finding patterns in the features that lead to early folding. The decision list generated by RIP is further interpreted by analyzing pairs of features and their range of values that appear together in the most confident rules. This analysis shows how high values of backbone features interact with other features in rules that detect early folding residues. Finally, based on the interacting features, we show that SlGb is able to find consistent rules across different choices of white boxes. The discussion of the results with an expert revealed that some rules obtained by the SlGb using decision lists were expected patterns, while others were meaningful and interesting new patterns that should be further validated. Further interpretation of these patterns by associating the feature values with domain knowledge is needed for unveiling insights in the early folding regions of the proteins.

8.2 Future Research Lines

From the research performed in this thesis, we have detected several ideas that are worth exploring in the near future.

The extension of the proposed self-labeling grey-box to unstructured SSC problems is an appealing idea. However, mapping unstructured data (e.g., images, text) to interpretable features that can be later used for building a white-box model is a challenge. In this direction, there has been some progress in recent years using variational autoencoders for obtaining disentangled representations. The first contribution to theory and metrics for learning disentangled representations taking into account interpretability was recently published in [37]. Including concepts defined by humans as constraints for the disentangled representation learning in an SSC setting is an attractive research direction.

It would also be interesting to explore other changes in the representation space before the white box training, e.g., the influence of adding more interpretable features before training the white box. It would be important to determine whether incorporating more attributes from the same or different semantic dimensions helps the white box obtain better performance or interpretability. Another alternative would be further preprocessing of the original features with a technique such as discretization. Discretization methods come with the advantage of reducing and simplifying the data, potentially making the white box learn more compact sets of rules. Generally, discrete attributes are considered easier to understand [102]. However, any discretization method also comes with a certain loss of information that should be considered.

Regarding the amending procedures, a natural extension of the proposed RST-based confidence measure is to partition the decision space with an extension of RST, such as Fuzzy RST [29], for example. This extension would model every instance as a member of all regions for each class with a certain degree, removing the need for a similarity threshold parameter. The bias present in the SlGb pipeline through the amending affects the interpretability since a strong preference for confident instances would make the white box focus on discovering safe but rather apparent patterns. The more uncertain instances are those more difficult to classify and generally lie near classification boundaries. These instances can add novelty, although they are risky due to their explanations have lower confidence. A softer approach to the amending can potentially reduce the bias towards very confident but rather obvious patterns by allowing instances to be confident and doubtful with different degrees. It would be interesting to evaluate the influence of using Fuzzy RST amending in the accuracy and interpretability of the SlGb.

From our literature review on machine learning interpretability, we can conclude that more work is needed in conceiving measures for the quality of explanations. An exciting idea derived from the proposed measures is to model the accuracy and inter-

pretability trade-off as a multi-objective optimization problem. Considering several measures of accuracy and interpretability, together with weights specified by a panel of experts, can be further explored.

Concerning the application of our pipeline in real case studies, it would be attractive to explore its results by contrasting them with traditional statistical analyses of the data or with established expert systems. We can separate the SlGb explanations related to domain rules from the new patterns that need to be studied. Known patterns are useful to validate the confidence of the predictions and explanations of the SlGb, while new patterns increase novelty by explaining instances that are not covered by the traditional analysis. Those new patterns that the expert considers plausible hypotheses can be further confirmed or rejected by traditional techniques using more data collected for this purpose. The novel patterns that the experts discard could be used as feedback and integrated as constraints when retraining our model. One step forward would be to integrate our model with second-generation expert systems, where the reasoning through rules have a more hierarchical organization with intermediate goals. More advanced rule mining algorithms that combine data with background knowledge would be needed for the white-box component.

In both of the case studies presented in this thesis, further validation of the obtained explanations with the help of experts is necessary. The guidelines published in [73] revising questionnaires and interview methods for the context of XAI are a good starting point for this aim. These questionnaires aim to measure the extent to which the user's mental model about the domain is refined after obtaining explanations. The interaction with the user helps assess the goodness of explanations in several aspects such as satisfaction, understanding, curiosity, and trust. Therefore, the validation of the explanations is fundamental for an appropriate deployment of an XAI tool based on our pipeline. Additionally, we want to explore the formalization of the expert's feedback in knowledge representations such as ontologies or knowledge graphs. The ontologies could be used for re-adjusting the models to the users' preferences or expectations. In the same direction, providing an interface using natural language, such as conversational agents (an idea we are currently exploring in another project), is an exciting way of incorporating the human in the loop.

A | Description of the Benchmark Datasets

Table A.1: Characterization of the datasets used in experiments in Chapter 5. The imbalance is computed as the ratio of the number of instances between the majority and the minority class of the dataset, NA means a ratio smaller than two.

Dataset	Instances	10%	20%	30%	40%	Features	Classes	Imbalance
abalone	4,174	417	835	1,252	1,670	8	28	689.00
appendicitis	106	11	21	32	42	7	2	4.04
australian	690	69	138	207	276	14	2	NA
autos	205	21	41	62	82	25	6	22.33
banana	5,300	530	1,060	1,590	2,120	2	2	NA
breast	286	29	57	86	114	9	2	2.36
bupa	345	35	69	104	138	6	2	NA
chess	3,196	320	639	959	1,278	36	2	NA
cleveland	297	30	59	89	119	13	5	12.61
coil2000	9,822	982	1,964	2,947	3,929	85	2	15.76
contraceptive	1,473	147	295	442	589	9	3	NA
crx	125	13	25	38	50	15	2	NA
dermatology	366	37	73	110	146	33	6	5.6
ecoli	336	34	67	101	134	7	8	71.5
flare-solar	1,066	107	213	320	426	9	6	7.69
german	1,000	100	200	300	400	20	2	2.33
glass	214	21	43	64	86	9	7	8.44
haberman	306	31	61	92	122	3	2	2.77
heart	270	27	54	81	108	13	2	NA
hepatitis	155	16	31	47	62	19	2	3.84
housevotes	435	44	87	131	174	16	2	NA
iris	150	15	30	45	60	4	3	NA
led7digit	500	50	100	150	200	7	10	NA
lymphography	148	15	30	44	59	18	4	40.5
magic	19,020	1,902	3,804	5,706	7,608	10	2	NA

Continued on next page

Table A.1 – *Continued from previous page*

Dataset	Instances	10%	20%	30%	40%	Features	Classes	Imbalance
mammographic	961	96	192	288	384	5	2	NA
marketing	8,993	899	1,799	2,698	3,597	13	9	2.48
monks	432	43	86	130	173	6	2	NA
movement_libras	360	36	72	108	144	90	15	NA
mushroom	8,124	812	1,625	2,437	3,250	22	2	NA
nursery	12,690	1,269	2,538	3,807	5,076	8	5	2,160.00
pageblocks	5,472	547	1,094	1,642	2,189	10	5	175.46
penbased	10,992	1,099	2,198	3,298	4,397	16	10	NA
phoneme	5,404	540	1,081	1,621	2,162	5	2	2.4
pima	768	77	154	230	307	8	2	NA
ring	7,400	740	1,480	2,220	2,960	20	2	NA
saheart	462	46	92	139	185	9	2	NA
satimage	6,435	644	1,287	1,931	2,574	36	7	2.44
segment	2,310	231	462	693	924	19	7	NA
sonar	208	21	42	62	83	60	2	NA
spambase	4,597	460	919	1,379	1,839	55	2	NA
spectheart	267	27	53	80	107	44	2	3.85
splice	3,190	319	638	957	1,276	60	3	2.15
tae	151	15	30	45	60	5	3	NA
texture	5,500	550	1,100	1,650	2,200	40	11	NA
tic-tac-toe	958	96	192	287	383	9	2	NA
thyroid	7,200	720	1,440	2,160	2,880	21	3	40.15
titanic	2,201	220	440	660	880	3	2	2.09
twonorm	7,400	740	1,480	2,220	2,960	20	2	NA
vehicle	846	85	169	254	338	18	4	NA
vowel	990	99	198	297	396	13	11	NA
wine	178	18	36	53	71	13	3	NA
wisconsin	683	68	137	205	273	9	2	NA

Continued on next page

Table A.1 – *Continued from previous page*

Dataset	Instances	10%	20%	30%	40%	Features	Classes	Imbalance
yeast	1,484	148	297	445	594	8	10	92.6
zoo	101	10	20	30	40	17	7	10.25

B | Detailed Results of the Statistical Tests

Table B.1: Friedman's p-values for all ratios when testing different black-box base classifiers. The prediction rates are measured using kappa coefficient. There are significant differences among all the configurations compared.

Ratio	Friedman's p-value	H_0
10%	2.10E-08	Rejected
20%	8.91E-13	Rejected
30%	9.87E-06	Rejected
40%	5.35E-04	Rejected

Table B.2: Wilcoxon's p-values and Holm's post-hoc correction when comparing different black-boxes configurations. The test supports the superiority of RF as black-box base classifier when comparing prediction rates.

Labeled ratio	Pair of configurations	Wilcoxon's p-value	R^-	R^+	Holm's p-value	H_0
10%	RF-PART - MLP-PART	2.08E-05	13	39	1.25E-04	Rejected
	RF-PART - SVM-PART	2.77E-04	16	39	1.38E-03	Rejected
	RF-C45 - SVM-C45	6.91E-04	16	39	2.76E-03	Rejected
	RF-RIP - MLP-RIP	9.63E-04	15	40	2.89E-03	Rejected
	RF-C45 - MLP-C45	1.6E-03	15	39	3.2E-03	Rejected
	RF-RIP - SVM-RIP	5.84E-03	19	36	5.84E-03	Rejected
20%	RF-C45 - MLP-C45	5.58E-06	9	45	3.35E-05	Rejected
	RF-PART - SVM-PART	2.1E-05	15	39	1.05E-04	Rejected
	RF-PART - MLP-PART	4.56E-05	13	41	1.83E-04	Rejected
	RF-RIP - SVM-RIP	1.83E-04	17	37	5.5E-04	Rejected
	RF-C45 - SVM-C45	3.04E-04	15	39	6.08E-04	Rejected
	RF-RIP - MLP-RIP	3.56E-03	18	36	3.56E-03	Rejected
30%	RF-RIP - MLP-RIP	1.18E-03	15	38	7.06E-03	Rejected
	RF-C45 - MLP-C45	1.19E-03	16	38	7.06E-03	Rejected
	RF-C45 - SVM-C45	1.75E-03	16	38	7.06E-03	Rejected
	RF-RIP - SVM-RIP	2.93E-03	18	36	8.79E-03	Rejected
	RF-PART - MLP-PART	5.64E-03	20	34	1.13E-02	Rejected
	RF-PART - SVM-PART	7.7E-03	20	34	1.13E-02	Rejected
40%	RF-RIP - SVM-RIP	1.55E-03	16	38	9.33E-03	Rejected
	RF-PART - MLP-PART	6.26E-03	19	35	3.13E-02	Rejected
	RF-C45 - MLP-C45	8.75E-03	20	34	3.5E-02	Rejected
	RF-PART - SVM-PART	1.43E-02	19	35	4.29E-02	Rejected
	RF-C45 - SVM-C45	2.28E-02	22	32	4.55E-02	Rejected
	RF-RIP - MLP-RIP	7.68E-02	23	31	7.68E-02	Not Rejected

Table B.3: Friedman's p-values for all ratios when testing the prediction performance (kappa) for different white-box and amending configurations. There are statistical differences in the prediction rates in at least one pair of the configurations compared.

Ratio	Friedman's p-value	H_0
10%	4.02E-07	Rejected
20%	9.27E-07	Rejected
30%	1.67E-03	Rejected
40%	7.35E-05	Rejected

Table B.4: Wilcoxon's p-values and Holm's post-hoc correction when comparing different white-box and amending configurations, for 10% and 20% ratio. Per ratio, first subsection compares using different amending procedures while fixing the white box and the second subsection fixes the amending for comparing the influence of white boxes. The vast majority of null hypothesis cannot be rejected, indicating that amending or white-box alternatives do not strongly influence the prediction rates.

Labeled ratio	Pair of configurations	Wilcoxon's p-value	R^-	R^+	Holm's p-value	H_0
10%	RF-RIP-RST - RF-RIP-NONE	2.57E-03	40	12	1.54E-02	Rejected
	RF-C45-RST - RF-C45-NONE	3.4E-02	36	17	0.170	Not Rejected
	RF-RIP-CONF - RF-RIP-NONE	4.23E-02	33	19	0.170	Not Rejected
	RF-PART-RST - RF-PART-NONE	5.24E-02	31	21	0.170	Not Rejected
	RF-C45-CONF - RF-C45-NONE	0.224	30	23	0.447	Not Rejected
	RF-PART-CONF - RF-PART-NONE	0.344	27	25	0.447	Not Rejected
	RF-RIP-CONF - RF-PART-CONF	7.09E-04	35	18	6.38E-03	Rejected
	RF-RIP-RST - RF-PART-RST	1.44E-02	35	18	0.115	Not Rejected
	RF-RIP-NONE - RF-PART-NONE	1.97E-02	34	19	0.138	Not Rejected
	RF-RIP-RST - RF-C45-RST	2.01E-02	33	20	0.138	Not Rejected
	RF-RIP-CONF - RF-C45-CONF	3.25E-02	33	20	0.163	Not Rejected
	RF-PART-CONF - RF-C45-CONF	9.52E-02	18	33	0.381	Not Rejected
	RF-PART-NONE - RF-C45-NONE	0.166	23	29	0.499	Not Rejected
	RF-PART-RST - RF-C45-RST	0.188	20	30	0.499	Not Rejected
	RF-RIP-NONE - RF-C45-NONE	0.510	31	23	0.510	Not Rejected
20%	RF-RIP-RST - RF-RIP-NONE	8.04E-05	38	14	4.82E-04	Rejected
	RF-C45-CONF - RF-C45-NONE	4.89E-03	36	14	2.45E-02	Rejected
	RF-PART-RST - RF-PART-NONE	4.92E-02	33	19	0.197	Not Rejected
	RF-PART-CONF - RF-PART-NONE	5.23E-02	33	18	0.197	Not Rejected
	RF-C45-RST - RF-C45-NONE	5.82E-02	34	18	0.197	Not Rejected
	RF-RIP-CONF - RF-RIP-NONE	0.169	31	21	0.197	Not Rejected
	RF-RIP-RST - RF-PART-RST	1.6E-03	39	14	1.44E-02	Rejected
	RF-RIP-RST - RF-C45-RST	6.12E-03	36	16	4.9E-02	Rejected
	RF-RIP-NONE - RF-PART-NONE	2.78E-02	36	17	0.195	Not Rejected
	RF-RIP-NONE - RF-C45-NONE	4.53E-02	36	18	0.272	Not Rejected
	RF-RIP-CONF - RF-C45-CONF	0.169	30	22	0.854	Not Rejected
	RF-RIP-CONF - RF-PART-CONF	0.259	28	25	1.000	Not Rejected
	RF-PART-RST - RF-C45-RST	0.769	26	23	1.000	Not Rejected
	RF-PART-NONE - RF-C45-NONE	0.827	27	25	1.000	Not Rejected
	RF-PART-CONF - RF-C45-CONF	0.889	27	23	1.000	Not Rejected

Table B.5: Wilcoxon's p-values and Holm's post-hoc correction when comparing different white-box and amending configurations, for 30% and 40% ratio. Per ratio, first subsection compares using different amending procedures while fixing the white box and the second subsection fixes the amending for comparing the influence of white boxes. The vast majority of null hypothesis cannot be rejected, indicating that amending or white-box alternatives do not strongly influence the prediction rates.

Labeled ratio	Pair of configurations	Wilcoxon's p-value	R^-	R^+	Holm's p-value	H_0
30%	RF-RIP-RST - RF-RIP-NONE	2.18E-04	37	15	1.31E-03	Rejected
	RF-C45-RST - RF-C45-NONE	8.45E-03	37	16	4.22E-02	Rejected
	RF-RIP-CONF - RF-RIP-NONE	4.14E-02	32	20	0.165	Not Rejected
	RF-C45-CONF - RF-C45-NONE	0.193	28	24	0.578	Not Rejected
	RF-PART-CONF - RF-PART-NONE	0.362	28	24	0.725	Not Rejected
	RF-PART-RST - RF-PART-NONE	0.388	28	25	0.725	Not Rejected
	RF-RIP-RST - RF-PART-RST	1.68E-03	37	15	1.51E-02	Rejected
	RF-RIP-RST - RF-C45-RST	7.03E-03	35	17	5.62E-02	Not Rejected
	RF-RIP-CONF - RF-C45-CONF	0.147	30	24	1.000	Not Rejected
	RF-PART-RST - RF-C45-RST	0.259	22	27	1.000	Not Rejected
	RF-RIP-CONF - RF-PART-CONF	0.324	30	24	1.000	Not Rejected
	RF-RIP-NONE - RF-PART-NONE	0.355	31	22	1.000	Not Rejected
	RF-RIP-NONE - RF-C45-NONE	0.498	28	25	1.000	Not Rejected
	RF-PART-NONE - RF-C45-NONE	0.555	22	29	1.000	Not Rejected
	RF-PART-CONF - RF-C45-CONF	0.573	23	25	1.000	Not Rejected
40%	RF-RIP-RST - RF-RIP-NONE	1.35E-05	40	13	8.13E-05	Rejected
	RF-RIP-CONF - RF-RIP-NONE	7.03E-03	31	21	3.51E-02	Rejected
	RF-PART-RST - RF-PART-NONE	0.136	30	23	0.543	Not Rejected
	RF-PART-CONF - RF-PART-NONE	0.579	28	24	1.000	Not Rejected
	RF-C45-RST - RF-C45-NONE	0.662	26	26	1.000	Not Rejected
	RF-C45-CONF - RF-C45-NONE	0.761	26	24	1.000	Not Rejected
	RF-RIP-RST - RF-C45-RST	2.9E-03	39	13	2.61E-02	Rejected
	RF-RIP-CONF - RF-PART-CONF	2.13E-02	36	18	0.170	Not Rejected
	RF-RIP-CONF - RF-C45-CONF	3.25E-02	33	20	0.228	Not Rejected
	RF-RIP-RST - RF-PART-RST	3.96E-02	33	20	0.237	Not Rejected
	RF-RIP-NONE - RF-PART-NONE	0.254	33	21	1.000	Not Rejected
	RF-PART-NONE - RF-C45-NONE	0.267	23	30	1.000	Not Rejected
	RF-PART-CONF - RF-C45-CONF	0.606	26	24	1.000	Not Rejected
	RF-RIP-NONE - RF-C45-NONE	0.806	29	23	1.000	Not Rejected
	RF-PART-RST - RF-C45-RST	0.858	26	24	1.000	Not Rejected

Table B.6: Friedman's p-values for all ratios when comparing the interpretability in terms of simplicity, for different white-box and amending configurations. There are significant differences among all the configurations compared, where RF-RIP-RST exhibits the highest mean for all ratios (see Table 5.3).

Ratio	Friedman's p-value	H_0
10%	3.14E-73	Rejected
20%	3.02E-75	Rejected
30%	3.96E-72	Rejected
40%	2.41E-71	Rejected

Table B.7: Wilcoxon's p-values and Holm's post-hoc correction when comparing different white-box and amending configurations against the highest mean simplicity combination: RF-RIP-RST. All null hypothesis can be safely rejected, showing statistically significant superiority in terms of simplicity.

Labeled ratio	Pair of configurations	Wilcoxon's p-value	R^-	R^+	Holm's p-value	H_0
10%	RF-RIP-RST - RF-C45-NONE	1.11E-10	0	55	8.86E-10	Rejected
	RF-RIP-RST - RF-PART-NONE	1.17E-10	1	54	8.86E-10	Rejected
	RF-RIP-RST - RF-C45-CONF	1.63E-10	0	54	9.75E-10	Rejected
	RF-RIP-RST - RF-C45-RST	1.63E-10	0	54	9.75E-10	Rejected
	RF-RIP-RST - RF-RIP-NONE	2.39E-10	0	53	9.75E-10	Rejected
	RF-RIP-RST - RF-PART-CONF	3.97E-10	2	52	1.19E-09	Rejected
	RF-RIP-RST - RF-PART-RST	4.08E-09	3	51	8.17E-09	Rejected
	RF-RIP-RST - RF-RIP-CONF	2.18E-07	5	46	2.18E-07	Rejected
20%	RF-RIP-RST - RF-C45-NONE	1.11E-10	0	55	8.86E-10	Rejected
	RF-RIP-RST - RF-PART-NONE	1.11E-10	0	55	8.86E-10	Rejected
	RF-RIP-RST - RF-C45-CONF	1.11E-10	0	55	8.86E-10	Rejected
	RF-RIP-RST - RF-PART-CONF	1.11E-10	0	55	8.86E-10	Rejected
	RF-RIP-RST - RF-C45-RST	1.63E-10	0	54	8.86E-10	Rejected
	RF-RIP-RST - RF-RIP-NONE	2.39E-10	0	53	8.86E-10	Rejected
	RF-RIP-RST - RF-RIP-CONF	8.76E-10	1	52	1.75E-09	Rejected
	RF-RIP-RST - RF-PART-RST	3.95E-08	3	50	3.95E-08	Rejected
30%	RF-RIP-RST - RF-C45-NONE	1.11E-10	0	55	8.86E-10	Rejected
	RF-RIP-RST - RF-C45-CONF	1.11E-10	0	55	8.86E-10	Rejected
	RF-RIP-RST - RF-PART-NONE	1.24E-10	1	54	8.86E-10	Rejected
	RF-RIP-RST - RF-C45-RST	1.24E-10	1	54	8.86E-10	Rejected
	RF-RIP-RST - RF-PART-CONF	1.31E-10	2	53	8.86E-10	Rejected
	RF-RIP-RST - RF-RIP-NONE	2.6E-10	1	52	8.86E-10	Rejected
	RF-RIP-RST - RF-RIP-CONF	8.03E-10	1	49	1.61E-09	Rejected
	RF-RIP-RST - RF-PART-RST	1.18E-09	3	51	1.61E-09	Rejected
40%	RF-RIP-RST - RF-C45-NONE	1.11E-10	0	55	8.86E-10	Rejected
	RF-RIP-RST - RF-C45-CONF	1.11E-10	0	55	8.86E-10	Rejected
	RF-RIP-RST - RF-PART-NONE	1.24E-10	1	54	8.86E-10	Rejected
	RF-RIP-RST - RF-PART-CONF	1.31E-10	1	54	8.86E-10	Rejected
	RF-RIP-RST - RF-C45-RST	1.63E-10	0	54	8.86E-10	Rejected
	RF-RIP-RST - RF-RIP-NONE	2.68E-10	1	52	8.86E-10	Rejected
	RF-RIP-RST - RF-RIP-CONF	3.18E-10	2	51	8.86E-10	Rejected
	RF-RIP-RST - RF-PART-RST	2.7E-09	4	51	2.7E-09	Rejected

Table B.8: Friedman's p-values for all ratios when comparing SlGb (RF-PART-RST) with state-of-the-art semi-supervised classifiers in terms of prediction rates (kappa). There are significant differences for all ratios, where SlGb exhibits the highest mean (see Table 5.4).

Ratio	Friedman's p-value	H_0
10%	1.91E-06	Rejected
20%	7.62E-07	Rejected
30%	3.50E-06	Rejected
40%	2.19E-03	Rejected

Table B.9: Wilcoxon's p-values and Holm's post-hoc correction using SlGb approach as control method against state-of-the-art semi-supervised classifiers. SlGb significantly outperforms other methods except for CT(SMO) and DCT when using 30% and 40% of labeled instances.

Labeled ratio	SSC algorithm	Wilcoxon's p-value	R^-	R^+	Holm's p-value	H_0
10%	CB(C45)	3.6E-06	12	43	1.44E-05	Rejected
	TT(C45)	4.23E-06	13	42	1.44E-05	Rejected
	DCT	1.86E-05	12	43	3.71E-05	Rejected
	CT(SMO)	1.74E-04	16	39	1.74E-04	Rejected
20%	CB(C45)	1.3E-07	9	46	5.21E-07	Rejected
	TT(C45)	8.37E-07	9	46	2.51E-06	Rejected
	DCT	2.77E-04	15	40	5.53E-04	Rejected
	CT(SMO)	2.35E-03	19	36	2.35E-03	Rejected
30%	CB(C45)	1.64E-07	9	46	6.54E-07	Rejected
	TT(C45)	4.84E-07	9	45	1.45E-06	Rejected
	DCT	7.16E-03	19	36	1.43E-02	Rejected
	CT(SMO)	5.4E-02	20	35	5.4E-02	Not Rejected
40%	TT(C45)	6.58E-05	12	42	2.63E-04	Rejected
	CB(C45)	8.18E-05	15	39	2.63E-04	Rejected
	DCT	0.172	24	31	0.344	Not Rejected
	CT(SMO)	0.633	27	28	0.633	Not Rejected

C | GeVaCT: Genomic Variant Classifier Tool

This appendix describes some implementation details of the GeVaCT software. GeVaCT is a semi-automated knowledge engineer software for the pathogenicity classification of BrS (see Section 6.2). It is based on a published study [74] and experience of clinical geneticist of the Center for Medical Genetics of the Universitair Ziekenhuis Brussel (UZBrussel) with whom we closely collaborated. GeVaCT can be executed through a graphical user interface or a console interface.

GeVaCT preprocess and labels BrS variants according to their pathogenicity in five classes:

- Class I: Non-pathogenic
- Class II: Unlikely pathogenic
- Class III: Unclear
- Class IV: Likely pathogenic
- Class V: Pathogenic

Input:

- Tab delimited annotated VCF file generated as output from Alamut Batch software [77].
- Optionally: A text file with a list of genes that will be analyzed.

Preprocessing steps (see Figure C.1):

1. Filter the variants based on the gene list (if provided)
2. Refer to the attribute *hgmdSubCategory*. Retain records with values: *DM* and *DM?*.
3. Refer to the attribute *clinVarClinSignifis*. Retain records with values: *pathogenic*, *likely pathogenic*.
4. Refer to the attributes *comment_DD* and *class* (these attributes are added as internal process of UZBrussel). Retain records with values: *VUS2*, *VUS3* and *pathogenic*.
5. Refer to the attribute *varLocation*. If attribute *clinVarClinSignifis* has value *pathogenic* or *likely pathogenic* retain records with values: *exon*, *splice site*, *exception - intron*, *UTR* and *downstream*. Otherwise retain records with values:

exon and *splice site*. Look into attribute *codingEffect* and retain all records except for value *synonymous*.

6. Select the attribute *rsMAF* (dbSNP [105] Minor Allele Frequency) and select variants with $rsMAF < 0.1$ (10%).

7. Separate variants in two groups: *missense* and *nonsense or frameshift*.

Labeling steps for missense variants:

1. Refer to the attributes *hdiv_prediction* and *hvar_prediction* from Polyphen database, filtering using attributes: *chrom*, *pos*, *gene*, *wtNuc*, *varNuc*, *wtAA*, *varAA* and *transcript*. Accumulate score value 1.0 if *probably damaging*, 0.5 if *possibly damaging* and 0.0 if benign.

2. Refer to the attributes *SIFTprediction* (*deleterious* or *tolerated*) and *SIFTWeight* (values from 0 to 1). Accumulate score value 1.0 if *deleterious* and *SIFTWeight* is in the interval (0.0, 0.05), accumulate 0.0 otherwise.

3. Refer to the attribute *granthamDIST*. If the value is larger than 140, accumulate score 2.0, otherwise if larger than 70, accumulate 1.0, else accumulate 0.0.

4. Refer to the attribute *AGVGDclass*. For each possible value accumulate the corresponding score:

   - C65 most likely $\longrightarrow$ score 1.25
   - C55 $\longrightarrow$ score 1.0
   - C45 $\longrightarrow$ score 0.75
   - C35 $\longrightarrow$ score 0.5
   - C15/25 $\longrightarrow$ score 0.25
   - C0 $\longrightarrow$ score 0.0

5. Refer to the attribute *BLOSUM62* and accumulate score 1.0 if value is less or equal than -2, 0.5 if value is exactly -1, and 0.0 otherwise.

6. Refer to the attribute *PhyloP* and accumulate score 0.0 if value is less than 1, 0.5 if value is greater or equal than 1 and less than 2.5, and 1.0 otherwise.

7. Refer to the attribute *espEAMAF* or *espAAMAF*, depending on ethnical background of the patient. **The ethnical background is obtained from a manual input step.** For each possible value accumulate the corresponding score:

   - equal to 0 $\longrightarrow$ 2.0

- greater than 0 and less or equal to 0.002 $\longrightarrow$ 1.5
- greater than 0.002 and less or equal to 0.005 $\longrightarrow$ 1.0
- greater than 0.005 and less or equal to 0.01 $\longrightarrow$ 0.5
- greater than 0.01 $\longrightarrow$ 0.0

8. Refer to the attribute *localSpliceEffect*. If *localSpliceEffect* is not missing, for each possible value accumulate the corresponding score:

   - New Donor Site $\longrightarrow$ 1.0
   - Cryptic Donor Strongly Activated $\longrightarrow$ 1.0
   - New Acceptor Site $\longrightarrow$ 1.0
   - Cryptic Acceptor Strongly Activated $\longrightarrow$ 1.0
   - Cryptic Donor Weakly Activated $\longrightarrow$ 0.5
   - Cryptic Acceptor Weakly Activated $\longrightarrow$ 0.5

   When *localSpliceEffect* is missing, refer to the attributes *wtSSFScore*, *wtMaxEntScore*, *wtNNSScore*, *wtGSScore*, *wtHSFScore*, *varSSFScore*, *varMaxEntScore*, *varNNSScore*, *varGSScore* and *varHSFScore*. Compute the percentage relative differences between each pair of variant and wild type variables, for example:

   $d = abs(wtSSFScore - varSSFScore)/max(wtSSFScore, varSSFScore) * 100.$

   For each possible value accumulate the corresponding score:

   - if at least two differences have value less than 40% $\longrightarrow$ 2.0
   - else if at least one difference has value less than 40% $\longrightarrow$ 1.0
   - else if at least one difference has value less than 70% $\longrightarrow$ 0.5
   - else all differences are greater or equal to 70% $\longrightarrow$ 0.0

9. Compute the cumulative score and assign the *first label* according to:

   - if cumulative score is greater or equal than 70% $\longrightarrow$ Class IV
   - else if cumulative score is greater or equal than 45% $\longrightarrow$ Class III
   - else if cumulative score is greater or equal than 25% $\longrightarrow$ Class II
   - else cumulative score is lower than 25% $\longrightarrow$ Class I

Labeling steps for nonsense and frameshift variants:

1. Refer to the attribute *codingEffect*. If *nonsense* or *frameshift* add 4.0 to cumulative score. Otherwise, apply step 8 for missense variants.

2. **Ask user for manual input of whether the variant type fits with the disease.** The scores assigned in this step are based on knowledge from literature, adding a still manual and subjective component to the labeling (see Figure C.3 for question asked in the console version of GeVaCT).

3. Analyze frequency in control population. Refer to step 7 for missense variants. **Involves manual input of ethnical background.**

4. Compute the cumulative score and assign the *first label*. Refer to step 9 for missense variants.

Final steps:

1. **Ask user for manual input of *family information or phenotype* and *functional analysis*.** The scores assigned in this step are based on expert's criteria and experience, adding a still manual and subjective component to the labeling (see Figure C.2).

   For family information the scores are the following:

   - Very likely pathogenic: de novo mutation or âĽě6 affected family members with the mutation and no affected without the mutation $\longrightarrow$ 4.0
   - Probably co-segregation: 5 affected family members with the mutation and no affected without the mutation $\longrightarrow$ 3.0
   - Possible co-segregation: 3-4 affected family members with the mutation and no affected without the mutation $\longrightarrow$ 2.0
   - Co-segregation unclear: 2 affected family members with the mutation and no affected without the mutation $\longrightarrow$ 1.0
   - Only index $\longrightarrow$ 0.0
   - No co-segregation: Affected family member without mutation $\longrightarrow$ 0.0
   - No score $\longrightarrow$ 0.0

   For functional analysis the following question is asked to the user: "Is the variant functionally tested *in vitro*, in culture or in an animal model? If so judge based on the method used and the experimental data how convincing the conclusion is. This is important because functional assays are often not well validated."

- Convincingly functionally aberrant $\longrightarrow$ 3.0
- Possibly functionally aberrant $\longrightarrow$ 1.0
- Unclear or not functionally aberrant $\longrightarrow$ 0.0
- No score $\longrightarrow$ 0.0

2. Refer to cumulative score from step 9 for missense variants and step 4 for nonsense and frameshift. Compute final label based on scores from previous steps and cumulative score:

   - A combined score of 2.0 or 3.0 for *family information or phenotype* and *functional analysis*, upgrades the score from the *first label* one level.
   - A maximum score for *functional analysis* and a score 0.0 for *family information or phenotype*, upgrades Classes I to III to Class IV (likely pathogenic).
   - A combined score of 4.0 for *family information or phenotype* and *functional analysis* with none of the parts having a maximum score, upgrades the Classes I to III to Class IV (likely pathogenic).
   - A maximum score for *family information or phenotype*, upgrades all classes to Class V (pathogenic).
   - A combined score of 5.0 or 6.0 for *family information or phenotype* and *functional analysis* and *family information or phenotype* has not the maximum score, upgrade all classes to Class V (pathogenic).

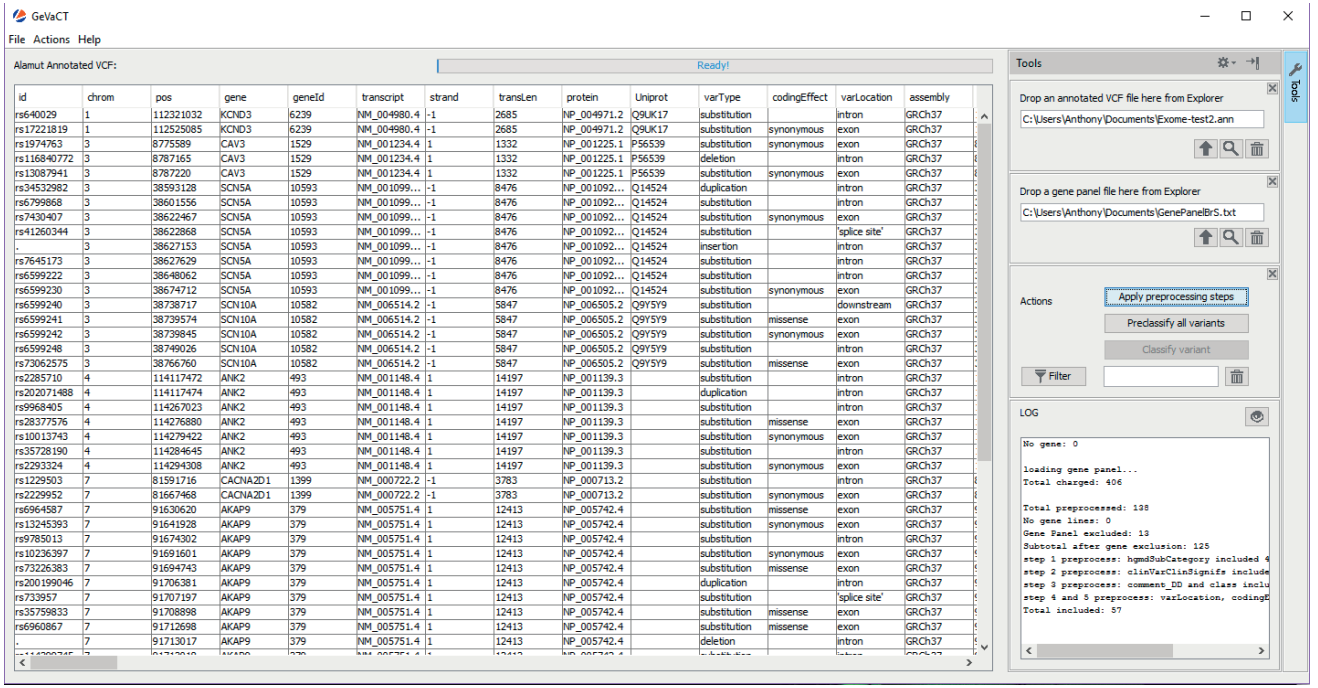


Figure C.1: Graphical user interface of GeVaCT showing one an example VCF file. The right panel has quick access to the most used functions. Additionally, it shows a log of all scores assigned.

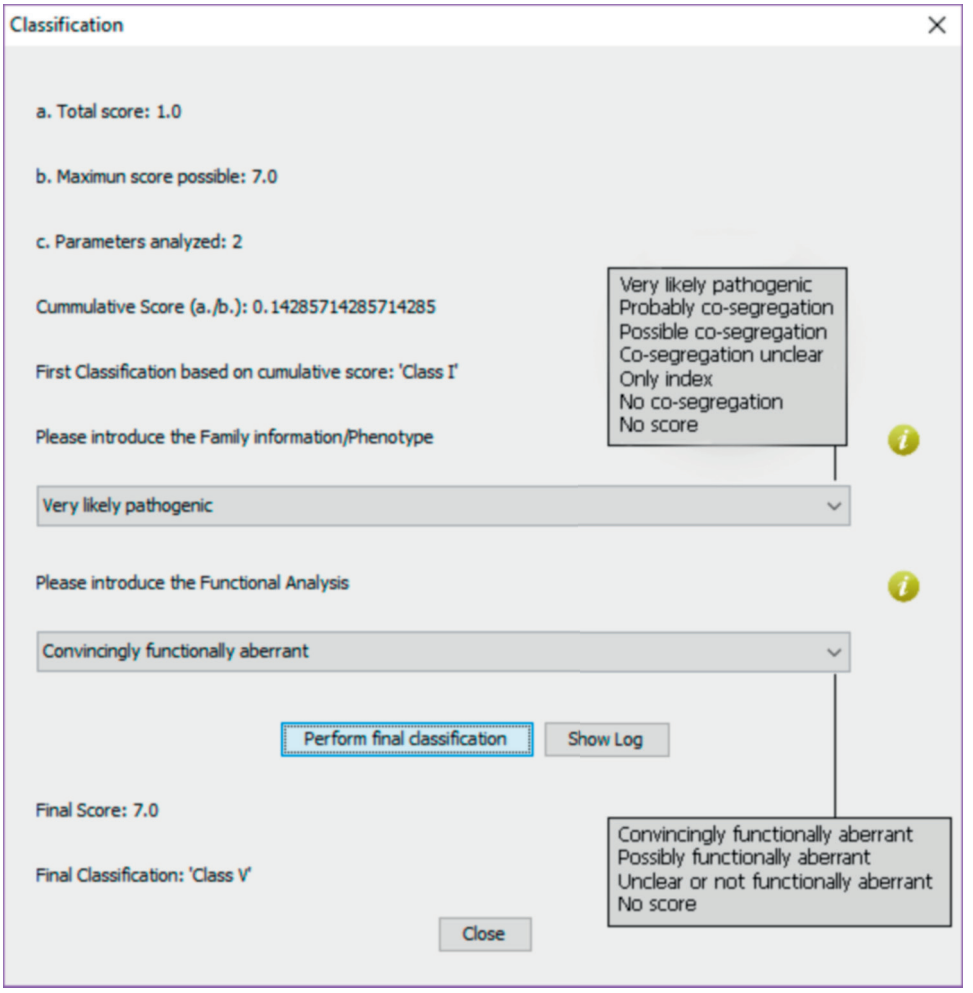


Figure C.2: After assigning the *first label* the cumulative score is used together with the feedback from the expert to compute the label for each sample. This constitutes a manual step, thus the semi-automated character of GeVaCT.

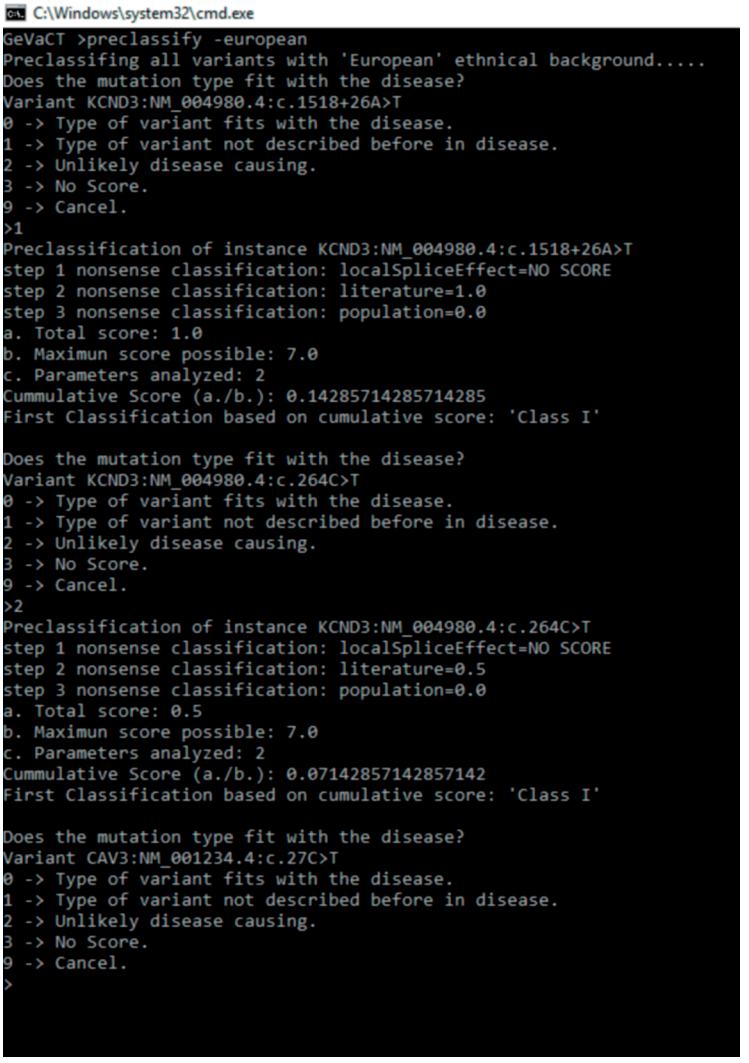


Figure C.3: GeVaCT is also executable from a console interface. The figure portrays the question in step 2 for nonsense and frameshift variants, which rely on the expert criteria based on literature.

Peer-reviewed Publications

Conference proceedings

Grau, I., Sengupta, D., Garcia Lorenzo, M. M., Nowe, A. (2020). An Interpretable Semi-Supervised Classifier using Rough Sets for Amended Self-labeling. In: Proceedings of IEEE International Conference on Fuzzy Systems FUZZ-IEEE 2020, IEEE. (Accepted/In press)

Grau, I., Sengupta, D., Garcia Lorenzo, M. M., Nowe, A. (2018). Interpretable self-labeling semi-supervised classifier. In: Proceedings of the 2nd Workshop on Explainable Artificial Intelligence, International Joint Conference on Artificial Intelligence IJCAI 2018, pp. 52-57.

Grau, I., Nápoles, G., Sengupta, D., Garcia Lorenzo, M. M., Nowe, A. (2017). Training Set Edition Using Rough Set Theory for Semi-supervised Classification. In: Proceedings of the 2nd International Symposium on Fuzzy and Rough Sets, pp. 1-10, Editorial Feijoó.

Nápoles, G., Leon, M., **Grau, I.,** Vanhoof, K. (2017). Fuzzy Cognitive Maps Tool for Scenario Analysis and Pattern Classification. In: Proceedings of the 29th International Conference on Tools with Artificial Intelligence ICTAI 2017, pp. 644-651, IEEE.

Grau, I., Sengupta, D., Farid, D., Manderick, M., Nowe, A., Garcia Lorenzo, M.M., Daneels, D., Bonduelle, M., Croes, D., Van Dooren, S. (2016). Genomic Variant Classifier Tool. In: Proceedings of SAI Intelligent Systems Conference IntelliSys 2016, Vol. 1, pp. 453-456. Springer.

Grau, I., Sengupta, D., Garcia Lorenzo, M. M., Nowe, A. (2016). Grey-Box Model: An ensemble approach for addressing semi-supervised classification problems. In: Pro-

ceedings of the 25th Belgian-Dutch Conference on Machine Learning BENELEARN 2016, pp. 1-3.

Grau, I., Nápoles, G., Papageorgiou, E., Vanhoof, K., Garcia, M. M., Nowe, A. (2015). A Java library for Fuzzy Cognitive Maps. In: Proceedings of the 27th Benelux Conference on Artificial Intelligence BNAIC 2015, pp. 1-8.

Mihaylov, M. E., Jurado Gomez, S., Avellana, N., Razo Zapata, I., Van Moffaert, K., Arco Garcia, L., Bezunartea M., **Grau, I.,** Canadas, A., Nowe, A. (2015). SCANERGY: A Scalable and Modular System for Energy Trading Between Prosumers. In: Proceedings of the International Conference on Autonomous Agents and Multi-agent Systems AAMAS 2015, pp. 1917-1918, ACM.

Journal articles

Nápoles, G., Leon, M., **Grau, I.,**, Vanhoof, K. (2018). FCM Expert: Software Tool for Scenario Analysis and Pattern Classification Based on Fuzzy Cognitive Maps. International Journal on Artificial Intelligence Tools, Vol. 27 (7), World Scientific.

Nápoles, G., Mosquera, C., Falcon, R., **Grau, I.,** Bello, R., Vanhoof, K. (2018). Fuzzy-Rough Cognitive Networks. Neural Networks, Vol. 97, pp. 19-27, Elsevier.

Daneels, D.*, **Grau, I.*,** Sengupta, D., Bonduelle, M-L., Farid, D., Croes, D., Nowe, A., Van Dooren, S. (2016). GeVaCT - Genomic Variant Classifier Tool. European Journal of Human Genetics, 24 (E-Supplement 1), pp. 341-341. Nature Publishing Group.

Nápoles, G., **Grau, I.,** Papageorgiou, E., Bello, R., Vanhoof, K. (2016). Rough Cognitive Networks. Knowledge-Based Systems, Vol. 91, pp. 46-61, Elsevier.

* denotes equal contribution

Book chapters

Bello, M., Nápoles, G., Fuentes, I., **Grau, I.,** Falcon, R., Bello, R., Vanhoof, K. (2019). Fuzzy Activation of Rough Cognitive Ensembles Using OWA Operators. In: Uncertainty Management with Fuzzy and Rough Sets, Vol. 377, pp. 317-335, Springer.

Nápoles G., Leon M., **Grau I.,** Vanhoof K., Bello R. (2018) Fuzzy Cognitive Maps Based Models for Pattern Classification: Advances and Challenges. In: Soft Computing Based Optimization and Decision Models. Studies in Fuzziness and Soft Computing, Vol. 360, Springer.

Bibliography

[1] Adadi A, Berrada M (2018) Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). IEEE Access 6:52138–52160

[2] Adzhubei IA, Schmidt S, Peshkin L, Ramensky VE, Gerasimova A, Bork P, Kondrashov AS, Sunyaev SR (2010) A method and server for predicting damaging missense mutations. Nature Methods 7(4):248–249

[3] Altman NS (1992) An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician 46(3):175–185

[4] Alvarez-Melis D, Jaakkola TS (2018) On the robustness of interpretability methods. In: Proceedings of the ICML 2018 Workshop on Human Interpretability in Machine Learning (WHI 2018), pp 66–71

[5] Arrieta AB, Díaz-Rodríguez N, Ser JD, Bennetot A, Tabik S, Barbado A, Garcia S, Gil-Lopez S, Molina D, Benjamins R, Chatila R, Herrera F (2020) Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion 58:82–115

[6] Bair E (2013) Semi-supervised clustering methods. WIREs Computational Statistics 5(5):349–361

[7] Bastani O, Kim C, Bastani H (2017) Interpretability via model extraction. In: 2017 Workshop on Fairness, Accountability, and Transparency in Machine Learning (FAT/ML 2017), pp 1–5

[8] Belkin M, Niyogi P, Sindhwani V (2006) Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. Journal of Machine Learning Research 7:2399–2434

[9] Bello R, Verdegay JL (2012) Rough sets in the soft computing environment. Information Sciences 212:1–14

[10] Ben-David A (2008) Comparison of classification accuracy using Cohen's weighted kappa. Expert Systems with Applications 34(2):825–832

[11] Benavoli A, Corani G, Mangili F (2016) Should we really use post-hoc tests based on mean-ranks? Journal of Machine Learning Research 17:1–10

[12] Bengio Y, Delalleau O, Le Roux N (2006) Label propagation and quadratic criterion, MIT Press, chap 11, pp 193–216

[13] Bennett KP, Demiriz A (1999) Semi-supervised support vector machines. In: Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II, MIT Press, pp 368–374

[14] Bertsimas D, Delarue A, Jaillet P, Martin S (2019) The price of interpretability. arXiv preprint arXiv:190703419

[15] Birch CP (1999) A new generalized logistic sigmoid growth equation compared with the Richards growth equation. Annals of Botany 83(6):713–723

[16] Blum A, Chawla S (2001) Learning from labeled and unlabeled data using graph mincuts. In: Proceedings of the Eighteenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., ICML '01, pp 19–26

[17] Blum A, Mitchell T (1998) Combining labeled and unlabeled data with co-training. In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, ACM, COLT' 98, pp 92–100

[18] Breiman L (2001) Random forests. Machine Learning 45(1):5–32

[19] Breiman L, Friedman J, Stone CJ, Olshen RA (1984) Classification and regression trees. CRC press

[20] Cevikalp H, Franc V (2017) Large-scale robust transductive support vector machines. Neurocomputing 235:199–209

[21] Chakraborty S, Tomsett R, Raghavendra R, Harborne D, Alzantot M, Cerutti F, Srivastava M, Preece A, Julier S, Rao RM, et al. (2017) Interpretability of deep learning models: a survey of results. In: Proceedings of the IEEE Smart World Congress 2017 Workshop: DAIS

[22] Che Z, Purushotham S, Khemani R, Liu Y (2015) Distilling knowledge from deep networks with applications to healthcare domain. arXiv preprint arXiv:151203542

[23] Chen J, Chang Y, Hobbs B, Castaldi P, Cho M, Silverman E, Dy J (2016) Interpretable clustering via discriminative rectangle mixture model. In: Proceedings of the IEEE 16th International Conference on Data Mining (ICDM), pp 823–828

[24] Chi S, Li X, Tian Y, Li J, Kong X, Ding K, Weng C, Li J (2019) Semi-supervised learning to improve generalizability of risk prediction models. Journal of Biomedical Informatics 92:103117

[25] Cilia E, Pancsa R, Tompa P, Lenaerts T, Vranken WF (2013) From protein sequence to dynamics and disorder with DynaMine. Nature Communications 4(1):1–10

[26] Cilia E, Pancsa R, Tompa P, Lenaerts T, Vranken WF (2014) The DynaMine webserver: predicting protein dynamics from sequence. Nucleic Acids Research 42(W1):W264–W270

[27] Cohen J (1960) A coefficient of agreement for nominal scales. Educational and Psychological Measurement 20(1):37–46

[28] Cohen WW (1995) Fast effective rule induction. In: Prieditis A, Russell S (eds) Proceedings of the Twelfth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., pp 115–123

[29] Cornelis C, De Cock M, Radzikowska AM (2008) Fuzzy rough sets: from theory into practice. Handbook of Granular Computing pp 533–552

[30] Cour T, Sapp B, Taskar B (2011) Learning from partial labels. Journal of Machine Learning Research 12:1501–1536

[31] Dai Z, Yang Z, Yang F, Cohen WW, Salakhutdinov RR (2017) Good semi-supervised learning that requires a bad gan. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) Advances in Neural Information Processing Systems 30, Curran Associates, Inc., pp 6510–6520

[32] Danecek P, Auton A, Abecasis G, Albers CA, Banks E, DePristo MA, Handsaker RE, Lunter G, Marth GT, Sherry ST, McVean G, Durbin R (2011) The variant call format and VCFtools. Bioinformatics 27(15):2156–2158

[33] Denis F, Laurent A, Gilleron R, Tommasi M (2003) Text classification and co-training from positive and unlabeled examples. In: Proceedings of the ICML 2003 Workshop: The Continuum from Labeled to Unlabeled Data, pp 80–87

[34] Desmet FO, Hamroun D, Lalande M, Collod-Béroud G, Claustres M, Béroud C (2009) Human Splicing Finder: an online bioinformatics tool to predict splicing signals. Nucleic Acids Research 37(9):e67–e67

[35] Diederik PK, Welling M, et al. (2014) Auto-encoding variational bayes. In: Proceedings of the 2nd International Conference on Learning Representations, ICLR 2014, vol 1

[36] Ding S, Zhu Z, Zhang X (2017) An overview on semi-supervised support vector machine. Neural Computing and Applications 28(5):969–978

[37] Do K, Tran T (2020) Theory and evaluation metrics for learning disentangled representations. In: Eighth International Conference on Learning Representations, ICLR 2020, pp 1–30

[38] Dobson CM (2003) Protein folding and misfolding. Nature 426(6968):884–890

[39] Doran D, Schulz S, Besold TR (2017) What does explainable AI really mean? A new conceptualization of perspectives. arXiv preprint arXiv:171000794

[40] Doshi-Velez F, Kim B (2018) Considerations for evaluation and generalization in interpretable machine learning. In: Explainable and Interpretable Models in Computer Vision and Machine Learning, Springer International Publishing, pp 3–17

[41] Dua D, Graff C (2019) UCI machine learning repository

[42] Efron B, Hastie T, Johnstone I, Tibshirani R, et al. (2004) Least angle regression. The Annals of Statistics 32(2):407–499

[43] Falcon R, Nápoles G, Bello R, Vanhoof K (2019) Granular cognitive maps: a review. Granular Computing 4(3):451–467

[44] Fan M, Zhang X, Du L, Chen L, Tao D (2018) Semi-supervised learning through label propagation on geodesics. IEEE Transactions on Cybernetics 48(5):1486–1499

[45] Fazakis N, Karlos S, Kotsiantis S, Sgarbas K (2016) Self-trained LMT for semisupervised learning. Computational Intelligence and Neuroscience 2016:10

[46] Fazakis N, Karlos S, Kotsiantis S, Sgarbas K (2017) Self-trained rotation forest for semi-supervised learning. Journal of Intelligent & Fuzzy Systems 32(1):711–722

[47] Fernández-Delgado M, Cernadas E, Barro S, Amorim D (2014) Do we need hundreds of classifiers to solve real world classification problems. Journal of Machine Learning Research 15(1):3133–3181

[48] Fisher A, Rudin C, Dominici F (2019) All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. Journal of Machine Learning Research 20(177):1–81

[49] Frank E, Witten IH (1998) Generating accurate rule sets without global optimization. In: Proceedings of the Fifteenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., ICML '98, pp 144–151

[50] Freund Y, Seung HS, Shamir E, Tishby N (1997) Selective sampling using the query by committee algorithm. Machine Learning 28(2-3):133–168

[51] Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Annals of Statistics 29(5):1189–1232

[52] Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the American Statistical Association 32(200):675–701

[53] Frosst N, Hinton G (2017) Distilling a neural network into a soft decision tree. In: Proceedings of the AI*IA 2017 First International Workshop on Comprehensibility and Explanation in AI and ML, pp 1–8

[54] Fürnkranz J, Gamberger D, Lavrač N (2012) Learning rule sets. In: Foundations of Rule Learning, Springer Berlin Heidelberg, pp 171–186

[55] Gevrey M, Dimopoulos I, Lek S (2003) Review and comparison of methods to study the contribution of variables in artificial neural network models. Ecological Modelling 160(3):249–264

[56] Gilpin LH, Bau D, Yuan BZ, Bajwa A, Specter M, Kagal L (2018) Explaining explanations: An overview of interpretability of machine learning. In: Proceedings of the 5th IEEE International Conference on Data Science and Advanced Analytics (DSAA), IEEE, pp 80–89

[57] Giuseppe C, Egle C, Antonio C, Giampiero M, Domenico O, Antonino M, Brugada P (2019) Update on Brugada syndrome 2019. Current Problems in Cardiology

[58] Goldstein A, Kapelner A, Bleich J, Pitkin E (2015) Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. Journal of Computational and Graphical Statistics 24(1):44–65

[59] Gomez O, Holter S, Yuan J, Bertini E (2020) ViCE: visual counterfactual explanations for machine learning models. In: Proceedings of the 25th International Conference on Intelligent User Interfaces, pp 531–535

[60] Gong C, Tao D, Liu W, Liu L, Yang J (2017) Label propagation via teaching-to-learn and learning-to-teach. IEEE Transactions on Neural Networks and Learning Systems 28(6):1452–1465

[61] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Advances in Neural Information Processing Systems 27, Curran Associates, Inc., pp 2672–2680

[62] Goodman B, Flaxman S (2017) European Union regulations on algorithmic decision-making and a "right to explanation". AI Magazine 38(3):50–57

[63] Gourraud JB, Barc J, Thollet A, Le Scouarnec S, Le Marec H, Schott JJ, Redon R, Probst V (2016) The Brugada syndrome: A rare arrhythmia disorder with complex inheritance. Frontiers in Cardiovascular Medicine 3(9):1–11

[64] Grau I, Sengupta D, Garcia Lorenzo M, Nowe A (2016) Grey-box model: An ensemble approach for addressing semi-supervised classification problems. In: Proceedings of the 25th Belgian-Dutch Conference on Machine Learning, BENELEARN 2016, pp 1–3

[65] Grau I, Sengupta D, Lorenzo MMG, Nowe A (2018) Interpretable self-labeling semi-supervised classifier. In: Proceedings of the IJCAI/ECAI 2018 2nd Workshop on Explainable Artificial Intelligence, pp 52–57

[66] Gunning D, Aha DW (2019) Darpa's explainable artificial intelligence program. AI Magazine 40(2):44–58

[67] Hady MFA, Schwenker F (2008) Co-training by committee: a new semi-supervised learning framework. In: IEEE International Conference on Data Mining Workshops, IEEE, ICDMW '08, pp 563–572

[68] Halder A, Ghosh S, Ghosh A (2010) Ant based semi-supervised classification. In: Dorigo M, Birattari M, Di Caro GA, Doursat R, Engelbrecht AP, Floreano D, Gambardella LM, Groß R, Şahin E, Sayama H, Stützle T (eds) Proceedings of the 7th International Conference on Swarm Intelligence, Springer Berlin Heidelberg, Ants 2010, pp 376–383

[69] Hara S, Hayashi K (2018) Making tree ensembles interpretable: A bayesian model selection approach. In: Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics, PMLR, vol 84, pp 77–85

[70] Hastie T, Tibshirani R, Friedman J (2001) The Elements of Statistical Learning. Springer Series in Statistics, Springer New York Inc.

[71] Hastie T, Tibshirani R, Wainwright M (2015) Statistical learning with sparsity: the lasso and generalizations. CRC press

[72] Hecht-Nielsen R (1989) Theory of the backpropagation neural network. In: Proceedings of the International Joint Conference on Neural Networks, IEEE, pp 593–605

[73] Hoffman RR, Mueller ST, Klein G, Litman J (2018) Metrics for explainable AI: Challenges and prospects. arXiv preprint arXiv:181204608

[74] Hofman N, Tan HL, Alders M, Kolder I, de Haij S, Mannens M, Lombardi MP, dit Deprez RLL, van Langen I, Wilde AA (2013) Yield of molecular and clinical testing for arrhythmia syndromes: report of a 15 years' experience. Circulation 128(14):1513–1521

[75] Holm S (1979) A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics 6(2):65–70

[76] Hüllermeier E, Cheng W (2015) Superset learning based on generalized loss minimization. In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, Springer International Publishing, ECML PKDD 2015, pp 260–275

[77] Interactive Biosoftware (2020) Alamut batch, a human variant annotation software

[78] Japkowicz N, Shah M (2011) Evaluating learning algorithms: a classification perspective. Cambridge University Press

[79] Jo B, Findling RL, Wang CP, Hastie TJ, Youngstrom EA, Arnold LE, Fristad MA, Horwitz SM (2017) Targeted use of growth mixture modeling: a learning perspective. Statistics in Medicine 36(4):671–686

[80] Joachims T (1999) Transductive inference for text classification using support vector machines. In: Proceedings of the Sixteenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., ICML '99, pp 200–209

[81] Kamal Nigam AM, Mitchell T (2006) Semi-Supervised Text Classification Using EM, MIT Press, pp 33–55

[82] Karczewski KJ, Weisburd B, Thomas B, Solomonson M, Ruderfer DM, et al. (2016) The ExAC browser: displaying reference data information from over 60,000 exomes. Nucleic Acids Research 45(D1):D840–D845

[83] Keerthi SS, Shevade SK, Bhattacharyya C, Murthy KRK (2001) Improvements to platt's SMO algorithm for SVM classifier design. Neural Computation 13(3):637–649

[84] Kidd AL (2012) Knowledge acquisition for expert systems: A practical handbook. Springer Science & Business Media

[85] Kim M, Lee Dg, Shin H (2019) Semi-supervised learning for hierarchically structured networks. Pattern Recognition 95:191–200

[86] Kingma DP, Mohamed S, Rezende DJ, Welling M (2014) Semi-supervised learning with deep generative models. In: Advances in Neural Information Processing Systems 27, Curran Associates, Inc., pp 3581–3589

[87] Kircher M, Witten DM, Jain P, O'roak BJ, Cooper GM, Shendure J (2014) A general framework for estimating the relative pathogenicity of human genetic variants. Nature Genetics 46(3):310–315

[88] Kostopoulos G, Karlos S, Kotsiantis S, Ragos O (2018) Semi-supervised regression: A recent review. Journal of Intelligent & Fuzzy Systems 35:1483–1500

[89] Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images. Tech. rep., University of Toronto

[90] Kumar P, Henikoff S, Ng PC (2009) Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm. Nature Protocols 4(7):1073–1082

[91] Lafferty JD, McCallum A, Pereira FC (2001) Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., ICML '01, pp 282–289

[92] Lander ES, Linton LM, Birren B, Nusbaum C, Zody MC, et al. (2001) Initial sequencing and analysis of the human genome. Nature 409(6822):860–921

[93] Landrum MJ, Lee JM, Benson M, Brown G, Chao C, Chitipiralla S, Gu B, Hart J, Hoffman D, Hoover J, Jang W, Katz K, Ovetsky M, Riley G, Sethi A, Tully R, Villamarin-Salomon R, Rubinstein W, Maglott DR (2016) Clinvar: Public archive of interpretations of clinically relevant variants. Nucleic Acids Research 44(D1):D862–d868

[94] Landrum MJ, Lee JM, Benson M, Brown GR, Chao C, Chitipiralla S, Gu B, Hart J, Hoffman D, Jang W, Karapetyan K, Katz K, Liu C, Maddipatla Z,

Malheiro A, McDaniel K, Ovetsky M, Riley G, Zhou G, Holmes JB, Kattman BL, Maglott DR (2017) Clinvar: improving access to variant interpretations and supporting evidence. Nucleic Acids Research 46(D1):D1062–d1067

[95] LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11):2278–2324

[96] Lee DH (2013) Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Proceedings of the ICML 2013 Workshop on Challenges in Representation Learning, pp 1–6

[97] Lee WS, Liu B (2003) Learning with positive and unlabeled examples using weighted logistic regression. In: Proceedings of the Twentieth International Conference on Machine Learning, vol 3, pp 448–455

[98] Li M, Zhou ZH (2005) Setred: Self-training with editing. In: Proceedings of the 9th Pacific-Asia Conference in Knowledge Discovery and Data Mining, Springer Berlin Heidelberg, PAKDD 2005, pp 611–621

[99] Li M, Zhou ZH (2007) Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 37(6):1088–1098

[100] Li Y, Wang Y, Bi C, Jiang X (2018) Revisiting transductive support vector machines with margin distribution embedding. Knowledge-Based Systems 152:200–214

[101] Lipton ZC (2016) The mythos of model interpretability. In: Proceedings of the ICML 2016 Workshop on Human Interpretability in Machine Learning, WHI 2016, pp 96–100

[102] Liu H, Hussain F, Tan CL, Dash M (2002) Discretization: An enabling technique. Data Mining and Knowledge Discovery 6(4):393–423

[103] Liu L, Dietterich T (2014) Learnability of the superset label learning problem. In: Proceedings of the 31st International Conference on Machine Learning, pp 1629–1637

[104] Lundberg SM, Lee SI (2017) A unified approach to interpreting model predictions. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) Advances in Neural Information Processing Systems 30, Curran Associates, Inc., pp 4765–4774

[105] Lundberg SM, Erion GG, Lee SI (2018) Consistent individualized feature attribution for tree ensembles. arXiv preprint arXiv:180203888

[106] Lundberg SM, Erion G, Chen H, DeGrave A, Prutkin JM, Nair B, Katz R, Himmelfarb J, Bansal N, Lee SI (2020) From local explanations to global understanding with explainable AI for trees. Nature Machine Intelligence 2(1):2522–5839

[107] Martin-Barragan B, Lillo R, Romo J (2014) Interpretable support vector machines for functional data. European Journal of Operational Research 232(1):146–155

[108] Mathe E, Olivier M, Kato S, Ishioka C, Hainaut P, Tavtigian SV (2006) Computational approaches for predicting the biological effect of p53 missense mutations: a comparison of three sequence analysis based methods. Nucleic Acids Research 34(5):1317–1325

[109] Mehdi TF, Singh G, Mitchell JA, Moses AM (2019) Variational infinite heterogeneous mixture model for semi-supervised clustering of heart enhancers. Bioinformatics 35(18):3232–3239

[110] Miller T (2019) Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence 267:1–38

[111] Mohseni S, Zarei N, Ragan ED (2018) A survey of evaluation methods and measures for interpretable machine learning. arXiv preprint arXiv:181111839

[112] Molnar C (2019) Interpretable Machine Learning. Leanpub

[113] Monasky MM, Micaglio E, Ciconte G, Pappone C (2020) Brugada syndrome: Oligogenic or mendelian disease? International Journal of Molecular Sciences 21(5):1–19

[114] Mothilal RK, Sharma A, Tan C (2020) Explaining machine learning classifiers through diverse counterfactual explanations. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (FAT* '20), ACM, pp 607–617

[115] N S, Paige B, van de Meent JW, Desmaison A, Goodman N, Kohli P, Wood F, Torr P (2017) Learning disentangled representations with semi-supervised deep generative models. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) Advances in Neural Information Processing Systems 30, Curran Associates, Inc., pp 5925–5935

[116] Nelles O (2013) Nonlinear system identification: from classical approaches to neural networks and fuzzy models. Springer Science & Business Media

[117] Netzer Y, Wang T, Coates A, Bissacco A, Wu B, Ng AY (2011) Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011, pp 1–9

[118] Niculescu-Mizil A, Caruana R (2005) Predicting good probabilities with supervised learning. In: Proceedings of the 22nd International Conference on Machine Learning, ACM, pp 625–632

[119] Nie F, Tian L, Wang R, Li X (2019) Multiview semi-supervised learning model for image classification. IEEE Transactions on Knowledge and Data Engineering

[120] NIH (2020) What kinds of gene mutations are possible? - genetics home reference

[121] Nowe A, Bonduelle M, Brugada P, Deschepper R, Lenaerts T, Van Dooren S, De Asmundis C, Gidron Y, Bilsen J, De Couck M (2020) IMAGica: An Integrative personalized Medical Approach for Genetic diseases, Inherited Cardiac Arrhythmias as a model

[122] Odena A (2016) Semi-supervised learning with generative adversarial networks. In: Proceedings of the ICML 2016 Workshop on Data Efficient Machine Learning, pp 1–3

[123] Oliver A, Odena A, Raffel CA, Cubuk ED, Goodfellow I (2018) Realistic evaluation of deep semi-supervised learning algorithms. In: Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N, Garnett R (eds) Advances in Neural Information Processing Systems 31, Curran Associates, Inc., pp 3235–3246

[124] Pancsa R, Varadi M, Tompa P, Vranken WF (2016) Start2Fold: a database of hydrogen/deuterium exchange data on protein folding and stability. Nucleic Acids Research 44:D429–D434

[125] Papadimitriou S, Gazzo A, Versbraegen N, Nachtegael C, Aerts J, Moreau Y, Van Dooren S, Nowé A, Smits G, Lenaerts T (2019) Predicting disease-causing variant combinations. Proceedings of the National Academy of Sciences of the United States of America 116(24):11878–11887

[126] Papernot N, McDaniel P (2018) Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. arXiv preprint arXiv:180304765

[127] Park S, Lee J, Kim K (2019) Semi-supervised distributed representations of documents for sentiment analysis. Neural Networks 119:139–150

[128] Pawlak Z (1982) Rough sets. International Journal of Computer & Information Sciences 11(5):341–356

[129] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12:2825–2830

[130] Pedrycz W, Skowron A, Kreinovich V (2008) Handbook of granular computing. John Wiley & Sons

[131] Platt J (1998) Fast training of support vector machines using sequential minimal optimization. In: Schoelkopf B, Burges C, Smola A (eds) Advances in Kernel Methods - Support Vector Learning, MIT Press

[132] Platt J (1999) Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in Large Margin Classifiers 10(3):61–74

[133] Pollard KS, Hubisz MJ, Rosenbloom KR, Siepel A (2010) Detection of nonneutral substitution rates on mammalian phylogenies. Genome Research 20(1):110–121

[134] Quinlan JR (1993) C4.5: Programs for machine learning. Morgan Kaufmann Publishers Inc.

[135] Raimondi D, Orlando G, Pancsa R, Khan T, Vranken WF (2017) Exploring the Sequence-based Prediction of Folding Initiation Sites in Proteins. Scientific Reports 7(1):1–11

[136] Rasmus A, Berglund M, Honkala M, Valpola H, Raiko T (2015) Semi-supervised learning with ladder networks. In: Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R (eds) Advances in Neural Information Processing Systems 28, Curran Associates, Inc., pp 3546–3554

[137] Rätsch G, Sonnenburg S, Schäfer C (2006) Learning interpretable SVMs for biological sequence classification. BMC Bioinformatics 7(S9)

[138] Ren X, Pan H, Jing Z, Gao L (2019) Semi-supervised video object segmentation with recurrent neural network. In: Proceedings of the 2019 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), pp 7284–7293

[139] Reuter JA, Spacek DV, Snyder MP (2015) High-throughput sequencing technologies. Molecular Cell 58(4):586–597

[140] Reynaud E (2010) Protein misfolding and degenerative diseases. Nature Education 3(9):28

[141] Ribeiro MT, Singh S, Guestrin C (2016) Why should I trust you?: Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, KDD 2016, pp 1135–1144

[142] Richards S, Aziz N, Bale S, Bick D, Das S, Gastier-Foster J, Grody WW, Hegde M, Lyon E, Spector E, Voelkerding K, Rehm HL (2015) Standards and guidelines for the interpretation of sequence variants: A joint consensus recommendation of the american college of medical genetics and genomics and the association for molecular pathology. Genetics in Medicine 17(5):405–424

[143] Robnik-Šikonja M, Bohanec M (2018) Perturbation-based explanations of prediction models. In: Human and Machine Learning, Springer, pp 159–175

[144] Roijers DM, Vamplew P, Whiteson S, Nl AW, Dazeley R (2013) A survey of multi-objective sequential decision-making. Journal of Artificial Intelligence Research 48:67–113

[145] Rădulescu R, Mannion P, Zhang Y, Roijers DM, Nowé A (2020) A utility-based analysis of equilibria in multi-objective normal-form games. The Knowledge Engineering Review 35:32

[146] Russell S, Norvig P (2010) Artificial Intelligence: A Modern Approach. Prentice Hall

[147] Rustamov RM, Klosowski JT (2018) Interpretable graph-based semi-supervised learning via flows. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, pp 3976–3983

[148] Sachan DS, Zaheer M, Salakhutdinov R (2019) Revisiting lstm networks for semi-supervised text classification via mixed objective function. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 33, pp 6940–6948

[149] Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X (2016) Improved techniques for training gans. In: Advances in Neural Information Processing Systems 29, Curran Associates, Inc., pp 2234–2242

[150] Settles B (2009) Active learning literature survey. Tech. Rep. 1648, University of Wisconsin–Madison

[151] Shapley LS (1953) A value for n-person games. Contributions to the Theory of Games 2(28):307–317

[152] Sherry ST (2001) dbsnp: the ncbi database of genetic variation. Nucleic Acids Research 29(1):308–311

[153] Stenson PD, Mort M, Ball EV, Howells K, Phillips AD, Cooper DN, Thomas NS (2009) The human gene mutation database: 2008 update. Genome Medicine 1(1):13

[154] Subramanya A, Talukdar PP (2014) Graph-based semi-supervised learning. Synthesis Lectures on Artificial Intelligence and Machine Learning 8(4):1–125

[155] Tamposis IA, Tsirigos KD, Theodoropoulou MC, Kontou PI, Bagos PG (2018) Semi-supervised learning of hidden markov models for biological sequence analysis. Bioinformatics 35(13):2208–2215

[156] Tan S, Caruana R, Hooker G, Lou Y (2018) Distill-and-compare: Auditing black-box models using transparent model distillation. In: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, pp 303–310

[157] Touw WG, Bayjanov JR, Overmars L, Backus L, Boekhorst J, Wels M, van Hijum SA (2012) Data mining in the life sciences with random forest: a walk in the park or lost in the jungle? Briefings in Bioinformatics 14(3):315–326

[158] Triguero I, García S, Herrera F (2015) Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. Knowledge and Information Systems 42(2):245–284

[159] Van Engelen JE, Hoos HH (2020) A survey on semi-supervised learning. Machine Learning 109(2):373–440

[160] Venter JC, Adams MD, Myers EW, Li PW, Mural RJ, et al. (2001) The sequence of the human genome. Science 291(5507):1304–1351

[161] Versbraegen N, FouchÃľ A, Nachtegael C, Papadimitriou S, Gazzo A, Smits G, Lenaerts T (2019) Using game theory and decision decomposition to effectively discern and characterise bi-locus diseases. Artificial Intelligence in Medicine 99:101690

[162] Vluymans S, Mac Parthaláin N, Cornelis C, Saeys Y (2016) Fuzzy rough sets for self-labelling: An exploratory analysis. In: Proceedings of the IEEE International Conference on Fuzzy Systems, IEEE, FUZZ-IEEE, pp 931–938

[163] Wachter S, Mittelstadt B, Russell C (2017) Counterfactual explanations without opening the black box: Automated decisions and the GDPR. Harv JL & Tech 31:841

[164] Wagner T, Guha S, Kasiviswanathan S, Mishra N (2018) Semi-supervised learning on data streams via temporal label propagation. In: Dy J, Krause A (eds)

Proceedings of the 35th International Conference on Machine Learning, PMLR, Proceedings of Machine Learning Research, vol 80, pp 5095–5104

[165] Wainberg M, Alipanahi B, Frey BJ (2016) Are random forests truly the best classifiers? Journal of Machine Learning Research 17(1):3837–3841

[166] Wang G, Dunbrack Jr RL (2003) PISCES: a protein sequence culling server. Bioinformatics 19(12):1589–1591

[167] Wang Y, Xu X, Zhao H, Hua Z (2010) Semi-supervised learning based on nearest neighbor rule and cut edges. Knowledge-Based Systems 23(6):547–554

[168] Wang Y, Mei C, Zhou Y, Wang Y, Zheng C, Zhen X, Xiong Y, Chen P, Zhang J, Wang B (2019) Semi-supervised prediction of protein interaction sites from unlabeled sample information. BMC Bioinformatics 20(25):1–10

[169] Weston J, Ratle F, Mobahi H, Collobert R (2012) Deep learning via semi-supervised embedding. In: Neural Networks: Tricks of the Trade, Springer, pp 639–655

[170] Wilcoxon F (1945) Individual comparisons by ranking methods. Biometrics 1:80–83

[171] Wilson DR, Martinez TR (1997) Improved heterogeneous distance functions. Journal of Artificial Intelligence Research 6:1–34

[172] Witten IH, Frank E, Hall MA, Pal CJ (2017) Chapter 11 - Beyond supervised and unsupervised learning, fourth edition edn, Morgan Kaufmann Publishers Inc., pp 467–478

[173] Wu C, Wu F, Wu S, Yuan Z, Liu J, Huang Y (2019) Semi-supervised dimensional sentiment analysis with variational autoencoder. Knowledge-Based Systems 165:30–39

[174] Wu Y, Ai M, Bardeesi ASA, Xu L, Zheng J, Zheng D, Yin K, Wu Q, Zhang L, Huang L, Cheng J (2017) Brugada syndrome: a fatal disease with complex genetic etiologies – still a long way to go. Forensic Sciences Research 2(3):115–125

[175] Xu Z, Liang J, Dang C, Chin K (2002) Inclusion degree: a perspective on measures for rough set data analysis. Information Sciences 141(3):227–236

[176] Yan P, Li G, Xie Y, Li Z, Wang C, Chen T, Lin L (2019) Semi-supervised video salient object detection using pseudo-labels. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV)

[177] Yarowsky D (1995) Unsupervised word sense disambiguation rivaling supervised methods. In: Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, ACL '95, pp 189–196

[178] Zadrozny B, Elkan C (2001) Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In: Proceedings of the Eighteenth International Conference on Machine Learning, vol 1, pp 609–616

[179] Zhang C, Liu C, Zhang X, Almpanidis G (2017) An up-to-date comparison of state-of-the-art classification algorithms. Expert Systems with Applications 82:128–150

[180] Zhang C, Cheng J, Tian Q (2019) Unsupervised and semi-supervised image classification with weak semantic consistency. IEEE Transactions on Multimedia 21(10):2482–2491

[181] Zhao J, Wang T, Yatskar M, Ordonez V, Chang KW (2017) Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp 2979–2989

[182] Zhou D, Bousquet O, Lal TN, Weston J, Schölkopf B (2004) Learning with local and global consistency. In: Advances in Neural Information Processing Systems 16, MIT Press, pp 321–328

[183] Zhou Y, Goldman S (2004) Democratic co-learning. In: Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence, IEEE, ICTAI 2004, pp 594–602

[184] Zhou ZH, Li M (2005) Tri-training: Exploiting unlabeled data using three classifiers. IEEE Transactions on Knowledge and Data Engineering 17(11):1529–1541

[185] Zhou ZH, Xu JM (2007) On the relation between multi-instance learning and semi-supervised learning. In: Proceedings of the 24th International Conference on Machine Learning, ACM, pp 1167–1174

[186] Zhu X (2005) Semi-supervised learning literature survey. Tech. Rep. 1530, University of Wisconsin-Madison

[187] Zhu X, Ghahramani Z (2002) Learning from labeled and unlabeled data with label propagation. Tech. Rep. Cmu-cald-02-107, School of Computer Science, Carnegie Mellon University

[188] Zhu X, Goldberg A (2009) Introduction to semi-supervised learning. Morgan & Claypool Publishers

[189] Zhu X, Wu X (2004) Class noise vs. attribute noise: A quantitative study. Artificial Intelligence Review 22(3):177–210

[190] Zintgraf LM, Roijers DM, Jonker CM, Nowé A (2018) Ordered preference elicitation strategies for supporting multi-objective decision making. In: Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), International Foundation for Autonomous Agents and Multiagent Systems, vol 9