Characterizing Model Uncertainty in Ensemble Learning -

Towards more Robust Representation and Learning of Tree Ensemble Methods

Malte Nalenz

München 2021

Characterizing Model Uncertainty in Ensemble Learning -

Towards more Robust Representation and Learning of Tree Ensemble Methods

Malte Nalenz

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig–Maximilians–Universität München

vorgelegt von
Malte Nalenz
aus München

München, den 16. November 2021

Erstgutachter: Prof. Thomas Augustin
Zweitgutachterin: Prof. Carolin Strobl
Drittgutachter: Prof. Volker Schmid
Tag der Einreichung: 16. November 2021
Tag der Disputation: 05. Mai 2022

Acknowledgements

I would like to thank all the people without whom this work would not have been possible in some way or another. Especially, I would like to thank ...

Thomas Augustin for his courage, dedication and encouragement as a supervisor. I greatly enjoyed the discussions, that made me think differently about statistics and from which I learned a lot. I'm very grateful for his trust in my activities that allowed me to pursue my research with freedom, supervise interesting thesis projects and teach interesting courses.

Carolin Strobl and Volker Schmid for their interest in my work and their kind willingness to examine this dissertation. Also, I would like to thank Anne-Laure Boulesteix and Christian Heumann for steering the examination committee.

Mattias Villani, whose great support and encouragement made me pursue a PhD.

the members of my AG, Thomas Augustin, Hannah Blocher, Eva Endres, Cornelia Fütterer, Christoph Jansen, Dominik Kreiss, Aziz Omar, Julian Rodemann, Georg Schollmeyer and Patrick Schwaferts for the interesting talks and discussions, and the smooth handling of administrative work. I also greatly enjoyed the lunch breaks at our all-favourite pizza place.

my co-authors Thomas Augustin, Luisa Ebner, Cornelia Fütterer, Frank can Harmelen, Dominik Kreiss, Heidi Seibold, Anette ten Teije and Mattias Villani for the interesting projects that I had the chance to be a part of and the smooth collaborations.

the colleagues at the institute for statistics. I greatly enjoyed the welcoming and inspiring atmosphere, the coffee breaks and interesting talks.

Elke Höfner and Brigitte Maxa for their friendly help with the administrative side.

the LMU mentoring program of faculty 16 for the interesting workshops and the support with equipment in the challenging COVID-times.

my sister Anna, and my parents Günter and Petra, for always having my back even if they don't always know what it exactly is that I'm doing for a living.

my partner Luisa, and my cats Nori and Billie for reminding me what's important in life besides statistics.

Zusammenfassung

Hauptziel dieser Arbeit ist es, Entscheidungswälder in strukturell einfacherere Modelle zu transformieren. Dabei gilt es die Vorteile von Entscheidungswäldern zu erhalten – hohe Prädiktionsgüte und Stabilität – aber gleichzeitig die Komplexität so stark zu reduzieren, dass sich zentrale Zusammenhänge in den Daten interpretieren lassen. Die vorgestellten Methoden ermöglichen zudem neue Wege um Modell- und Prädiktionsunsicherheiten zu quantifizieren. Desweiteren befasst sich diese Arbeit mit Anwendungen von maschinellem Lernen in Situationen mit hoher – und teils komplexer – Unsicherheit, sowie mit der Reproduzierbarkeit von statistischen Methoden in wissenschaftlichen Artikeln.

Der erste Teil der Arbeit präsentiert Ansätze, um Entscheidungswälder umzuformen und zu vereinfachen. Dabei bauen drei der Beiträge direkt auf dem *RuleFit* Ansatz auf, bei welchem in einem ersten Schritt Entscheidungswälder in ihre elementaren Regeln zerlegt werden. In einem zweiten Schritt werden regularisierte Regressionverfahren verwendet, um zu einer möglichst kleinen Menge von Entscheidungsregeln zu gelangen. In dieser Arbeit werden sowohl Bayesianische Ansätze, über die *horseshoe prior*, als auch L1-regularisierte Regressionsverfahren verwendet. Dabei wird auch die Komplexität der Regeln – Anzahl Bedingungen und Umfang – mit berücksichtigt.
Der Bayesianische Ansatz erlaubt eine Inklusion der Regelkomplexität direkt in das hierarchische Modell, in Form einer *rule structured prior*. Zudem lassen sich aus der a-Posteriori Verteilung neue Wege zur Quantifierung von Unsicherheit, sowohl von Koeffizientenschätzern, als auch der daraus abgeleiteten Statistiken, wie der Variablenwichtigkeit, ableiten.
Ein weiterer untersuchter Ansatz ist die Zusammenfassung von ähnlichen Entscheidungsregeln in Regeln mit mengenwertigen Schwellenwerten. Durch das Mitteln über viele ähnliche Regeln werden so die Glättungseigenschaften von Entscheidungswäldern imitiert. Gleichzeitig sind diese komprimierten Regeln weiterhin interpretierbar. Zur Gruppierung der Entscheidungsregeln werden Clustering Verfahren verwendet, welche die zentralsten mengenwertigen Regeln in den Daten extrahieren.
Desweiteren wird die Inkludierung von vorhandenem Expertenwissen in Rule Ensembles untersucht. Dazu wird das Rule Ensemble mit Regeln basierend auf Expertenwissen oder aus medizinischen Richtlinien angereichert. Ein modifiziertes Penalisierungsverfahren wird entwickelt um der Unsicherheit der verschiedenen Wissensquellen angemessen Rechnung zu tragen.
In einem weiteren Beitrag wird die Darstellung von Entscheidungswäldern durch einen mengenwertigen Entscheidungsbaum vorgestellt. Die Modellunsicherheit, über den korrekten Schwellenwert und die zu verwendende Kovariable für den Test, wird bereits während der Induktion berücksichtigt. Der resultierende Entscheidungsbaum mit mengenwertigen Entscheidungen an jedem Knoten ist eine spezielle Form eines Entscheidungswaldes, welcher Eigenschaften eines Random Forest imitiert, aber gleichzeit um eine zentrale Baumstruktur zentriert ist.

Der zweite Teil der Arbeit befasst sich mit Anwendungen von regularisierter Regression und Entscheidungswäldern in Situationen mit hoher Unsicherheit.
Ein wichtiges und herausforderndes Feld der personalisierten Medizin ist die Identifizierung von genetischen Biomarkern. Um die Konsistenz der Variablenselektion zu erhöhen, wird eine Erweiterung der L1-Regularisierung vorgeschlagen, welche die univariate Kompaktheit und Trennung von Genen bezüglich der abhängigen Variable mit berücksichtigt. Dabei werden Gütekriterien aus der Clustering Theorie und klassischer Testtheorie verwendet, um Gene zu identifizieren, in welchen sich die Klassen deutlich in ihrer Genexpression unterscheiden. Vielversprechende Gene werden in der Regularisierung bevorzugt.
Ausserdem wird die Anwendung von Entscheidungswäldern für Wahlprognosen von unentschlossenen Wählern untersucht. Dabei wird ein Entscheidungswald in ein komplexeres Modell eingebettet, welches die mengenwertige Zielgröße von noch unentschlossenen Wählern berücksichtigt.

Im dritten Teil der Arbeit wird eine Reproduzierbarkeitsstudie vorgestellt. Untersucht wird der aktuelle Stand der Reproduzierbarkeit von Artikeln in der Fachzeitschrift PLOS ONE. Die Ergebnisse bestärken die Forderung nach besseren Standards der Methodenbeschreibung in Veröffentlichungen, insbesondere bei komplexen statistischen Modellen, sowie eine Zugänglichmachung von Quellcode.

Summary

This thesis explores alternative representations of tree ensemble methods. The general goal is to keep the desirable properties – good predictive performance and stability – while lowering the structural model complexity to a point that allows interpretation. The alternative representation of tree ensembles also open up new ways to asses model and prediction uncertainty. Additionally, this work is concerned with applications of supervised learning methods in situations of high and complex uncertainty and with the current state of reproducibility of scientific articles.

The first part of the thesis presents ways to reshape and simplify tree ensembles. Three of the contributions build directly upon the *RuleFit* approach, that decomposes tree ensembles into decision rules and uses regularized regression to find a sparse set of decision rules. Modified regularization schemes are proposed, that directly incorporate information about the rule structure – support and rule length – in the penalization. Bayesian shrinkage priors and L1-regularized regression are explored for the regularization step.
In the Bayesian model, the *horseshoe prior* is used to induce sparsity. The aforementioned information about the rule structure can be included in the resulting Bayesian hierarchical model in a straightforward way as a *rule structured prior*. Besides improved accuracy, the Bayesian framework allows a better quantification of uncertainty of both parameter estimates and derived statistics, such as the variable importance scores.
A second approach is to mimic the smoothing behaviour of the original forest method, by means of soft rules. To this end, a large number of similar rules extracted from the tree ensemble is compressed into set-valued rules. This allows to carry over the smoothing and good predictive performance from tree ensembles, while keeping the model interpretable. Clustering algorithms are used directly on the decision rules, to identify groups of rules.
Additionally, this work explores the incorporation of expert knowledge and domain knowledge, such as textbooks and guidelines, into rule ensembles. In order to account for the different degrees of uncertainty about the validity of the knowledge sources, a customized regularization scheme is presented.
Besides rule ensembles, also a framework to represent a tree ensemble by a single tree structure is presented. At each node several tests are allowed using both set-valued splitting points and multiple covariates, capturing the uncertainty about the correct splitting positions and covariate used for splitting. The model uncertainty is already taken into account during tree induction, leading to more stable and accurate tree models. The resulting tree is a special case of a tree ensemble, that mimics the behaviour of random forests, while being structurally much simpler.

The second part of this thesis contributes to applications of machine learning methods in situations of high and complex uncertainty.
An important but challenging task is the identification of genetic biomarkers for personalized medicine. To improve the variable selection consistency, we extend the L1-regression by taking the univariate properties of genes into account. To this end, measures de-

rived from clustering theory and statistical testing are incorporated into the regularization scheme. Genes that decompose into well separated and compact classes are promoted. Furthermore, the application of machine learning methods for election forecasting of undecided voters is explored. To this end, a random forest model is embedded into a larger model, that takes into account the complex and set-valued response for voters that are not decided yet. The framework allows to make use of the complex uncertainty, instead of neglecting it.

The third part of this thesis presents an empirical reproducibility study, that evaluates the state of reproducibility of articles published in the journal PLOS ONE. The study emphasizes the need of better reporting standards for complex statistical methods and the publication of source code.

Declaration of the author's specific contributions

This thesis consists of seven contributions listed in the following with the abbreviations that are used later in this work for referencing:

HR: Nalenz, M. and Villani, M. (2018). Tree ensembles with rule structured horseshoe regularization. *Annals of Applied Statistics*, 12(4):2379–2408

CRE: Nalenz, M. and Augustin, T. (2021a). Compressed rule ensemble learning. Under review for AISTATS. Preprint available under: `https://github.com/maltenlz/Malte-Nalenz/blob/main/CRE.pdf`

CRF: Nalenz, M. and Augustin, T. (2021b). Cultivated random forests: Robust decision tree learning through tree structured ensembles. Technical Report. Available under: `https://epub.ub.uni-muenchen.de/77861`

ERF: Ebner, L., Nalenz, M., ten Teije, A., van Harmelen, F., and Augustin, T. (2021). Expert rulefit: Complementing rule ensembles with expert knowledge. In *19th International Conference on Artificial Intelligence in Medicine, KR4HC Workshop*. Currently unavailable under the original address. Instead available under: `https://github.com/maltenlz/Malte-Nalenz/blob/main/ERF.pdf`

DPL: Fütterer, C., Nalenz, M., and Augustin, T. (2021). Discriminative power Lasso – incorporating discriminative power of genes into regularization-based variable selection. Technical Report. Available under: `https://epub.ub.uni-muenchen.de/77862`

UV: Kreiss, D., Nalenz, M., and Augustin, T. (2020). Undecided voters as set-valued information–machine learning approaches under complex uncertainty. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Tutorial and Workshop on Uncertainty in Machine Learning*. Available under: `https://github.com/maltenlz/Malte-Nalenz/blob/main/UV.pdf`

PLOS: Seibold, H., Czerny, S., Decke, S., Dieterle, R., Eder, T., Fohr, S., Hahn, N., Hartmann, R., Heindl, C., Kopper, P., Lepke, D., Loidl, V., Mandl, M., Musiol, S., Peter, J., Piehler, A., Rojas, E., Schmid, S., Schmidt, H., Schmoll, M., Schneider, L., To, X.-Y., Tran, V., Völker, A., Wagner, M., Wagner, J., Waize, M., Wecker, H., Yang, R., Zellner, S., and Nalenz, M. (2021). A computational reproducibility study of PLOS ONE articles featuring longitudinal data analyses. *PLOS ONE*, 16(6)

Specific Contributions In the following a detailed declaration of the authors' contributions to each of the presented articles is given. The contributions are ordered by topic.

For Nalenz and Villani (2018), Malte Nalenz developed the idea of combining rule ensemble learning with Bayesian shrinkage priors, as well as changing crucial steps in the rule extraction step and the development of the rule structured prior. Also the implementation in R, the experiments and the development of the R-package was solely due to Malte Nalenz. Mattias Villani contributed with very insightful input and discussion about the application of the Bayesian framework, setting up the sampling process and the implications and interpretation of the results. The quantification of uncertainty derived from the posterior distribution was developed in close cooperation, as well as computational considerations, such as the restricted Gibbs-sampling process. The paper is based on the master thesis by Malte Nalenz (under supervision of Mattias Villani) but was significantly changed and improved by both authors to make it more concise. Proof-reading and revision was done by both authors.

In the contribution Nalenz and Augustin (2021a), the development and implementation of the methodology was mainly by Malte Nalenz. The implementation in R and the experiments were solely carried out by Malte Nalenz. Thomas Augustin contributed with very insightful discussions, especially about the interpretation of set-valued decision rules. While the article was written by Malte Nalenz, both authors contributed with proof-reading and discussion. Also Thomas Augustin aided with making the mathematical notations and methods description more concise.

The contribution Nalenz and Augustin (2021b) was mainly written and developed by Malte Nalenz. However, the ideas were developed in close discussion and Thomas Augustin hinted at earlier work on uncertainty of splitting points, that was very relevant for the development of the article. The implementation in R and the experiments were solely carried out by Malte Nalenz. Both authors contributed to proof-reading of the paper.

Contribution Ebner et al. (2021) started as a masters thesis project of Luisa Ebner under joint supervision from Malte Nalenz, Frank van Harmelen and Thomas Augustin. The conference paper was mostly written by Luisa Ebner. Malte Nalenz wrote parts of the methods chapter, regarding regularization. All authors contributed with discussion, input and proof-reading to the paper. The development and implementation of the method, as well as the empirical experiments was due to Luisa Ebner, but in co-operation and discussion with Malte Nalenz.

For Fütterer et al. (2021) the main idea of incorporating univariate clustering quality criteria was developed jointly by Cornelia Fütterer and Malte Nalenz. Cornelia Fütterer and Malte Nalenz contributed equally to this paper. The implementation was mostly due to Cornelia Fütterer with initial input from Malte Nalenz. All authors contributed with discussion and proof-reading of the paper.

In the contribution Kreiss et al. (2020), the main idea of factorizing the distribution to make it accessible for machine learning methods was due to Dominik Kreiss, who also mainly wrote the paper, and Thomas Augustin. Malte Nalenz contributed with the application of random forests for the election data, as well as writing the chapter in the methods section on random forests. All authors contributed with proof-reading of the paper.

The idea for the reproducibility study Seibold et al. (2021) was developed by Heidi Seibold, as well as the preparations for the experiments, such as selecting and filtering the papers, contacting the authors and setting up the university course. Malte Nalenz contributed two-fold. Firstly, as a teaching assistant to carry out the experiment and assist the student groups in their reproduction process, when it was necessary and appropriate. Secondly, with writing the paper. While Heidi Seibold wrote the main part of the paper, Malte Nalenz mainly wrote the results section. Heidi Seibold and Malte Nalenz contributed with proof-reading and revising the paper.

Contents

1 Introduction

1.1 Tree Ensemble Learning and its limitations

In general, supervised learning is concerned with approximating an unknown function that maps the covariate space to the outcome. In the last decades supervised learning was a topic of major interest and myriads of different approaches have been proposed. Good overviews can be for example found in Fernandez-Delgado et al. (2014), Murphy (2012) and Hastie et al. (2017). One extremely popular supervised learning framework is ensemble learning, where multiple weak models, in the following referred to as weak learners, are combined into a stronger model. The intuition of ensemble learning is that each weak learner is able to capture a different aspect of the dataset. Combining the different 'experts' leads to a solution that takes the different possible ways to describe the relationship between covariates and outcomes into account and thus produces more stable and accurate predictions than any single model that lays all its eggs into one basket. The improvement gained from combining models into an ensemble was found to be the greatest when combining unstable models, as long as their accuracy remains higher than random guessing (Freund et al., 1999).

In general, almost any class of models can be combined into an ensemble for different learning tasks, e.g. stepwise regression (Breiman, 1996), clustering (Kiselev et al., 2017), time series forecasting (Oliveira and Torgo, 2015) and deep learning (Deng and Platt, 2014). In the context of supervised learning it is particularly popular to use decision trees [1] as weak learners. Decision trees are an interesting candidate for ensemble learning, as they are known to be unstable. Small perturbations in the training data can lead to a completely different model structure which makes them prone to over-fitting. Interestingly, in the context of ensemble learning this downside becomes a merit, as building trees on re-samples of the training data directly translates into an increase in model variance of the ensemble, which in return reduces the prediction variance.

It was often found that decision forests[2] show a remarkably strong predictive performance with little to no over-fitting (Probst and Boulesteix, 2017). The most popular tree-based ensemble approaches to date are random forests and gradient boosting. Their appeal stems

[1] The term decision tree (or sometimes abbreviated as tree) is used for both classification trees and regression trees in a supervised learning setting throughout this work.

[2] In this work we use the term decision forests is used for all ensemble frameworks that combine decision trees as weak learners, not only for random forest type models. Therefore the terms decision forest and tree ensemble are used synonymously.

from their relatively simple 'out-of-the-box' usage, robustness towards parameter choices (Bernard et al., 2009; Probst et al., 2019) and good predictive accuracy. Also they are flexible towards all kinds of tabular data, freeing the modeller from making many assumptions on the underlying relationship, such as interaction effects. Another advantage is their robustness towards extreme data situations, such as $p >> n$, extremely large datasets or datasets that contain a lot of noise covariates (Couronné et al., 2018). Random forests for example are a very popular tool for feature selection in gene expression data, where the number of covariates typically exceeds the number of samples. This has led to a widespread use of tree ensemble methods in the last decades, for example in medicine (Brajer et al., 2020), genomics (Boulesteix et al., 2012) and official statistics (Tam and Clarke, 2015).

However, a major downside is the black-box character of decision forests. As the final model is a combination of hundreds or thousands of decision trees, it is difficult to understand the inner workings of the forest. This makes it difficult to extract pattern or hypotheses from the data, that go beyond predictions. Many methods to address this caveat have been proposed, such as (conditional) Variable Importance (Strobl et al., 2008), Partial Dependence (Hastie et al., 2017) and SHAP-values (Lundberg and Lee, 2017). These methods can describe which covariates the ensemble relies on either globally, or for an individual prediction and also can interpret the marginal impact of individual covariates or interactions thereof. However, while those methods can account for the covariates used (which in many cases might be sufficient) and their marginal impact, it is still unclear *how exactly* they are used by the model. In recent years increasing scepticism was expressed about the safety, fairness and reliability of machine learning (ML) models (Barocas et al., 2017). ML was for example found to reproduce current social and racial inequalities and various kind of biases (Mehrabi et al., 2021). Another problem is that pattern found in the current data may not extend well to situations in other temporal or spacial context. In all these situations, the ability to inspect the pattern that a model is based on is key for a 'safe' application. This allows to identify pattern found by the model that may be either unethical, data artefacts or pattern that may not generalize well to future data (Caruana et al., 2015).

The lack of insight, characteristic to black box models, such as decision forests is problematic when judging the generalizability of the model. Cross validation (CV), the most common method to estimate the generalization error, is often only a weak proxy for the true generalization error (Toll et al., 2008). Potential problems that are not covered with CV are shifts in covariate distributions (Altman et al., 2009), methodological artefacts, measurement errors and more generally a pattern drift. Generalizability was for example found problematic in clinical applications, where models trained in one hospital decreased in performance for patients in other hospitals (Zech et al., 2018). Reasons for the decrease include shifts in the distribution of patient subgroups, stratified for example by sex, age or socio-economic background, different ways of measurement and artefacts due to the measurement (or the non-measurement) process (Gennatas et al., 2020). Additionally, advancements in medicine make models based on data from decades ago not extend well to the present. Therefore, without knowing the pattern on which the model is based on, it

is hard to judge, if the model will generalize well for (future) unseen data (Justice et al., 1999). While vice versa, if one understands the underlying pattern that the model relies on, it may be possible to estimate the decrease in performance that is to be expected and issue a warning if the risk of failure is high. External validation is also very important to control for the effect of potential confounder variables (Steyerberg et al., 2013).

The above arguments imply that interpretability, generalizability and robustness are strongly entangled. Additionally, scientific results are often subject to a large degree of uncertainty about the correct model and data preprocessing steps necessary to reach valid conclusions, also often referred to as researchers degrees of freedom (Hoffmann et al., 2021; Simmons et al., 2011). Widely used procedures, such as step-wise model selection often do not communicate this uncertainty to the user and give a false impression of definite results. Trying to replicate findings with different data or preprocessing steps can lead to vastly different results. Therefore, it is very important to have insight into the inner workings of the model, its robustness and stability as a way to allow for external validation (Gennatas et al., 2020). A lack in either aspect can basically invalidate any interpretation made upon the model. On the other hand, the ability to follow the models reasoning also builds trust from domain experts in the model (Buchanan and Shortliffe, 1984) and allows a contextualisation of the results (Boulesteix et al., 2019).

It was often stated that there exists a clear trade-off between interpretation and predictive performance (Kuhn and Johnson, 2013). A reformulation of this statement is that a model needs to be complex in order to generalize well. This thesis work tries to question this inverse relationship, by looking at in-between approaches that are just complex enough to generalize well, but reshape the complexity in a form that allows interpretation. The complexity is necessary to account for the uncertainty in the model building process. The guiding question of this work is if it may be possible to account for the uncertainty in a way that does not turn the model into a black-box.

1.2 Aim of this work

This cumulative dissertation explores different representations of tree ensemble methods. The goal is to preserve the upsides of ensemble learning – strong predictive performance, robustness and smooth decision boundaries – but reshape the ensemble in a form that is more accessible to human interpretation. A special emphasis is hereby on the approximation of the forest by a set of decision rules that are more interpretable than the original forest. The assumption is that the greedy learning procedure in tree ensembles produces overly complex models. By removing unnecessary complexity, it is therefore possible to simplify the model significantly and still preserve the most important pattern.

Decision forests can be interpreted as a special form of model uncertainty in decision tree learning. Typically, a large number of trees is necessary to capture the different possi-

ble ways to model the data. This work explores the possibility to 'compress' much of this complexity in a much smaller number of decision rules or even a single tree, that inherit an explicit representation of the uncertainty about the exact structure. To this end, concepts from the theory of model imprecision are borrowed to capture the rich set of tree models that might fit the data well, instead of only an optimal one. One line of work is therefore to approximate the forests behaviour as an imprecise version of a much simpler model, such as an set-valued decision tree or rule ensemble.

In a similar mindset, Bayesian regularization approaches are combined with rule ensemble learning, which enables natural ways to quantify uncertainty of both parameter estimates, as well as derived statistics, such as variable importance scores. Sampling from the posterior distribution of the rule ensemble can be interpreted as drawing different ensembles all together, thus allows inspection of the robustness and variability of the model.

Different regularization schemes – in both Bayesian regularization and frequentist L1-regularization – are explored that directly take into account the 'interpretational complexity' and trustworthiness of the terms, instead of only penalizing the magnitude of coefficients. To this end a rule structured prior is proposed that allows a direct penalization of overly complex rules. A second line of thought is to penalize terms differently, depending on its source and reliability. This opens up the inclusion and prioritization of expert knowledge into otherwise purely data driven models. The aim is to enrich the ensemble with valuable information and to build trust from domain experts.
Lastly, in the context of biomarker selection we explore regularization schemes that prioritize genes that appear as trustworthy predictors when analysed univariatly. The goal is to improve the variable selection stability and reduce over-fitting on spurious relationships.

In a wider context, the proposed methods aim at improving on the trustworthiness of ensemble models, by allowing the user to judge if the found mechanisms are reasonable – and perhaps interesting – or are likely just random fluctuations and spurious correlations. It is argued here that these aspects are highly relevant to produce reliable results, in both science and industry. Representing tree ensembles in a way, that allow a glimpse into their inner working are therefore highly desirable for sanity checks, building trust from domain experts and their application in high stake situations. Being able to follow the models 'reasoning' makes it also easier to compare and contrast new results with existing research and to derive hypotheses for further research (Boulesteix et al., 2019).

The general difficulty of reproducibility and its pitfalls are also studied in this work, to emphasise this importance of interpretable, reliable and yet accurate ensemble models.

This thesis is structured as follows: Chapter 2 introduced notations and gives an overview on existing decision tree and tree-based ensemble approaches that the contributions build upon. Also the framework of rule ensembles is introduced, as an existing alternative representation of decision forests. Chapter 3 provides an overview and a contextualisation of

the contributions made in this thesis. Additionally, for each contribution a brief summary is given, together with critical comments and an outlook of possible future research directions. In Section 3.1 alternative representations of tree ensembles are explored. Section 3.2 presents applications of machine learning methods in situations with high and complex uncertainty. Section 3.3 finishes with a contribution to the state of reproducibility in statistical research. Chapter 4 concludes with final remarks by the author.

2 Methodological Background

In supervised learning, we are given a training data set consisting of N tuples $\{(y_i, x_i)\}_{i=1}^N$ where $y \in \mathbb{R}$ or $y \in \{0, 1\}$ for regression and binary classification respectively. Y is referred to as outcome, response or label, with realisation y and y_i denoting the realisation for observation i. X denotes the vector consisting of the p random variables $X_1, \ldots, X_p$, referred to as covariates. x denotes the realisation of X and x_i denotes the observed vector of covariates for observation i. The j'th component of the covariate vector is accessed via $x^{(j)}$. In this work, if not stated otherwise, the covariates are assumed to be numeric, hence $X \in \mathbb{R}^p$ where p is the number of covariates. The goal in supervised learning is to 'learn' the unknown function $F(X) = Y$ that maps the covariate space to the outcome and to use this function for predictions. As the data is typically observed under noise and with a finite sample size N, $F(X)$ is barely an approximation of the underlying true function [1].

2.1 Decision Trees

This section provides a non-exhaustive overview on decision tree learning. The focus is on areas and concepts that are relevant within the presented work. More extensive surveys on the topic of classic decision trees can be found in (Rokach and Maimon, 2005; Strobl et al., 2009).

Decision trees are non-parametric, graphical models, that recursively partition the covariate space X into hyper-rectangles $\mathcal{S} \in \mathbb{R}^p$, and assign piecewise constant predictions to all values that fall into each hyper-rectangle.[2] Starting at the *root node* where all data points are present, at each *inner node* a decision is made based on a splitting rule $x^{(v)} \in s$, where $s \in \mathbb{R}$ is a subspace in covariate v, that is used for the comparison. Assuming numeric covariates, s has the form of an interval $(-\infty, t]$ or $[t, \infty)$ where $t \in \mathbb{R}$ is the *splitpoint*. The splitting function ϕ can be expressed via

$$\phi(x, v, t) = I(x^{(v)} \leq t) \quad \text{or} \quad \phi(x, v, t) = I(x^{(v)} \geq t) = 1 - I(x^{(v)} < t), \tag{2.1}$$

depending on the direction that is encoded in the tree and $I \in \{0, 1\}$ is the indicator function. All observations that fulfil the splitting rule are moved to the left child node or

[1] If such a function even exists.

[2] Even though the partitions can be p-dimensional, due to the selective manner of decision trees, they usually live in much lower dimensions.

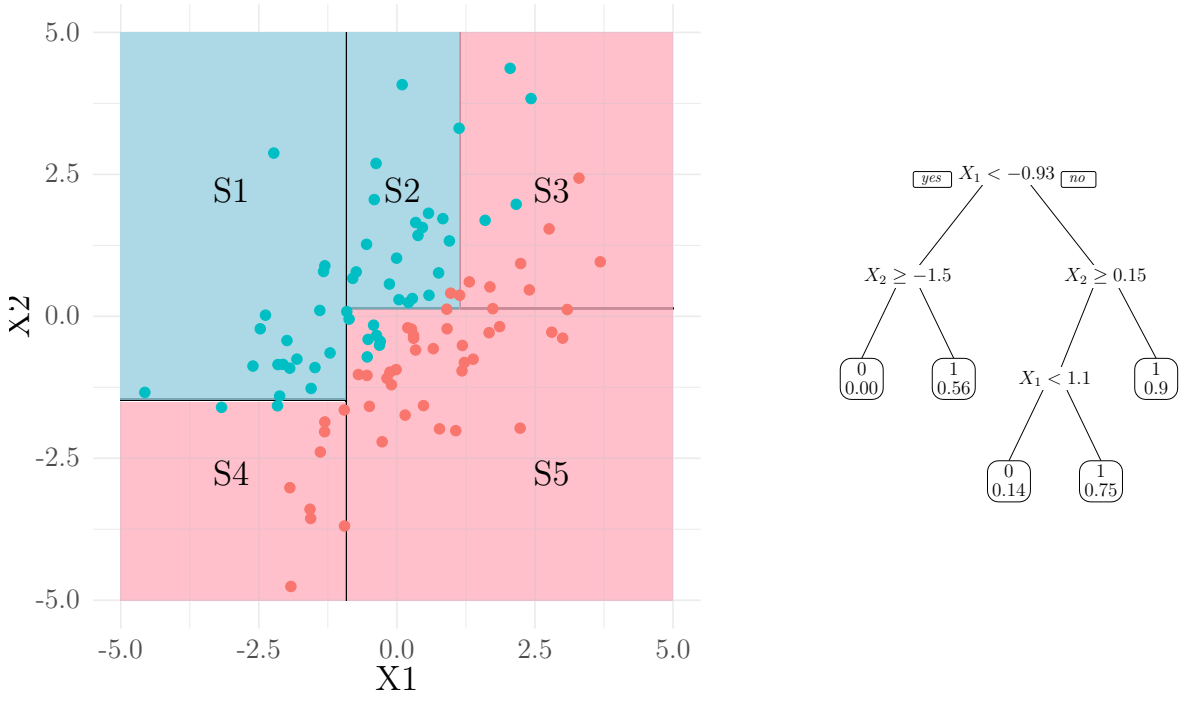


Figure 2.1: Representation of a tree for binary classification (Y=1 as red dots and Y=0 as blue dots) with two covariates X_1 and X_2. Left: Representation of the subspaces $\mathcal{S}$ in feature space, with colors indicating the predicted class. Right: Representation of the tree as a graph, where each node can be written as a decision rule $r(x)$. Inside the leaf-nodes (rectangles) the upper number specifies the predicted class, the bottom number the predicted probability for the predicted class. At each inner node the splitting rule is shown.

to the right child node otherwise [3]. Here we will represent the splitting rule or condition as the triplet $c = (z, v, t)$, where z specifies the leafnode on which path the splitting rule is a part of [4].
The recursive partitioning is repeated until a *leaf node* is reached and the assigned (constant) *leaf value* is returned as a prediction for all observations reaching this leaf node[5]. Each leaf node specifies a partition $\mathcal{S}_h$. $\mathcal{S}_h$ can be seen as a decision rule that combines

[3] Assuming binary decision trees. Other popular classes of decision trees are oblique trees (Murthy et al., 1994; Carreira-Perpinán and Tavallali, 2018; Bennett et al., 2000), where hyperplanes are used as splitting condition and multi-way trees, with more than two childnodes (Khandagale et al., 2020; Breiman et al., 1984).
[4] Many splitting rules will therefore be part of several paths.
[5] An alternative are model-trees that use a parametric model in each leaf node to make a prediction, instead of the constant value (Zeileis et al., 2008; Seibold et al., 2016; Strobl et al., 2015).

the splitting rules along its path with the logical $\wedge$ and can be written as,

$$I(x \in \mathcal{S}_h) = r_h(x) = \prod_{a:z_a=h} \phi(x, v_a, t_a) \quad r_h \in \{0, 1\}. \tag{2.2}$$

Figure 2.1 shows the different representations of the same decision tree in feature space and as a graph.
The output of the whole decision tree can be written as the sum of decision rules leading to leafnodes , multiplied with the corresponding prediction value μ_h,

$$\hat{P}(Y|X = x) = \sum_{h \in \mathcal{M}} I(x \in \mathcal{S}_h)\mu_h = \sum_{h \in \mathcal{M}} r_h(x)\mu_h, \tag{2.3}$$

where $\mathcal{M}$ is the index set of leafnodes.
One concept to deal with uncertainty in the measurement of covariates, or when using logistic decision functions in the inner nodes, is to use fractional observations (Ripley, 1996; Quinlan, 1993). The idea is to move observations to both childnodes, if a decision cannot be made with certainty. This approach can also be applied if missing values are encountered, by splitting up the observation on both child nodes and thus abstaining from making any definite decision. A final prediction for such fractional observations can be obtained by averaging the prediction values of all leaf nodes, weighted by the fraction (or probability) of the observation that is found in each leaf node.

Decision trees are very flexible models and can capture non-linear relationships. Theoretically, decision trees can approximate any given function $F(X)$. However linear relationships require very deep trees and will still ultimately remain non-smooth.
Through their hierarchical structure, decision trees are able to detect interactions between covariates, without any prior specifications. Decision trees can also combine categorical and numerical features in a natural way. Several ways to deal with missing values in the covariates have been proposed, which makes them applicable in many data situations (Quinlan, 1993).
Typically, at each node only a single covariate is used for splitting, making the graphical representation relatively easy to interpret. The structure of binary decision trees and its typical representation as a graph of splitting rules deliver an easy human comprehensibility (Podgorelec et al., 2002), as shown in Figure 2.1. To deduce the characteristics of the subgroups, specified by the leaf nodes of a tree, one has to simply start at the top of the tree and successively move down the tree, which somewhat resembles human reasoning. This properties make decision trees popular in the medical domain, as the practitioner can explain what a prediction is based on.

Tree Induction and the Problem of Instability. While decision trees have a number of attractive properties, they also have major shortcomings. One problem is the lack of smoothness found in trees, as typically piecewise constant functions are used. While being arguably more problematic for regression, as the underlying function is expected to be

smooth here, this can also be problematic in the case of classification, as already minor changes in the covariate values can lead to a completely different prediction.

Another well known drawback of decision trees is their instability (Breiman, 1996; Philipp et al., 2016; Strobl et al., 2009). Instability stems mostly from the tree induction process. Learning globally optimal decision trees is typically computationally infeasible. For this reason a variety of (greedy) decision tree induction methods have been proposed in the literature that instead use local quality criteria to successively partition the training data. At each node, beginning at the root node, a search is performed over all possible splitting points. For each possible split the quality in the two child nodes is evaluated and the split taken that promises the biggest improvement in purity. In classification the most popular quality criteria are gini gain (Breiman et al., 1984) and entropy (Quinlan, 1993). Similarly, for regression trees the mean squared error can be used as a measure of purity. The most common splitting criteria are biased towards selecting covariates with many unique values and missing values. Unbiased selection criteria, that decouple variable selection and split-point selection, have been proposed in (Strobl et al., 2007a; Hothorn et al., 2006). Once the locally optimal splitting covariate and point is found, the training samples are moved to the two child nodes and the procedure is repeated until no partitioning improves the impurity further or some other stopping criteria, such as maximum tree depth, is reached [6]. Instability can be traced back to the all-in decision at each node (Mirzamomen and Kangavari, 2017). Even though two splitting rules can be close in terms of expected gain, they can lead to very different data partitionings. Due to the recursive process, the choice will greatly impact the following selections of splitting rules and thus the structure of the tree. This leads to a large degree of structural instability [7], with respect to small changes in the training data. Removing or adding only a small fraction of observations can lead to completely different overall tree structures.

As an alternative to the all-in decisions at each node, option trees (Buntine, 1992) and alternating decision trees (Freund and Mason, 1999; Frank et al., 2015) have been proposed. Instead of using only the single best split at each node multiple splitting rules are allowed, capturing the uncertainty involved with the decision. This essentially leads to a set of trees, that share large subtrees. Following each subtree leads to an exponentially growing number of different trees that can be averaged to get a final prediction.

Another approach to robustify decision trees is through the use of imprecise probabilities (Corani et al., 2014; Bernard, 2005). Instead of evaluating only the observed data, virtual observations from both classes are added to the 'true' observations in each leaf,

[6] A common alternative is to grow overly large trees and 'prune' away subtrees that contribute little (Mingers, 1989; Patil et al., 2010).

[7] Structural instability refers to the graphical representation of the tree. Different trees may lead to the same data partitioning through different splitting rules, making interpretations based on a single tree highly suspicious. Model instability refers to the Variance of the model $F(x)$ when learned on slightly different training samples.

leading to a credal set of possible target distributions for which lower and upper bounds of entropy can be calculated (Mantas and Abellán, 2014; Abellan and Moral, 2003). This approach takes into account the effect of small perturbations in the training data and leads to more robust trees that perform better in the presence of label errors and noise.

2.2 Ensemble Learning as Generalized Additive Model

Ensemble learning can be used to address the problems of poor generalization due to overfitting and instability of greedily learned decision trees. Combining several decision trees built on different samples of the training data (Friedman et al., 2003), called weak learners, generally leads to more accurate and robust predictions. The benefits of ensemble learning have also been attributed to its smoothing behaviour (Bühlmann and Yu, 2002; Bühlmann, 2012): Averaging over several piecewise constant functions leads to an overall smoother decision boundary [8]. The sacrifice to be made, when adopting the ensemble approach, is the immediate loss in structural interpretability of the individual decision trees.
From a statistical point of view, most ensemble methods can be framed as generalized additive models (Dietterich, 2002; Hastie and Tibshirani, 2017). For example an ensemble regressor may be written as a linear combination of M weak learners f_l

$$F(x) = \alpha_0 + \sum_{l=1}^{M} \alpha_l f_l(x), \tag{2.4}$$

where α_0 is an intercept term and $\alpha_l, l = 1, ..., M$ are weights. In the context of tree ensemble methods typically $\alpha_l \in [0, 1], \sum_{l=1}^{M} \alpha_l = 1$ is used, which is the weighted mean of the individual predictions. Also more sophisticated models can be used to learn the weights via 'stacking' (Zhou, 2021). This work will focus on linear real weights $\alpha_l \in \mathbb{R}$ with no further restrictions.

Interestingly, the ensemble $F(x)$ that combines several weak learners can represent much more complex decision boundaries than any individual learner $f(x)$, also known as the representational problem (Dietterich, 2002). In the context of decision trees and decision rules this means that ensembles of decision trees can represent (almost) smooth and other complex decision boundaries, whereas single (binary) trees can only represent axis parallel hyper-rectangles.

Obviously, combining different weak learners can only be effective, if the weak learners capture different hypotheses about the data. Therefore when learning an ensemble, diversity among the learners needs to be promoted. The two most dominant approaches, boosting and random forests, encourage diversity through different importance sampling schemes (Friedman et al., 2003). These will be discussed in the following.

[8] that is ultimately still non-smooth in the case of a finite number of weak learners.

2.3 Random Forests

Random forests combine decorrelated, independently grown, decision trees. By increasing the variance between the individual trees the variance of the combined classifier will decrease, leading on average to lower generalization errors. Decorrelation is achieved in random forests by inducing randomness in the tree building process through simultaneously applying two techniques: Bagging (Breiman, 1996) and random subspaces (Ho, 1998). This leads to a decorrelation between the individual trees, or in other words, the trees become experts of different aspects of the data.

In bagging, each tree f_l is build on an independent bootstrap sample (Efron, 1982), leading to different datasets presented to each learner. While the original random forest uses bootstrapping, it was argued in (Scornet, 2017; Bühlmann and Yu, 2002; Boulesteix et al., 2012) that using subsampling (drawing without replacement) instead of bootstrapping does not decrease the performance of random forests. On the other hand, using subsampling reduces bias in variable importance measures (Strobl et al., 2007b) and allows to derive theoretical properties of random forests such as consistency and asymptotic behaviour (Biau and Scornet, 2016; Bernard et al., 2009). Resampling (using either bootstrap samples or subsampling) was shown to greatly improve the predictive performance of unstable classifiers, such as decision trees. One important property of bagging is that it produces smoother decision boundaries which leads to a better generalization (Bühlmann and Yu, 2002; Breiman, 2001; Bühlmann, 2012).

As a second source of randomness, random forests use the random subspaces method (Ho, 1998). At each node the next split is restricted to only a random subset of the covariates. This decorrelates the trees further. The fraction of features to be used is also considered the most important tuning parameter in random forests.
Predictions in random forests can be obtained by averaging over the individual predictions

$$F_{RF}(x) = \frac{1}{M}(f_1(x, \Theta_1) + ... + f_M(x, \Theta_M)) \tag{2.5}$$

where $f_1, .., f_M$ are the trees built on the independent re-samples with randomized feature subsets at each node, denoted as $\Theta_1, ..., \Theta_M$. In the original formulation of random forests trees are grown until purity, but recently random forests have also been shown to work well with more shallow trees (Duroux and Scornet, 2018), which is beneficial from a computational and an interpretational point of view.

Various extensions have been proposed to induce other sources of randomness and thus decorrelate the trees further. This includes sampling of the splitting points (Geurts et al., 2006) and using transformations such as PCA (Rodriguez et al., 2006) or random projections on the input data (Dasgupta and Freund, 2008).

2.4 Gradient Boosting

Similar to random forests, boosting can be framed as building a sequence of trees on importance samples of the training data (Friedman et al., 2003). Instead of building the trees independently, gradient boosting (Friedman, 2001) uses a stagewise approach. After initializing with a first guess such as $F_0 = E(Y)$, gradient boosting proceeds to iteratively calculate the error gradient of the current ensemble via

$$u(x) = E_y \left[\frac{\partial L(\mathbf{y}, F(x))}{\partial F(x)} \Big| x \right]_{F(x)=F_{m-1}(x)}, \tag{2.6}$$

where L as an appropriate loss function evaluated at the current ensemble $F_{m-1}(x) = \sum_{l=0}^{m-1} \beta_l f_l(x)$ and fits the next learner f_m (e.g. CART) jointly with a linear weight β_m to the negative error gradient as the 'pseudo-responses' $\tilde{y} = -u(x)$ instead of the original response (Bühlmann and Hothorn, 2007). The next learner is added, with a shrinkage factor ν to the current ensemble $F_m(x) = F_{m-1}(x) + \nu\beta_m f_m(x)$ and the process repeated until some stopping criteria is reached. As a greedy procedure, the previous ensemble is fixed and remains unchanged in later iterations, in contrast to step-wise approaches. An extension to the gradient boosting algorithm is to also induce randomness, by only using a subsample to fit each new decision tree $f_m(x)$ (Friedman, 2002) and random feature subsets. This *stochastic gradient boosting* was shown to lead to higher accuracy ensembles, due to a decorrelation between the individual learners, most notably for smaller datasets. Subsampling and shrinkage can be also viewed as a form of regularization (Bentéjac et al., 2021). An additional penalty term was included recently in the popular eXtreme gradient boosting (XGBoost) model (Chen and Guestrin, 2016) that directly penalizes the tree complexity for each newly introduced base learner. As a different form of regularization it was proposed to randomly drop-out previously trained trees from the ensemble in order to enhance decorrelation and combat over-specialisation of trees (Vinayak and Gilad-Bachrach, 2015).

The gradient boosting algorithm is very popular due to its flexibility and general formulation. A large class of base learners can be employed for f_m and any differentiable function can be used as loss function L in equation 2.6, making it applicable in many domains e.g. image processing (Feilke et al., 2016) or genomics (Ogutu et al., 2011). Important choices – besides decision trees with log-loss – are the exponential loss for binary classification which leads to the adaBoost algorithm (Bühlmann and Hothorn, 2007; Freund et al., 1996; Schapire, 2003) or to use splines or generalized linear models as base learners (Bühlmann and Hothorn, 2007; Bühlmann, 2006).

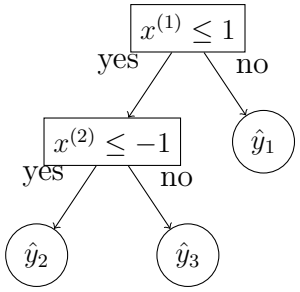
2.5 Rule Ensemble Methods

Besides decision trees, there exists a steady tradition of ensemble learning that uses decision rules as weak learners. One major advantage of decision rules is that they are composed of a conjunction of simple if-else statements and are therefore highly interpretable for humans. Another advantage of decision rules is that as long as the set of rules is small, it is easy to employ them offline. One example for this are clinical guidelines, where the practitioner performs several tests and can make a prediction based on the result of decision rules. The goal in rule ensemble learning is therefore to produce as small sets as possible and reduce the amount of overlap between rules, if interpretability is the goal.

Learning globally optimal decision rules is an NP-hard problem. Several (greedy) approaches to rule ensemble learning have been proposed, including divide and conquer approaches (Cohen, 1995; Fürnkranz, 1999; Weiss and Indurkhya, 2000). Another very effective line of research is to combine decision rules with boosting (Dembczyński et al., 2010; Schapire, 1999). Boosted decision rule ensemble were shown to posses great predictive performance, but a downside is that the ensembles can be quite large and the rules overlapping, making them again more or less a black-box model.
Learning decision rules in a linear ensemble jointly with the weights and linear terms was introduced in (Wei et al., 2019; Jawanpuria et al., 2011).

An alternative approach is to extract candidate rules from learned tree ensembles. RuleFit (Friedman and Popescu, 2008) proposes to not learn the decision rules directly, but instead take a three-step procedure.
First, a tree ensemble is generated. RuleFit uses gradient boosted trees, but also different types of trees and ensemble approaches such as random forests (Nalenz and Villani, 2018) and conditional random forests (Fokkema, 2020; Hothorn et al., 2006) can be used to generate the decision rules.
As a second step, each tree in the forest is decomposed into their defining decision rules, by harvesting all paths to each node. Both inner nodes and leaf nodes are harvested, to allow the ensemble to use simpler rules, when possible. The decomposition into the defining splitting rules is shown in Figure 2.2. A total of 4 rules can be extracted from this tree, such as $r_3(x) = I(x^{(1)} \leq 1) \cdot I(x^{(2)} \leq -1)$.
In the third step, the decision rules $r_h(x)$ are transformed to dummy covariates using the product of splitting functions ϕ (cf. Section 2.1, Equation 2.2) and included, together with linear terms, in a regularized regression model. Additionally, some form of cleansing should be performed beforehand. For example, as each inner node is the union of its two child nodes, the dummy terms will be linearly dependent. Therefore, only one of each pair of child nodes should be kept. This aspect is perhaps under-appreciated in the literature and rarely discussed (Nalenz and Villani, 2018), but might have considerable effect on the solution.

The rules extracted from the tree ensembles will show a high degree of redundancy, as



$\phi(c)$	z	v	t
$I(x^{(1)} \leq 1)$	1	1	1
$I(x^{(1)} > 1)$	2	1	1
$I(x^{(1)} \leq 1)$	3	1	1
$I(x^{(2)} \leq -1)$	3	2	-1
$I(x^{(1)} \leq 1)$	4	1	1
$I(x^{(2)} > -1)$	4	2	-1

Figure 2.2: Left: Binary decision tree with 3 leafs, 1 internal node and the root node. Right: Further decomposition of the decision rules into the elementary conditions. Multiple conditions per rule are combined with the logical AND.

well as contain uninformative rules. Variable selection and regularization are therefore a necessity. Nodeharvest (Meinshausen, 2010) solves this by a regularized linear program, leading to relatively sparse solutions. RuleFit combines decision rules extracted from the gradient boosted decision trees together with linear terms in the L1-regularized final model,

$$\{\alpha^*, \beta^*, \beta_0^*\} = \underset{\beta_0,\beta,\alpha}{\arg\min} \left[L(y, F(x, \beta_0, \beta, \alpha)) + \lambda \left(\sum_{j=1}^{p} |\beta_j| + \sum_{h=1}^{H} |\alpha_h| \right) \right], \tag{2.7}$$

with

$$F(x) = \sigma(\beta_0 + \sum_{j=1}^{p} \beta_j x^{(j)} + \sum_{h=1}^{H} \alpha_h r_h(x)). \tag{2.8}$$

Usually, all covariates are scaled before applying regularization techniques, as otherwise features with a lower scale will be penalized more heavily. This can easily be seen by the penalty for a covariate that is rescaled (given the same effect) as $\lambda|\beta^*| = c \cdot \lambda|\beta|$. RuleFit chooses to not scale the decision rules, which puts additional penalty on rules with a low support. This implies that rules that cover close to half of all observations will be penalized the least, as those will have the highest scale. However, one potential downside of this heuristic is that it penalizes a low support, but does not directly penalizes unnecessary conditions (Nalenz and Villani, 2018).
The output of RuleFit (and most rule ensemble models) is a list of rules, together with their coefficients, which allows to order them by their effect size $|\beta|$.

Lastly, an interesting approach to rule ensemble learning was proposed recently in SIRUS (Bénard et al., 2021). As a first step, SIRUS uses a quantile transformation to reduce the number of unique values per covariate, that can be used for splitting. Then a random forest is trained on the transformed dataset. Due to the low number of unique values, many paths to nodes will be shared across the forest and SIRUS simply aggregates the extracted rules and uses the k most frequent for prediction. This approach avoids the linear weighting step, which leads to more stable and interpretable results.

3 About the contributing material: Relations, summaries and outlooks

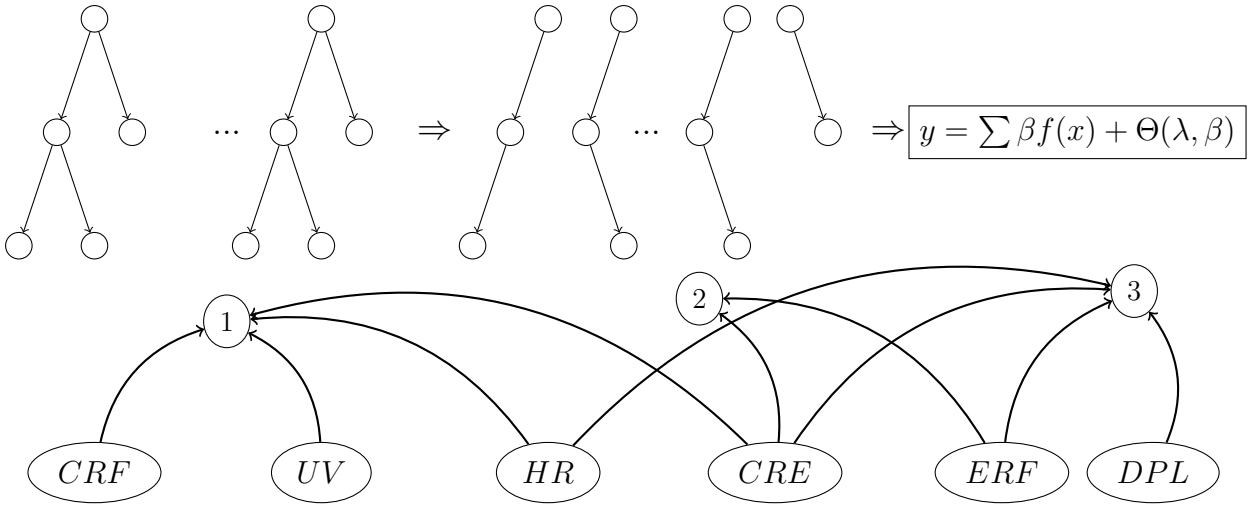


Figure 3.1: Schematic graph of the contributions topics, aligned on the three step procedure of the RuleFit approach where (1) stands for ensemble learning, (2) for the extraction, gathering and processing of decision rules and (3) for the regularized regression. Also shown are the contributions with the following abbreviations. CRF: Cultivated Random Forests, UV: Undecided Voters, HR: horserule, CRE: Compressed Rule Ensembles, ERF: Expert RuleFit, DPL: Discriminative Power Lasso. More details can be found in the Declaration of Contributions (cf. page XI). A connection means that the contribution was concerned with the topic. Not shown in the graph is the reproducibility study PLOS.

This cumulative dissertation explores the topic of representing tree ensembles by means of simpler models, with a special focus on the RuleFit approach. Figure 3.1 positions the contributions along the three steps of the RuleFit procedure, (1) general ensemble learning, (2) decision rule extraction and processing and (3) the regularized linear model. The contributions HR, CRE, ERF are directly within the field of rule ensemble learning while the other works contribute directly to one specific area. Not shown in Figure 3.1 is the reproducibility study PLOS, which is not so much concerned with a specific method but with the general state of reproducibility in statistical research. In the following the authors

contributions are summarized and the main findings and results, together with a critical reflection and future directions, presented. The contributions are divided in three parts. The contributions in the first part (Section 3.1) are concerned with alternative representations of tree ensembles. The second part (Section 3.2) consists of applications of machine learning methods in situations of high or complex uncertainty. The third part (Section 3.3) presents the contribution PLOS, that analyses the current state of reproducibility.

3.1 Alternative Representations of Tree Ensembles

3.1.1 Bayesian Rule Ensemble Learning

Nalenz, M. and Villani, M. (2018). Tree ensembles with rule structured horseshoe regularization. *Annals of Applied Statistics*, 12(4):2379–2408.
Code: `horserule` R-package on CRAN.

In this project we propose two alterations to the RuleFit model described in Section 2.7. The first alteration is based on the known over-shrinkage effect in L1-regularized regression (Zhao and Yu, 2006). With increasing number of noise covariates or redundant covariates, a higher penalty parameter λ needs to be selected for sufficient shrinkage. At the same time, a higher shrinkage will also shrink the relevant predictors towards zero, which the model counteracts by taking in correlated predictors to substitute for the overshrinkage. This behaviour is especially undesirable, when the goal is to build an interpretable model, as it will lead to larger model sizes and jeopardizes the validity of the interpretation. Also, in the context of rule ensembles, the number of unimportant predictors will always be high (cf. Section 2.5).
Instead, we propose to use a modified version of the horseshoe prior (Carvalho et al., 2009, 2010) to find a sparse rule set. Horseshoe priors belong to the class of global-local shrinkage priors (Bhadra et al., 2019). The standard horseshoe regression can be expressed as the hierarchical Bayesian model

$$\begin{aligned} y|\mathbf{X}, \boldsymbol{\beta}, \sigma^2 &\sim \mathcal{N}_n(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}_n), \\ \beta_j|\lambda_j, \tau^2, \sigma^2 &\sim \mathcal{N}(0, \lambda_j\tau^2\sigma^2), \\ \sigma^2 &\sim \sigma^{-2}d\sigma^2, \\ \lambda_j &\sim \mathcal{C}^+(0,1), \\ \tau &\sim \mathcal{C}^+(0,1), \end{aligned} \tag{3.1}$$

where $\mathcal{C}^+(0,1)$ denotes the standard half-Cauchy distribution. The main advantage is that individual predictors can still become large, through their local scale parameters λ_j, even when the global shrinkage τ becomes high. This counteracts the over-shrinkage and leaves important predictors virtually untouched. As the coefficients are either included in full size, or shrunk towards zero, the horseshoe prior can be seen as a continuous approximation of the behaviour of a well specified discrete-mixture model (Carvalho et al., 2009). In the

context of rule ensembles this leads to sparse solutions, with only a few rules remaining in the model, which are allowed to have high β coefficients. As in the standard horseshoe regression all predictors have the same prior inclusion probability, we extend the model by including the *rule structure*. This is done by setting the local scale prior to

$$\lambda_j \sim \mathcal{C}^+(0, A_j),$$

with

$$A_j = \frac{(2 \cdot \min(1 - s(r_j), s(r_j)))^\mu}{(l(r_j))^\eta}, \tag{3.2}$$

where $l(r_j)$ denotes the length of rule j defined as its number of conditions and s is the support of the rule. This expresses the prior belief that complicated and very specific rules are unlikely to reflect any true relationships. The hyper-parameters η and μ control the strength of this rule structured prior. Through the heavy tails of the half-Cauchy distribution on λ, the prior can still be overwhelmed by the likelihood, if a rule fits the data very well. Sampling from the posterior distribution is done using a more efficient Gibbs-Sampling scheme proposed in (Makalic and Schmidt, 2015).

Building upon the benefits of the horseshoe regularization, as a second alteration we propose to use trees generated by both random forests and gradient boosting. This will naturally lead to an overall larger number of rules, which the horseshoe regularization is capable to deal with. The advantage is that random forests and gradient boosting will find quite different rules and including both increases the chance of finding good ones.

Using both simulated data and regression benchmark data we show that the aforementioned changes greatly increase the predictive performance and the recovery of true linear signals (if present). Additional experiments show, that the usage of the horseshoe prior is the most influential change, whereas using both boosting and random forests to generate the decision rules leads to a smaller, yet still notable improvement.

Additionally, the Bayesian framework allows a quantification of the uncertainty that is associated with the coefficient vector β. As we draw samples from the posterior distribution of the coefficients $P(\beta|\cdot, X, Y)$, we also obtain the posterior distribution for all derived statistics. Following (Friedman and Popescu, 2008), the importance of a decision rule or linear terms can be calculated as

$$\text{Imp}(r_l) = |\beta_l| \, \text{sd}(r_l),$$

where sd denotes the standard deviation. Variable importance scores are calculated by summing up the importance of all rules that involve covariate j, discounted by the number of covariates involved in the rule. Therefore the variable importance of covariate j can be written as,

$$\text{VarImp}(j) = \text{Imp}(x^{(j)}) + \sum_{l: j \in Q_l} \text{Imp}(r_l) / \left|Q_l\right|,$$

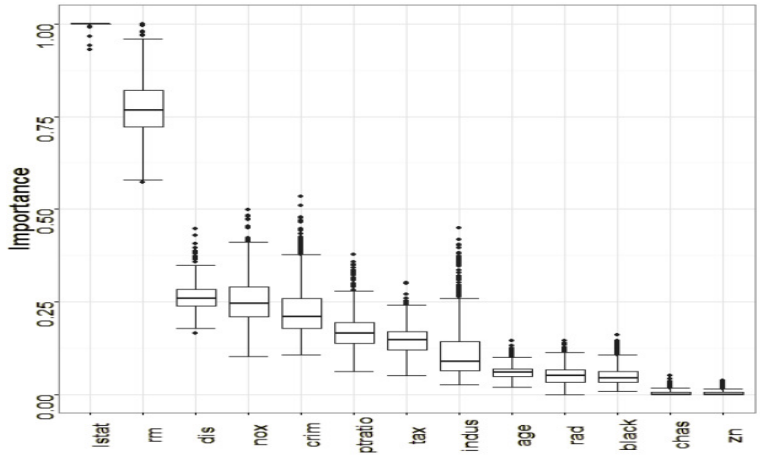


Figure 3.2: Posterior distribution of the VarImp for the 13 covariates of the Boston Housing dataset, which analyses the impact of different factors on the average housing prices in areas of Boston. Graph taken from Nalenz and Villani (2018).

where Q_l is the set of covariates involved in rule l and $\text{Imp}(x^{(j)})$ is the importance of the linear term (Friedman and Popescu, 2008). As both of these measures are functions of β, we can derive posterior distributions, that allows to quantify the uncertainty of importance scores. Figure 3.2 shows the posterior distribution of VarImp for the Boston housing data. While some covariates have very narrow credible intervals, for others the uncertainty about the importance is quite high. This quantification of uncertainty for measures such as VarImp is interesting, as it can be used to guide further research hypotheses towards the most promising covariates. Also it gives a measure of robustness for any conclusions that are based on the VarImp.

Comments and Outlook. Even though the classification worked well on the dataset that we tried in our experiments, we later found that the convergence properties of the Gibbs-Sampling scheme can be poor. This is connected to the problem of separability in combination with heavy tailed priors on the scales of regression coefficients (Ghosh et al., 2018), that allows individual coefficients to become unreasonably large. To deal with this problem it was recently proposed to use a more informative prior on the global scale parameter τ, to avoid extreme behaviours (Piironen and Vehtari, 2017). Another remedy could be to use distributions with less heavy tails for τ. A second approach is the regularized horseshoe prior, which puts a second level of penalty on values that are deem unreasonably high (Piironen and Vehtari, 2017). It would be interesting to adopt these methods for the classification setting.

The current implementation uses a rather efficient Gibbs sampling scheme based on the work of Makalic and Schmidt (2015). However, for large datasets the current implementation is still too slow. Recently proposed sampling schemes promise even further perfor-

mance increase for the horseshoe regression models, such as the elliptical slice sampling (Hahn et al., 2019) or the recently proposed GPU accelerated sampling (Terenin et al., 2019) which promise additional speed ups for large datasets.

The idea of applying Bayesian inference to re-estimate coefficients of an otherwise fixed model to produce uncertainty estimates is also interesting for other model classes such as deep learning (Klein et al., 2021). While fully Bayesian ensemble methods such as BART (Hill et al., 2020; Chipman et al., 2010) and DART (Linero and Yang, 2018; Linero, 2018) are very powerful, they can also be computationally costly for big datasets. Using a two-step approach of estimating the structure and applying a Bayesian regression approach on top can be a good compromise for large datasets.

Lastly, it would be interesting to sample the split points of otherwise fixed rules, by specifying a prior distribution for split points. The idea is, that this way the model does not need to take in very similar rules to express the uncertainty about the split point, but instead samples from the posterior distribution of split points. This idea is similar to the approaches taken in CRE and CRF, discussed in the following.

3.1.2 Compressed Rule Ensembles

Nalenz, M. and Augustin, T. (2021a). Compressed rule ensemble learning. Under review for AISTATS. Preprint available under: `https://github.com/maltenlz/Malte-Nalenz/blob/main/CRE.pdf`.

Even though the RuleFit approach provides much simpler models compared to tree ensembles, interpretability is still often suboptimal due to two reasons: (1) The final output from RuleFit often still contains a relatively high number of rules that can be fairly complex (involve a high number of conditions). This is partly due to the linear combination and the over-shrinkage effect of the L1-regularized regression, as argued in section 3.1.1. (2) Secondly, small changes in the data can lead to fairly different rule models. While this may not be a problem in terms of predictive performance, it raises questions whether the interpretation of the final rule set is valid (Bénard et al., 2021).

In Bühlmann and Yu (2002) the authors show that the good performance of forest methods can be attributed to the smoothing of the hard-thresholding rules. This is achieved by averaging over similar trees with different splitting points. When reducing the set of rules in the rule ensemble approach to a small number of non-smooth rules, it stands to reason that the smoothing behaviour is lost. To maintain good predictive accuracy, the rule-set therefore needs to become larger and contain similar and overlapping rules, in order to fit smooth decision boundaries, which makes interpretation difficult.

To resolve this dilemma we propose to compress similar rules into soft rules with set-

valued splitting points, called ensemble rules. This allows to carry over the smoothing behaviour, while being relatively easy to interpret. To find similar rules, we perform univariate k-means clustering on the split points found in the original forest. Given $T^{(j)}$, the set of all split-points from conditions that involve covariate j, the clusterlabels l are chosen to minimize the cluster-criteria

$$C(k, \mu, T^{(j)}) = \sum_{l=1}^{k} \sum_{\{z: g_z^{(j)} = l, t_z \in T^{(j)}\}} (t_z - \mu_l)^2, \tag{3.3}$$

where k is a pre-specified number of clusters, $g^{(j)}$ is the vector of clusterlabels and μ the vector of mean values of the clusters. Using the cluster solution from Equation 3.3, the split-points in each group $T_l^{(j)}$ are compressed into soft *ensemble conditions* via

$$\Phi(x, j, l) = |T_l^{(j)}|^{-1} \sum_{t \in T_l^{(j)}} \phi(x, j, t). \tag{3.4}$$

The averaging over several discrete outputs from the splitting function ϕ turns the binary decision into a soft decision, that gives a transition interval depending on $T_l^{(j)}$. Thus, per covariate only k ensemble conditions remain that capture the most central aspects of the forest method and are relatively robust towards changes in the training data.
By using groups of split points we capture the model variance and uncertainty of the tree ensemble about the exact position of the optimal split-point. At the same time the rules in Equation 3.4 are still simple to interpret, as they involve only one covariate at a time and involve only a neighbourhood of split-points.
The conditions in each original rule are replaced with their corresponding ensemble condition. The whole rule output is calculated by the product over all ensemble conditions involved, leading to an overall soft rule output, denoted as $\mathcal{R}$.
The *Compressed Rule Ensemble (CRE)* uses a linear combination of ensemble rules $\mathcal{R}$. A modified L1-regularized regression model is applied to achieve a sparse solution. To put additional penalty on overly complex rules with many conditions, they are rescaled, using

$$\mathcal{R}_h^*(x) = \frac{\mathcal{R}_h(x)}{l(\mathcal{R}_h)^\eta}, \quad \eta > 0, \tag{3.5}$$

where $l(R)$ is the rule length and η a parameter controlling the amount of extra penalty. This is inspired by the rule structured prior from Section 3.1.1 and can be seen as a variant of the adaptive Lasso (Zou, 2006), which is also applied in Section 3.1.3 and Section 3.2.1.

Empirically, compressed rule ensembles are on average smaller, while achieving higher accuracies, compared to competing rule ensemble methods. This is due to the smooth decision boundaries introduced by the ensemble conditions (also discussed in Section 3.1.4). An additional advantage of the CRE approach is that it provides a notion of prediction stability. In the original RuleFit an observation can be close to several split-points and

already small changes in the input space can lead to a significant change in the predictive distribution. In CRE each ensemble rule provides a transition interval and small changes will not have an unreasonable high impact on the prediction. At a first glance a set of split-points or a transition interval makes interpretation more complex. However, in high stake situations it is almost a necessity to know how reliable an active rule is and how close the predictions are, which is a blind spot of hard thresholding rule ensembles. With that CRE provides a better way to quantify the uncertainty in both the model structure and predictions.

Comments and Outlook. One interesting future work direction would be to provide better visualisation tools for CRE. In many cases it could be possible to back transform the final rule output[1] into a small number of very shallow (soft) decision trees.

The framework of rule compression is also interesting for the more general setting of clustering and summarising of forest methods. Through the clustered nature of random forests, it could be possible to approximate the decision forest by means of a small number of compressed trees, granting a glimpse on the inner workings of the black-box. Using the same logic, the CRE framework could be interesting for interpreting predictions. By gathering all active leaf nodes for a given prediction and using the ensemble compression approach, most of the model variance could be expressed by a relatively small number of ensemble rules.

Lastly, a downside of the current approach is, that it ignores the depth of the rules, from where the split points are extracted. If an effect is very specific, and only relevant in interaction with other covariates, it might be blurred, when looking only at the univariate distributions of split points. The key question that arises is, in what situations and how much compression is reasonable without tempering with the signal, and when perhaps a multivariate clustering approach would be more appropriate.

[1]The rule output is a list of ensemble rules with corresponding coefficients.

3.1.3 Incorporating Expert Knowledge into Rule Ensembles

Ebner, L., Nalenz, M., ten Teije, A., van Harmelen, F., and Augustin, T. (2021). Expert rulefit: Complementing rule ensembles with expert knowledge. In *19th International Conference on Artificial Intelligence in Medicine, KR4HC Workshop*. Currently unavailable under the original address. Instead available under: `https://github.com/maltenlz/Malte-Nalenz/blob/main/ERF.pdf`

Decision rules closely resemble the way of human reasoning and arguing. This similarity allows the inclusion of already available knowledge, in the form of knowledge bases, expert systems and domain knowledge, that can be expressed in form of decision rules. This contribution explores the inclusion of already existing expert knowledge sources, in the form of decision rules, to complement the purely data based rules generated in rule ensemble methods. The rationale is that knowledge represented in textbooks and guidelines can be, to a certain degree, seen as validated. This validation may not be only statistically, but through clinical experiments, which adds valuable information. Including expert knowledge can also improve generalization performance, when pattern drift can be expected, the amount of training data is limited or the chance of confounder effects is high. As discussed above, one example is the application of a predictive model trained on hospital data. Purely data derived rules might capture characteristics that are specific to one hospital and not work well in another (Lee and He, 2019). On the other hand, validated rules are expected to also work well in other hospitals, as they are based on general knowledge. Another argument for using expert rules is that decision rules that comply with knowledge from other sources are to be preferred, and build trust in the model.
The proposed method Expert RuleFit (ERF) combines the set of rules extracted from decision trees with optional and confirmatory decision rules extracted from expert knowledge. Optional and confirmed decision rules differ in their degree of certainty. Confirmed rules might be knowledge acquired from textbooks that express biologically confirmed knowledge [2], whereas optional rules can be prior knowledge from previous studies, which can not be seen as validated (yet), but still add external information.
The full ERF model becomes

$$F(x) = \alpha_0 + \sum_{d=1}^{D} \alpha_d r_d(x) + \sum_{c \in \mathcal{I}_c} \alpha_c r_c(x) + \sum_{o \in \mathcal{I}_o} \alpha_o r_o(x) + \sum_{c \in \mathcal{I}_{c_l}} \alpha_{c_l} l_{c_l}(x) + \sum_{c \in \mathcal{I}_{l_\updownarrow}} \alpha_{c_o} l_{c_o}(x), \quad (3.6)$$

where $\mathcal{I}_c, \mathcal{I}_{c_l}$ are the indices of confirmed rules and linear effects respectively, $\mathcal{I}_o, \mathcal{I}_{o_l}$ the indices of the optional rules and optional linear effects.
Given the expert knowledge enriched rule-set, L1-regularized regression is used to find a sparse solution. As validated knowledge is preferred, if it fits the data reasonable well, we

[2] Knowledge that is based on the understanding of the underlying biological processes.

adjust the loss function to

$$\{\alpha^*, \beta^*, \beta_0^*\} = \underset{\beta_0,\beta,\alpha}{\arg\min} \left[L(y, F(x, \beta_0, \beta, \alpha)) + \lambda \left(\sum_{d=1}^{D} |\alpha_d| + \sum_{o \in \mathcal{I}_o} \nu |\alpha_o| + \sum_{o_l \in \mathcal{I}_{o_l}} \eta |\beta_{o_l}| \right) \right] \tag{3.7}$$

$$= \underset{\beta_0,\beta,\alpha}{\arg\min} \left[L(y, F(x, \beta_0, \beta, \alpha)) + \sum_{d=1}^{D} \lambda |\alpha_d| + \sum_{o \in \mathcal{I}_o} \lambda\nu |\alpha_o| + \sum_{o_l \in \mathcal{I}_{o_l}} \lambda\eta |\beta_{o_l}| \right], \tag{3.8}$$

where $\nu, \eta \in [0, 1]$ are discount factors. Choosing $\nu, \eta < 1$ leads to a preference of optional terms over data generated ones. In Equation 3.8 no penalty is given to the inclusion of validated knowledge, which leads to an automatic inclusion in the final model.
In first empirical results on a diabetes dataset we show, that the ERF model shows predictive performance on-par with the standard RuleFit model, while including a large proportion of expert derived knowledge. This can build trust from domain experts and also potentially generalize better to other datasets.

Comments and Outlook. The idea to induce expert knowledge acquired through different means, such as biological experiments, is very promising. In the medical domain, for many diseases a huge body of literature exists. It would be quite wasteful, to discard all prior knowledge. However, currently the knowledge acquisition is completely manual which may be problematic in two ways.
With manual acquisition, the extracted expert knowledge might be biased by the expectations and prior knowledge of the modeller. This point may also be seen from the opposite direction, as the modeller can make informed decisions about which knowledge to include, which might actually improve the model. The second downside of manual acquisition is the effort that it requires, as the amount of prior information can be vast. Automated acquisition might help in both regards, as it is somewhat objective and able to process large databases efficiently.

Another interesting application of ERF could be its usage to validate already acquired knowledge. Given a repository of standardized datasets, the relevant expert knowledge could be automatically acquired and validated empirically against other knowledge sources, as well as against data derived rules, to get a measure of their accuracy. This idea is connected to a problem in classical testing, where hypotheses are tested against null hypotheses, which is often unreasonable. Instead one could use the RuleFit approach to derive alternative hypotheses from the data, which gives a more realistic comparison to test against.

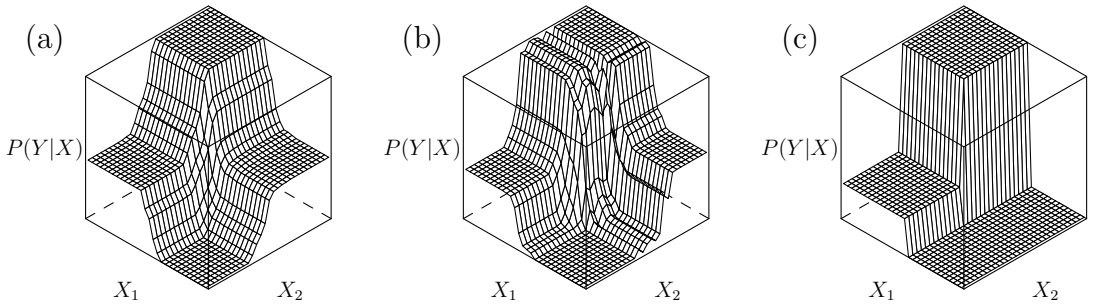


Figure 3.3: Decision Surface for a simulated dataset of (a) The first split using CRF, (b) full random forest model (c) CART tree with depth 2. CRF mimics the behaviour of random forests. Graph taken from Nalenz and Augustin (2021b).

3.1.4 Decision Forests as Decision Tree Uncertainty

Nalenz, M. and Augustin, T. (2021b). Cultivated random forests: Robust decision tree learning through tree structured ensembles. Technical Report. Available under: `https://epub.ub.uni-muenchen.de/77861`

This work addresses the issues of instability and poor generalization of classical tree induction methods. As argued in Section 2.1 the instability mostly stems from the greedy recursive learning procedure, which partitions the data very fast. Ensemble methods solve this issue with combining multiple individually unstable decision trees to produce a more stable and accurate – but also hard to interpret – final model.

This contribution explores an in-between approach, that mimics the behaviour of random forests while providing a relatively simple tree structure. At each node, instead of keeping only the locally best splitting rule, we keep all splits that are 'almost as good'. To this end two type of *ensemble modules* are introduced that combine sets of decisions into soft decision, similar to the compressed rules in Section 3.1.2. To capture the uncertainty about the correct split point, *robust split modules* keep the k closest splitpoints on both sides of the optimal splitpoint. Let t_0 denote the central splitpoint and $(x_{(i)}, w_{(i)})$ the i'th *ordered* covariate value and its fraction present in the current node to split then the robust split module consists of the set

$$\mathcal{T}(t_0) = \{t_{-j} = x_{(i-j)} \leq \ldots \leq t_{-1} = x_{(i-1)} \leq t_0 = x_{(i)} \leq t_1 = x_{(i+1)} \leq \ldots \leq t_m = x_{(i+m)}\},$$

where j and m are chosen as the highest value that the sum of weights on the left and right side of t_0 lower than k. The splitting rule can be calculated as in CRE (cf. equation 3.4) by

averaging over the individual decisions. During the induction process the whole set of split points is evaluated by averaging the individual impurity values implied by each split point. This can be seen as a form of regularization, as it forces the algorithm to prefer regions that are stable towards the purity measure and punishes regions where small changes in the covariate values lead to a drastic decline in the impurity measure. The idea is similar to the idea of robust split points in (Strobl and Augustin, 2009). If a splitpoint is good, small changes in the input space should not have a dramatic influence on the impurity value. This can also be seen from the viewpoint of data imprecision, with a fixed splitpoint, but asking the question how much the prediction would change, if the observation was slightly different. In contrast to previous soft decision tree models, the approach is non-parametric and purely based on the distribution of the training data around the split point in a given node.
To capture the uncertainty about the correct covariate to chose, *option modules* that consist of all covariates that lead to impurity criteria within a margin of the optimal one. The decision rules are directed, such that the left childnode has the higher implied target probability as in (Zimmermann, 2008).
At each node, while training and prediction, observations are passed to both childnodes as fractional observations, expressing our uncertainty about the correct split and leading to more stable tree structures (Abbasian et al., 2013), as 'close call' observations will be found in both childnodes. Combining multiple ensemble modules $\mathcal{M}$, provides relatively smooth decision boundaries, that mimic random forests. The decision boundary of the resulting *Cultivated Random Forest (CRF)* and its ability to mimic a random forest is shown in Figure 3.3 for simulated data.

CRF can be also interpreted as a specific kind of ensemble. The fraction of an observation that is present in a leafnode can be written as the average of an ensemble of trees, where the individual trees are elements of the Cartesian product of each ensemble module, via

$$\mathcal{L}(\mathbf{x}, \mathcal{M}_{\mathcal{L}}) = \prod_{\mathcal{M} \in \mathcal{M}_{\mathcal{L}}} \left(\frac{1}{|\mathcal{M}|} \sum_{d \in \mathcal{M}} d(\mathbf{x}) \right) = \frac{1}{|D_{\times}|} \sum_{D_{\mathcal{L}} \in D_{\times}} \left(\prod_{d \in D_{\mathcal{L}}} d(\mathbf{x}) \right) = \frac{1}{|D_{\times}|} \sum_{D_{\mathcal{L}} \in D_{\times}} \mathcal{L}(\mathbf{x}, D_{\mathcal{L}}), \tag{3.9}$$

where $M_{\mathcal{L}}$ is the set of modules that lead to the leaf node, d are the individual decisions in each module $\mathcal{M}$, and $D_{\times}$ the Cartesian product of all modules on the path to the leaf node. Intuitively, it is the same, if we make multiple decisions at each node, or average over the set of binary trees, spanned by the ensemble modules.
Empirically, we show that CRF reaches predictive performance close to random forests, while being structurally much simpler, as the model is centralized around a single tree structure. With that CRF offers a nice trade-off between predictive performance and model parsimony. Especially noteworthy is that using only robust split modules already leads to very decent predictive performance on some datasets, close to random forests and sometimes better. These models have the advantage of having an interpretation that is

similar to normal decision trees and hence relatively open for human interpretation.
CRF also provides a natural measure of model uncertainty: By looking at the spread of a given observations over the different leafs and its predictions, one can easily identify if the decisions were often close and hence the observation ends up in many different leaf nodes. For this observations the sensible approach could be to abstain from a prediction and instead use a different prediction method or consult an domain expert.

Comments and Outlook. In the current approach the margin within which different covariates are seen as 'almost as good' in the option modules needs to be pre-specified. This is clearly suboptimal, as the parameter most likely will depend on the data at hand. A more statistically motivated approach would be beneficial. The ideas presented in (Strobl and Augustin, 2009) might be interesting to this end. One heuristic could be to specify a fraction of observations that need to flip label, in order for two splits to be equally good.

A second potential improvement would be to also allow multiple robust split modules per covariate, to allow for multi-modal impurity surfaces. It is however unclear if this is desirable, as it will come at the price of making the model more complicated and one loses the relatively easy tree structure.

In general the proposed method does not require much additional computation at training time, as the entropy for each possible splitting point and covariate is computed regardless. A slight increase in computation, if trees are grown to full depth is due to the fact, that the data is separated at a slower rate, as many observations will be present in both child nodes, which on the other hand is beneficial in terms of model stability and generalizability. More optimised implementations, preferably written in high performance languages, such as C^{++}. would make CRF computationally slightly worse than CART and C4.5 but cheaper compared to full ensemble methods.

Given that computation is feasible, it would also be interesting to combine a few CRF trees again into an ensemble. The expectation is that the effect of re-sampling is less notable, as the weak learners are much more stable and CRF already mimics an ensemble of re-sampled trees. Therefore, we would expect less gains from bagging and random forests, but the combination with boosting could be quite interesting.

3.2 Machine Learning Applications under Severe Uncertainty

3.2.1 Regularized Regression for Single Cell Data

Fütterer, C., Nalenz, M., and Augustin, T. (2021). Discriminative power Lasso – incorporating discriminative power of genes into regularization-based variable selection. Technical Report. Available under: `https://epub.ub.uni-muenchen.de/77862`

The identification of biomarkers that can be used as indicators for certain outcomes of a disease is an important but challenging task. The problems mostly arise from the high dimensionality of the data, coupled with often relatively few observations. Additionally, the dependency structure can be complex, with high correlations between genes. Often already a small number of decisive genes can be found, that explain a certain disease reasonably well. The goal in this setting is therefore to select the most decisive covariates.

The commonly used L1-regularized regression can be used to select relatively small sets of candidate genes that can be used for either further analysis or predictions. However, due to the over-shrinkage effect and small N the variable selection performance and consistency of LASSO may be suboptimal.
In this contribution we explore the combination of clustering metrics with supervised regularized regression modelling. In a first step, for each gene the separation of the target groups is evaluated univariately. The biological reasoning is, that decisive genes should express differently in each group. To this end the clustering evaluation metrics Davies-Bouldin and Silhouette indices are used (Arbelaitz et al., 2013). Additionally, discriminative power based on ANOVA scores are considered. Instead of using the output from a clustering algorithm, directly the target classes are used as groups. This measure of *discriminative power* of each gene contains information about the compactness and difference in means between the target groups.
In a second step a customized L1-regression is used, where the individual penalties are proportional to the discriminative power, which is similar to the adaptive LASSO (Zou, 2006). This gives the multivariate model a push towards genes that also appear decisive univariatly and decompose nicely into groups. On the other hands, genes that only work well in a multivariate model, are penalized more heavily. As in this extreme $p >> N$ situation the overall uncertainty is high, this more cautious approach focuses on the clearly relevant genes. The resulting Discriminative Power Lasso (DP-L) has an interesting interpretation as a soft filtering approach. Instead of the often used hard filtering of genes prior to the modelling, we do not exclude genes, but instead promote the promising ones. This also reduces the 'researchers degrees of freedom' (Simmons et al., 2011) and thus may lead to a better reproducibility.

In experiments on single cell data and on simulated data, we show that the inclusion

of the discriminative power measure leads to significantly smaller final models, while being on par in terms of accuracy compared to Lasso, adaptive Lasso and the elastic net (Zou and Hastie, 2005). Especially, the precision of identifying relevant genes is significantly improved.

Comments and Outlook. An interesting extension would be to use Bayesian shrinkage priors, as in Section 3.1.1, which were already applied successfully previously to genetic data (Li and Yao, 2018). The Bayesian framework allows for more intuitive formulation of prior inclusion probabilities. Similar to the rule structured prior, one could therefore adjust the prior inclusion probabilities of promising genes.

Secondly, the application in other domains would be interesting. In areas, where the uncertainty of finding the correct covariates is high and the target groups are expected to express differently in their univariate distributions, DP-L could be used to perform a more robust variable selection compared to the Lasso.

3.2.2 Ensemble Learning under Complex Uncertainty

Kreiss, D., Nalenz, M., and Augustin, T. (2020). Undecided voters as set-valued information–machine learning approaches under complex uncertainty. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Tutorial and Workshop on Uncertainty in Machine Learning*. Available under: `https://github.com/maltenlz/Malte-Nalenz/blob/main/UV.pdf`

Undecided voters are a non-neglectable phenomenon in elections. Most current polling and forecasting approaches either force the undecided voter to give a (unjustified) precise answer, or simply drop them all-together. In this article we argue that both approaches do not represent the inherent imprecision of the party preference in a satisfying way. Let $S = \{1, .., s\}$ be the different options in an election, the true answer of an undecided voter can not be represented by any single element from S, but instead several elements, between which the individual is still pondering.

This can be expressed by the concept of *consideration sets*, that contain multiple elements from S, that the undecided voter has not yet decided against (Oscarsson and Rosema, 2019). With that, the consideration set can be seen as the power set $\mathcal{Y} = \mathcal{P}(S)$ of S. Each element in $\mathcal{Y}$ can be interpreted, from the so called ontic perspective, as an entity on their own, allowing the application of classic supervised learning approaches. For example, a multinomial regression model can be applied where each $\ell \in \mathcal{Y}$ is used as outcome category (Plass, 2018). In this article clustering algorithms are applied, in order to find structural differences between the different groups of undecided voters.
On the other hand, from the so called epistemic point of view, the consideration set can be interpreted as a coarse version of the true but at this time unknown outcome. This

interpretation is especially interesting for forecasting. Let $\ell \in \mathcal{P}(S)$ be the individuals consideration set and X the covariates, then under the assumption of random coarsening and identical conditional distributions of Y given X for the decided and the undecided voters, point valued estimated can be obtained via

$$\hat{P}(Y = l|\mathcal{Y} = \ell, X = x) = \frac{\hat{P}(Y = l|X = x, I_d = 1)}{\sum_{a \in \ell} \hat{P}(Y = a|X = x, I_d = 1)}, \tag{3.10}$$

where I_d is an indicator if the individual belongs to the group of decided voters (Kreiss and Augustin, 2020). This approach uses the decided voters to estimate the conditional distribution and normalizes the multinomial distribution with the probabilities from the outcomes that are part of the consideration set, excluding all other classes. The conditional distributions can be estimated using standard machine learning approaches. Because we expect voter preferences to depend on complex covariate interactions as well as non-linear dependencies, random forests is a natural choice for this estimation problem. Random forests are also naturally capable of using both numeric and categorical covariates. After estimating the conditional probability distribution we use (3.10) to refine the random forests estimates, by including the information of the consideration sets, producing a final point estimate for each individual. By that, all available information is used in a satisfying way.

Comments and Outlook. The decomposition in Equation 3.10 opens up the application of standard statistical or machine learning approaches, while at the same time taking into account the complex structure of the outcome. This approach is very data efficient, as it can make use of partial information, which otherwise would be discarded.

In this work the set valued consideration sets were reduced to a point valued estimate under the assumption, that the conditional distribution of the undecided is identical to the conditional distribution of the decided voters. However, if the goal is for example to forecast coalitions of parties, the consideration sets offer an even more natural way to reach point forecasts. For example, if a person is pondering between the green party and the SPD, one knows for certain that a potential coalition of green, SPD and FDP will receive its vote. This also reduces the forecast uncertainty as no estimate for the precise choice is required (Kreiss and Augustin, 2021).

3.3 Reproducibility Study

Seibold, H., Czerny, S., Decke, S., Dieterle, R., Eder, T., Fohr, S., Hahn, N., Hartmann, R., Heindl, C., Kopper, P., Lepke, D., Loidl, V., Mandl, M., Musiol, S., Peter, J., Piehler, A., Rojas, E., Schmid, S., Schmidt, H., Schmoll, M., Schneider, L., To, X.-Y., Tran, V., Völker, A., Wagner, M., Wagner, J., Waize, M., Wecker, H., Yang, R., Zellner, S., and Nalenz, M. (2021). A computational reproducibility study of PLOS ONE articles featuring longitudinal data analyses. *PLOS ONE*, 16(6).
Code: `https://gitlab.com/HeidiSeibold/reproducibility-study-plos-one`

Computational reproducibility, even when the used dataset and a description of the method used are available, is far from trivial (Artner et al., 2020). Typical problems involve vagueness in the description of the methods, especially missing specifications of parameters and preprocessing steps, missing data descriptions or coding errors and a lack of code in an open source language, such as `R` (R Core Team, 2021) or Python (Van Rossum and Drake Jr, 1995).

In this project computational reproducibility was measured empirically in the setting of a masters level course in 'longitudinal data analysis'. First, we selected papers published in PLOS ONE, that used longitudinal statistical methods (i.e. Generalized linear mixed models or Generalized estimated equations), had data and a data description available. Authors were asked for their cooperation in case of arising questions and only responsive authors included in the study. Using this criteria eleven papers were selected and distributed to student groups of 2-3 students each. The exercise sessions were used by the groups to work on their project during the semester. As an end result, each student group handed in a detailed report, including a summary of the content, the methods involved, the reproduction process, problems in the reproduction process, as well as their correspondence with the authors of the article [3]. Reproduction was performed solely with `R`, independent from the language used in the article.

Successful reproducibility was defined as reaching the same interpretation, given the data analysis. This relatively loose definition was required, as the analysed papers used very different models and interpreted different parameters and statistics. However, we performed a qualitative analysis for each article, about the difficulties, problems and solutions that were encountered. Overall, for eight out of the eleven articles we were able to reach the same interpretation as the authors. In the non-reproducible papers, the problems arose mainly through software issues coupled with a very vague methods description. Even though 8 out of 11 appears as a good quota for reproduction, for many articles, a considerable amount of reverse engineering was necessary to deduce important specifications, such as the correlation structure. Also only two papers were reproducible without contacting the authors to receive additional information or code. Overall, the study supports the demand that

[3] The student groups were encouraged to write emails to the authors in case of not surmountable problems.

source code and data should be provided for each submission. Additionally, vague methods descriptions make reproduction very difficult.

Comments and Outlook. The idea of using students to test reproducibility appears to be a win-win situation. After the course, we received very positive feedback from the student side. The motivation, to contribute to something important (reproducibility) was an important motivator. Through informal feedback we were told, that reproducing the results also helped to understand the methods better. This makes sense, as often the devil lays in the detail, when it comes to reproducing an article and therefore a thorough understanding of the methods is required. At the same time more studies are reproduced, which is a big win as well, as it directly adds to the credibility of results. To make this teaching framework easier to implement, it could help to define more detailed check lists as an orientation for the student groups. However such a detailed check list is hard to define for more complex models and presumably has to be topic specific.

The definition of 'reaching the same interpretation' used in this project was flexible enough for the study at hand, but is not suitable for larger scale studies. More formal definitions of reproducibility, such as the ones in Artner et al. (2020) would be very important for more complex statistical data analysis, but are hard to define. More research on reproducibility, such as Hoffmann et al. (2021) is therefore very important to ensure scientific progress.

4 Concluding remarks

This chapter contains a general resume and outlook. More detailed conclusions and outlooks for each contribution can be found in the previous chapter.

This thesis explored alternative representations of tree ensemble methods, by means of simpler models. In many cases interpretability can be improved by regularizing away unnecessary complexity and reshaping the remaining model variance in a more comprehensible form. Contrary to the often stated trade-off between simplicity and accuracy, simplifying forest methods does not necessarily decrease accuracy – and sometimes even improves it. Especially promising is the approach to represent tree ensembles by means of set-valued rules, that can compress the uncontrolled growth of the original forest into a much simpler form that focusses on the most central pattern. The possibility that a syntheses of simplicity and accuracy is possible gives hope that the current rise of machine learning methods in critical areas of society such as the digitalization of official statistics and healthcare will not necessarily be followed by a great depression due to arising problems connected to the lack of insight. The ability to give an explanation, as well as an honest characterization of model uncertainty, must be seen as a prerequisite for a successful and safe application, that is compliant with societal, political and ethical standards.

Even though decision rule ensembles promise a nice trade-off between simplicity and accuracy, in future work the validity, consistency and stability of rule ensembles need to be analysed more closely. The results presented in Gennatas et al. (2020) imply that decision rule ensembles often coincide with the expectation of domain experts, but more studies are needed on this important topic.

The frameworks presented here are very general and allow different directions moving forward. As they mostly build on forest methods in a post-processing manner, their relevance is directly connected to the popularity of forest methods. The application of forest methods in areas, where interpretation is often more important than prediction, such as psychology (Fokkema and Strobl, 2020) makes interpretable models even more appealing. An interesting future direction could be to use simplified representations of forests to allow a descriptive analysis of the most important pattern that the model relies on, rather than building a new predictive model. To this end, it may be possible to derive Pareto-efficient points between accuracy and simplicity, similar to the proportion of variance explained in principal component analysis. With that, the approach of representing tree ensemble methods by set-valued decision rule sets or trees is promising and a first step towards a safer application of tree ensemble methods.

Further references

Abbasian, H., Drummond, C., Japkowicz, N., and Matwin, S. (2013). Inner ensembles: Using ensemble methods inside the learning algorithm. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 33–48. Springer.

Abellan, J. and Moral, S. (2003). Maximum of entropy for credal sets. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 11(05):587–597.

Altman, D. G., Vergouwe, Y., Royston, P., and Moons, K. G. (2009). Prognosis and prognostic research: validating a prognostic model. *BMJ*, 338.

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., and Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1):243–256.

Artner, R., Verliefde, T., Steegen, S., Gomes, S., Traets, F., Tuerlinckx, F., and Vanpaemel, W. (2020). The reproducibility of statistical results in psychological research: An investigation using unpublished raw data. *Psychological Methods*.

Barocas, S., Hardt, M., and Narayanan, A. (2017). Fairness in machine learning. *Nips Tutorial*, 1:2017. `https://fairmlbook.org/pdf/fairmlbook.pdf`.

Bénard, C., Biau, G., Da Veiga, S., and Scornet, E. (2021). Sirus: Stable and interpretable rule set for classification. *Electronic Journal of Statistics*, 15:427–505.

Bennett, K. P., Cristianini, N., Shawe-Taylor, J., and Wu, D. (2000). Enlarging the margins in perceptron decision trees. *Machine Learning*, 41(3):295–313.

Bentéjac, C., Csörgő, A., and Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54(3):1937–1967.

Bernard, J.-M. (2005). An introduction to the imprecise dirichlet model for multinomial data. *International Journal of Approximate Reasoning*, 39(2-3):123–150.

Bernard, S., Heutte, L., and Adam, S. (2009). Influence of hyperparameters on random forest accuracy. In *International Workshop on Multiple Classifier Systems*, pages 171–180. Springer.

Bhadra, A., Datta, J., Polson, N. G., and Willard, B. (2019). Lasso meets horseshoe: A survey. *Statistical Science*, 34(3):405–427.

Biau, G. and Scornet, E. (2016). A random forest guided tour. *Test*, 25(2):197–227.

Boulesteix, A.-L., Janitza, S., Hornung, R., Probst, P., Busen, H., and Hapfelmeier, A. (2019). Making complex prediction rules applicable for readers: Current practice in random forest literature and recommendations. *Biometrical Journal*, 61(5):1314–1328.

Boulesteix, A.-L., Janitza, S., Kruppa, J., and König, I. R. (2012). Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(6):493–507.

Brajer, N., Cozzi, B., Gao, M., Nichols, M., Revoir, M., Balu, S., Futoma, J., Bae, J., Setji, N., Hernandez, A., et al. (2020). Prospective and external evaluation of a machine learning model to predict in-hospital mortality of adults at time of admission. *JAMA Network Open*, 3(2).

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Buchanan, B. G. and Shortliffe, E. H. (1984). *Rule-based expert systems: the MYCIN experiments of the Stanford Heuristic Programming Project*. Addison-Wesley Longman Publishing.

Bühlmann, P. (2006). Boosting for high-dimensional linear models. *The Annals of Statistics*, 34(2):559–583.

Bühlmann, P. (2012). Bagging, boosting and ensemble methods. In *Handbook of Computational Statistics*, pages 985–1022. Springer.

Bühlmann, P. and Hothorn, T. (2007). Boosting algorithms: Regularization, prediction and model fitting. *Statistical science*, 22(4):477–505.

Bühlmann, P. and Yu, B. (2002). Analyzing bagging. *The Annals of Statistics*, 30:927–961.

Buntine, W. (1992). Learning classification trees. *Statistics and Computing*, 2(2):63–73.

Carreira-Perpinán, M. A. and Tavallali, P. (2018). Alternating optimization of decision trees, with application to learning sparse oblique trees. *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 31:1211–1221.

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

Carvalho, C. M., Polson, N. G., and Scott, J. G. (2009). Handling sparsity via the horseshoe. In *Artificial Intelligence and Statistics*, pages 73–80. PMLR.

Carvalho, C. M., Polson, N. G., and Scott, J. G. (2010). The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. Association for Computing Machinery.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298.

Cohen, W. W. (1995). Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning*, pages 115–123. Association for Computing Machinery.

Corani, G., Abellán, J., Masegosa, A., Moral, S., and Zaffalon, M. (2014). Classification. In Augustin, T., Coolen, F. P., De Cooman, G., and Troffaes, M. C., editors, *Introduction to Imprecise Probabilities*, pages 916–954. Wiley.

Couronné, R., Probst, P., and Boulesteix, A.-L. (2018). Random forest versus logistic regression: a large-scale benchmark experiment. *BMC Bioinformatics*, 19(1):1–14.

Dasgupta, S. and Freund, Y. (2008). Random projection trees and low dimensional manifolds. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 537–546. Association for Computing Machinery.

Dembczyński, K., Kotłowski, W., and Słowiński, R. (2010). Ender: a statistical framework for boosting decision rules. *Data Mining and Knowledge Discovery*, 21(1):52–90.

Deng, L. and Platt, J. (2014). Ensemble deep learning for speech recognition. In *15th Annual Conference of the International Speech Communication Association*. International Speech Communication Association.

Dietterich, T. (2002). Ensemble learning. *The Handbook of Brain Theory and Neural Networks*, 2(1):110–125.

Duroux, R. and Scornet, E. (2018). Impact of subsampling and tree depth on random forests. *ESAIM: Probability and Statistics*, 22:96–128.

Ebner, L., Nalenz, M., ten Teije, A., van Harmelen, F., and Augustin, T. (2021). Expert rulefit: Complementing rule ensembles with expert knowledge. In *19th International Conference on Artificial Intelligence in Medicine, KR4HC Workshop*. Currently unavailable under the original address. Instead available under: `https://github.com/maltenlz/Malte-Nalenz/blob/main/ERF.pdf`.

Efron, B. (1982). *The jackknife, the bootstrap and other resampling plans*. SIAM.

Feilke, M., Bischl, B., Schmid, V. J., and Gertheiss, J. (2016). Boosting in nonlinear regression models with an application to dce-mri data. *Methods of Information in Medicine*, 55(01):31–41.

Fernandez-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research*, 15:3133–3181.

Fokkema, M. (2020). Fitting prediction rule ensembles with R package pre. *Journal of Statistical Software*, 92:1–30.

Fokkema, M. and Strobl, C. (2020). Fitting prediction rule ensembles to psychological research data: An introduction and tutorial. *Psychological Methods*, 25(5):636.

Frank, E., Mayo, M., and Kramer, S. (2015). Alternating model trees. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, pages 871–878. Association for Computing Machinery.

Freund, Y. and Mason, L. (1999). The alternating decision tree learning algorithm. In *Proceedings of the 16th International Conference on Machine Learning*, volume 99, pages 124–133.

Freund, Y., Schapire, R., and Abe, N. (1999). A short introduction to boosting. *Artificial Intelligence*, 14(771-780):1612.

Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning*, volume 96, pages 148–156. Citeseer.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 29:1189–1232.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378.

Friedman, J. H. and Popescu, B. E. (2008). Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3):916–954.

Friedman, J. H., Popescu, B. E., et al. (2003). Importance sampled learning ensembles. *Journal of Machine Learning Research*, 94305:1–32.

Fürnkranz, J. (1999). Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13(1):3–54.

Fütterer, C., Nalenz, M., and Augustin, T. (2021). Discriminative power Lasso – incorporating discriminative power of genes into regularization-based variable selection. Technical Report. Available under: `https://epub.ub.uni-muenchen.de/77862`.

Gennatas, E. D., Friedman, J. H., Ungar, L. H., Pirracchio, R., Eaton, E., Reichmann, L. G., Interian, Y., Luna, J. M., Simone, C. B., Auerbach, A., Delgado, E., van der Laan, M. J., Solberg, T. D., and Valdes, G. (2020). Expert-augmented machine learning. *Proceedings of the National Academy of Sciences*, 117(9):4571–4577.

Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1):3–42.

Ghosh, J., Li, Y., and Mitra, R. (2018). On the use of cauchy prior distributions for Bayesia logistic regression. *Bayesian Analysis*, 13(2):359–383.

Hahn, P. R., He, J., and Lopes, H. F. (2019). Efficient sampling for gaussian linear regression with arbitrary priors. *Journal of Computational and Graphical Statistics*, 28(1):142–154.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2017). *The elements of statistical learning: Data mining, inference, and prediction*. Springer.

Hastie, T. J. and Tibshirani, R. J. (2017). *Generalized additive models*. Routledge.

Hill, J., Linero, A., and Murray, J. (2020). Bayesian additive regression trees: a review and look forward. *Annual Review of Statistics and Its Application*, 7:251–278.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844.

Hoffmann, S., Schönbrodt, F., Elsas, R., Wilson, R., Strasser, U., and Boulesteix, A.-L. (2021). The multiplicity of analysis strategies jeopardizes replicability: lessons learned across disciplines. *Royal Society Open Science*, 8(4):201925.

Hothorn, T., Hornik, K., and Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical statistics*, 15(3):651–674.

Jawanpuria, P., Jagarlapudi, S. N., and Ramakrishnan, G. (2011). Efficient rule ensemble learning using hierarchical kernels. In *Proceedings of the 28th International Conference on Machine Learning*, pages 161–168. Omnipress.

Justice, A. C., Covinsky, K. E., and Berlin, J. A. (1999). Assessing the generalizability of prognostic information. *Annals of Internal Medicine*, 130(6):515–524.

Khandagale, S., Xiao, H., and Babbar, R. (2020). Bonsai: diverse and shallow trees for extreme multi-label classification. *Machine Learning*, 109(11):2099–2119.

Kiselev, V. Y., Kirschner, K., Schaub, M. T., Andrews, T., Yiu, A., Chandra, T., Natarajan, K. N., Reik, W., Barahona, M., and Green, A. R. (2017). SC3: consensus clustering of single-cell rna-seq data. *Nature Methods*, 14(5):483–486.

Klein, N., Nott, D. J., and Smith, M. S. (2021). Marginally calibrated deep distributional regression. *Journal of Computational and Graphical Statistics*, 30(2):467–483.

Kreiss, D. and Augustin, T. (2020). Undecided voters as set-valued information–towards forecasts under epistemic imprecision. In *International Conference on Scalable Uncertainty Management*, pages 242–250. Springer.

Kreiss, D. and Augustin, T. (2021). Towards a paradigmatic shift in pre-election polling adequately including still undecided voters – some ideas based on set-valued data for the 2021 german federal election. *arXiv*. arXiv:2109.12069.

Kreiss, D., Nalenz, M., and Augustin, T. (2020). Undecided voters as set-valued information–machine learning approaches under complex uncertainty. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Tutorial and Workshop on Uncertainty in Machine Learning*. Available under: `https://github.com/maltenlz/Malte-Nalenz/blob/main/UV.pdf`.

Kuhn, M. and Johnson, K. (2013). *Applied predictive modeling*, volume 26. Springer.

Lee, J. and He, Q. P. (2019). Understanding the effect of specialization on hospital performance through knowledge-guided machine learning. *Computers & Chemical Engineering*, 125:490–498.

Li, L. and Yao, W. (2018). Fully Bayesian logistic regression with hyper-LASSO priors for high-dimensional feature selection. *Journal of Statistical Computation and Simulation*, 88(14):2827–2851.

Linero, A. R. (2018). Bayesian regression trees for high-dimensional prediction and variable selection. *Journal of the American Statistical Association*, 113(522):626–636.

Linero, A. R. and Yang, Y. (2018). Bayesian regression tree ensembles that adapt to smoothness and sparsity. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(5):1087–1110.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4768–4777. Curran Associates Inc.

Makalic, E. and Schmidt, D. F. (2015). A simple sampler for the horseshoe estimator. *IEEE Signal Processing Letters*, 23(1):179–182.

Mantas, C. J. and Abellán, J. (2014). Credal-C4.5: Decision tree based on imprecise probabilities to classify noisy data. *Expert Systems with Applications*, 41(10):4625–4637.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35.

Meinshausen, N. (2010). Node harvest. *The Annals of Applied Statistics*, 4:2049–2072.

Mingers, J. (1989). An empirical comparison of pruning methods for decision tree induction. *Machine Learning*, 4(2):227–243.

Mirzamomen, Z. and Kangavari, M. R. (2017). A framework to induce more stable decision trees for pattern classification. *Pattern Analysis and Applications*, 20(4):991–1004.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Murthy, S. K., Kasif, S., and Salzberg, S. (1994). A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–32.

Nalenz, M. and Augustin, T. (2021a). Compressed rule ensemble learning. Under review for AISTATS. Preprint available under: `https://github.com/maltenlz/Malte-Nalenz/blob/main/CRE.pdf`.

Nalenz, M. and Augustin, T. (2021b). Cultivated random forests: Robust decision tree learning through tree structured ensembles. Technical Report. Available under: `https://epub.ub.uni-muenchen.de/77861`.

Nalenz, M. and Villani, M. (2018). Tree ensembles with rule structured horseshoe regularization. *Annals of Applied Statistics*, 12(4):2379–2408.

Ogutu, J. O., Piepho, H.-P., and Schulz-Streeck, T. (2011). A comparison of random forests, boosting and support vector machines for genomic selection. In *BMC Proceedings*, volume 5, pages 1–5. BioMed Central.

Oliveira, M. and Torgo, L. (2015). Ensembles for time series forecasting. In *Asian Conference on Machine Learning*, pages 360–370. PMLR.

Oscarsson, H. and Rosema, M. (2019). Consideration set models of electoral choice: Theory, method, and application. *Electoral Studies*, 57:256–262.

Patil, D. D., Wadhai, V., and Gokhale, J. (2010). Evaluation of decision tree pruning algorithms for complexity and classification accuracy. *International Journal of Computer Applications*, 11(2):23–30.

Philipp, M., Zeileis, A., and Strobl, C. (2016). A toolkit for stability assessment of tree-based learners. Technical report, Working Papers in Economics and Statistics, University of Innsbruck. Available under: `https://www2.uibk.ac.at/downloads/c4041030/wpaper/2016-11.pdf`.

Piironen, J. and Vehtari, A. (2017). Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics*, 11(2):5018–5051.

Plass, J. (2018). Statistical modelling of categorical data under ontic and epistemic imprecision: contributions to power set based analyses, cautious likelihood inference and (non-)testability of coarsening mechanism. PhD thesis, Department of Statistics, LMU Munich.

Podgorelec, V., Kokol, P., Stiglic, B., and Rozman, I. (2002). Decision trees: an overview and their use in medicine. *Journal of Medical Systems*, 26(5):445–463.

Probst, P. and Boulesteix, A.-L. (2017). To tune or not to tune the number of trees in random forest. *Journal of Machine Learning Research*, 18(1):6673–6690.

Probst, P., Wright, M. N., and Boulesteix, A.-L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(3):e1301.

Quinlan, J. R. (1993). *C4. 5: Programs for Machine Learning*. Morgan Kaufmann.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. Vienna, Austria. `https://www.R-project.org/`.

Ripley, B. D. (1996). *Pattern recognition and neural networks*. Cambridge University Press.

Rodriguez, J. J., Kuncheva, L. I., and Alonso, C. J. (2006). Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1619–1630.

Rokach, L. and Maimon, O. (2005). Top-down induction of decision trees classifiers-a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(4):476–487.

Schapire, R. E. (1999). A brief introduction to boosting. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2*, pages 1401–1406. Morgan Kaufmann publishers Inc.

Schapire, R. E. (2003). The boosting approach to machine learning: An overview. *Nonlinear Estimation and Classification*, pages 149–171.

Scornet, E. (2017). Tuning parameters in random forests. *ESAIM: Proceedings and Surveys*, 60:144–162.

Seibold, H., Czerny, S., Decke, S., Dieterle, R., Eder, T., Fohr, S., Hahn, N., Hartmann, R., Heindl, C., Kopper, P., Lepke, D., Loidl, V., Mandl, M., Musiol, S., Peter, J., Piehler, A., Rojas, E., Schmid, S., Schmidt, H., Schmoll, M., Schneider, L., To, X.-Y., Tran, V., Völker, A., Wagner, M., Wagner, J., Waize, M., Wecker, H., Yang, R., Zellner, S., and Nalenz, M. (2021). A computational reproducibility study of PLOS ONE articles featuring longitudinal data analyses. *PLOS ONE*, 16(6).

Seibold, H., Zeileis, A., and Hothorn, T. (2016). Model-based recursive partitioning for subgroup analyses. *The International Journal of Biostatistics*, 12(1):45–63.

Simmons, J. P., Nelson, L. D., and Simonsohn, U. (2011). False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, 22(11):1359–1366.

Steyerberg, E. W., Moons, K. G., van der Windt, D. A., Hayden, J. A., Perel, P., Schroter, S., Riley, R. D., Hemingway, H., Altman, D. G., et al. (2013). Prognosis research strategy (progress) 3: prognostic model research. *PLOS Medicine*, 10(2).

Strobl, C. and Augustin, T. (2009). Adaptive selection of extra cutpoints – an approach towards reconciling robustness and interpretability in classification trees. *Journal of Statistical Theory and Practice*, 3(1):119–135.

Strobl, C., Boulesteix, A.-L., and Augustin, T. (2007a). Unbiased split selection for classification trees based on the Gini index. *Computational Statistics & Data Analysis*, 52(1):483–501.

Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., and Zeileis, A. (2008). Conditional variable importance for random forests. *BMC Bioinformatics*, 9(1):1–11.

Strobl, C., Boulesteix, A.-L., Zeileis, A., and Hothorn, T. (2007b). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, 8(1):1–21.

Strobl, C., Kopf, J., and Zeileis, A. (2015). Rasch trees: A new method for detecting differential item functioning in the rasch model. *Psychometrika*, 80(2):289–316.

Strobl, C., Malley, J., and Tutz, G. (2009). An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological Methods*, 14(4):323.

Tam, S.-M. and Clarke, F. (2015). Big data, official statistics and some initiatives by the Australian bureau of statistics. *International Statistical Review*, 83(3):436–448.

Terenin, A., Dong, S., and Draper, D. (2019). Gpu-accelerated gibbs sampling: a case study of the horseshoe probit model. *Statistics and Computing*, 29(2):301–310.

Toll, D., Janssen, K., Vergouwe, Y., and Moons, K. (2008). Validation, updating and impact of clinical prediction rules: a review. *Journal of Clinical Epidemiology*, 61(11):1085–1094.

Van Rossum, G. and Drake Jr, F. L. (1995). *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands. `https://ir.cwi.nl/pub/5008` (visited on 16/11/2021).

Vinayak, R. K. and Gilad-Bachrach, R. (2015). Dart: Dropouts meet multiple additive regression trees. In *Artificial Intelligence and Statistics*, pages 489–497. PMLR.

Wei, D., Dash, S., Gao, T., and Gunluk, O. (2019). Generalized linear rule models. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6687–6696. PMLR.

Weiss, S. M. and Indurkhya, N. (2000). Lightweight rule induction. In *Proceedings of the 17th International Conference on Machine Learning*, pages 1135–1142. Morgan Kaufmann Publishers Inc.

Zech, J. R., Badgeley, M. A., Liu, M., Costa, A. B., Titano, J. J., and Oermann, E. K. (2018). Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLOS Medicine*, 15(11).

Zeileis, A., Hothorn, T., and Hornik, K. (2008). Model-based recursive partitioning. *Journal of Computational and Graphical Statistics*, 17(2):492–514.

Zhao, P. and Yu, B. (2006). On model selection consistency of Lasso. *The Journal of Machine Learning Research*, 7:2541–2563.

Zhou, Z.-H. (2021). Ensemble learning. In *Machine Learning*, chapter 8, pages 181–210. Springer.

Zimmermann, A. (2008). Ensemble-trees: Leveraging ensemble power inside decision trees. In *Proceedings of the 11th International Conference on Discovery Science*, pages 76–87. Springer.

Zou, H. (2006). The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.

Attached contributions

The Annals of Applied Statistics
2018, Vol. 12, No. 4, 2379–2408
https://doi.org/10.1214/18-AOAS1157

TREE ENSEMBLES WITH RULE STRUCTURED HORSESHOE REGULARIZATION

BY MALTE NALENZ AND MATTIAS VILLANI

Linköping University

We propose a new Bayesian model for flexible nonlinear regression and classification using tree ensembles. The model is based on the RuleFit approach in Friedman and Popescu [*Ann. Appl. Stat.* **2** (2008) 916–954] where rules from decision trees and linear terms are used in a L1-regularized regression. We modify RuleFit by replacing the L1-regularization by a horseshoe prior, which is well known to give aggressive shrinkage of noise predictors while leaving the important signal essentially untouched. This is especially important when a large number of rules are used as predictors as many of them only contribute noise. Our horseshoe prior has an additional hierarchical layer that applies more shrinkage a priori to rules with a large number of splits, and to rules that are only satisfied by a few observations. The aggressive noise shrinkage of our prior also makes it possible to complement the rules from boosting in RuleFit with an additional set of trees from Random Forest, which brings a desirable diversity to the ensemble. We sample from the posterior distribution using a very efficient and easily implemented Gibbs sampler. The new model is shown to outperform state-of-the-art methods like RuleFit, BART and Random Forest on 16 datasets. The model and its interpretation is demonstrated on the well known Boston housing data, and on gene expression data for cancer classification. The posterior sampling, prediction and graphical tools for interpreting the model results are implemented in a publicly available R package.

1. Introduction. Learning and prediction when the mapping between input and outputs is potentially nonlinear and observed in noise remains a major challenge. Given a set of N training observations $(\mathbf{x}, y)_i, i = 1, \ldots, N$, we are interested in learning or approximating an unknown function f observed in additive Gaussian noise

$$y = f(\mathbf{x}) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}\big(0, \sigma^2\big),$$

and to use the model for prediction. A popular approach is to use a learning ensemble [Breiman (1996, 2001), Freund and Schapire (1996), Friedman (2001)]

$$f(\mathbf{x}) = \sum_{l=1}^{m} \alpha_l f_l(\mathbf{x}),$$

Received February 2017; revised February 2018.
Key words and phrases. Nonlinear regression, classification, decision trees, Bayesian, prediction, MCMC, interpretation.

where $f_l(\mathbf{x})$ is a basis function (also called a weak learner in the machine learning literature) for a subset of the predictors. A variety of basis functions f_l have been proposed in the last decades, and we will here focus on decision rules. Decision rules are defined by simple if-else statements and therefore highly interpretable by humans. Finding a set of optimal rules is NP hard [Friedman and Popescu (2008)], and most practical algorithms therefore use a greedy learning procedure. Among the most powerful are divide and conquer approaches [Cohen (1995), Fürnkranz (1999)] and boosting [Schapire (1999), Dembczyński, Kotłowski and Słowiński (2010)].

A new way to learn decision rules is introduced in Friedman and Popescu (2008) in their RuleFit approach. RuleFit is estimated by a two-step procedure. The *rule generation* step extracts decision rules from an ensemble of trees trained with gradient boosting. The second *regularizaton* step learns the weights α_l for the generated rules via L1-regularized (Lasso) regression, along with weights on linear terms included in the model. This is similar to stacking [Wolpert (1992), Breiman (1996)], with the important difference that the members of the ensemble are not learned decision trees or other predictors, but individual rules extracted from trees. As argued in Friedman and Popescu (2008), this makes RuleFit a more interpretable model and, we argue below, has important consequences for the regularization part. RuleFit has been successfully applied in particle physics, in medical informatics and in life sciences. Our paper makes the following contributions to improve and enhance RuleFit.

First, we replace the L1-regularization [Tibshirani (1996)] in RuleFit by a generalized horseshoe regularization prior [Carvalho, Polson and Scott (2010)] tailored specifically to covariates from a rule generation step. L1-regularization is computationally attractive, but has the well-known drawback of also shrinking the effect of the important covariates. This is especially problematic here since the number of rules from the rule generation step can be very large while potentially only a small subset is necessary to explain the variation in the response. Another consequence of the overshrinkage effect of the L1-regularization is that it is hard to choose an optimal number of rules; increasing the number of rules affects the shrinkage properties of the Lasso. This makes it very hard to determine the number of rules a priori, and one has to resort to cross-validation, thereby mitigating the computational advantage of the Lasso. A horseshoe prior is especially attractive for rule learning since it shrinks uninformative predictors aggressively while leaving important ones essentially untouched. Inspired by the prior distribution on the tree depth in Bayesian Additive Regression Trees (BART) [Chipman, George and McCulloch (2010)], we design a generalized horseshoe prior that shrinks overly complicated and specific rules more heavily, thereby mitigating problems with overfitting. This is diametrically opposed to RuleFit, and to BART and boosting, which all combine a myriad of rules into a collective where single rules only play a very small part.

Second, we complement the tree ensemble from gradient boosting [Friedman (2001)] in RuleFit with an additional set of trees generated with Random Forest.

The error-correcting nature of boosting makes the rules highly dependent on each other. Trees from Random Forest [Breiman (2001)] are much more random and adding them to rules from boosting therefore brings a beneficial diversity to the tree ensemble. Note that it is usually not straightforward to combine individual trees from different ensemble strategies in a model; our combination of RuleFit and horseshoe regularization is an ideal setting for mixing ensembles since RuleFit makes it easy to combine ensembles, and the horseshoe prior can handle a large number of noise rules without overfitting.

Third, an advantage of our approach compared to many other flexible regression and classification models is that predictions from our model are based on a relatively small set of interpretable decision rules. The possibility to include linear terms also simplifies interpretation since it avoids a common problem with decision trees that linear relationships need to be approximated with a large number of rules. To further aid in the interpretation of the model and its predictions, we also propose graphical tools for analyzing the model output. We also experiment with post-processing methods for additional pruning of rules to simplify the interpretation even further using the method in Hahn and Carvalho (2015).

We call the resulting two-step procedure with mixed rule generation followed by generalized rule structured horseshoe regularization the *HorseRule* model. We show that HorseRule's ability to keep the important rules and aggressively removing unimportant noise rules leads to both great predictive performance and high interpretability.

The structure of the paper is as follows. Section 2 describes the decision rule generation method in HorseRule. Section 3 presents the horseshoe regularization prior and the MCMC algorithm for posterior inference. Section 4 illustrates aspects of the approach on simulated data and evaluates and compares the predictive performance of HorseRule to several main competing methods on a wide variety of real datasets. Section 5 concludes.

2. Decision rule generation. This section describes the *rule generation step* of HorseRule, which complements the rules from gradient boosting in Friedman and Popescu (2008) with rules from Random Forest with completely different properties.

2.1. *Decision rules.* Let S_k denote the set of possible values of the covariate x_k and let $s_{k,m} \subseteq S_k$ denote a specific subset. A decision rule can then be written as

$$r_m(\boldsymbol{x}) = \prod_{k \in Q_m} I(x_k \in s_{k,m}), \tag{2.1}$$

where $I(x)$ is the indicator function and Q_m is the set of variables used in defining the mth rule. A decision rule $r_m \in \{0, 1\}$ takes the value 1 if all of its $|Q_m|$ conditions are fulfilled and 0 otherwise. For orderable covariates $s_{k,m}$ will be an

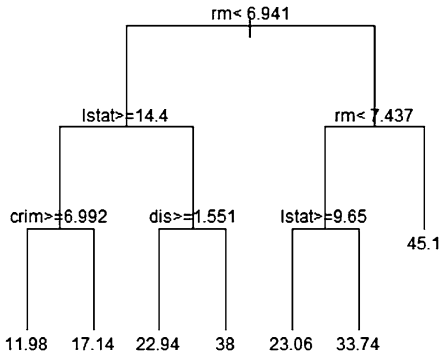


FIG. 1. *Decision tree for the Boston housing data.*

interval or a disjoint union of intervals, while for categorical covariates $s_{k,m}$ are explicitly enumerated. There is a long tradition in machine learning to use decision rules as weak learners. Most algorithms learn decision rules directly from data, such as in Cohen (1995), Dembczyński, Kotłowski and Słowiński (2010). RuleFit exploits the fact that decision trees can be seen as a set of decision rules. In a first step a tree ensemble is generated, which is then decomposed into its defining decision rules. Several efficient (greedy) algorithmic implementations are available for constructing the tree ensembles. The generated rules typically correspond to interesting subspaces with great predictive power. Each node in a decision tree is defined by a decision rule. Figure 1 shows an example tree for the Boston housing dataset and Table 1 its corresponding decision rules. We briefly describe this dataset here since it will be used as a running example throughout the paper. The Boston housing data consists of $N = 506$ observations which are city areas in Boston and $p = 13$ covariates are recorded. These variables include ecological

TABLE 1
Corresponding rules, defining the decision tree

Rules	**Conditions**
r_1	$RM \geq 6.94$
r_2	$RM < 6.94$
r_3	$RM < 6.94$ & $LSTAT < 14.4$
r_4	$RM < 6.94$ & $LSTAT \geq 14.4$
r_5	$RM < 6.94$ & $LSTAT < 14.4$ & $CRIM < 6.9$
r_6	$RM < 6.94$ & $LSTAT < 14.4$ & $CRIM \geq 6.9$
r_7	$RM \geq 6.94$ & $LSTAT < 14.4$ & $DIS < 1.5$
r_8	$RM \geq 6.94$ & $LSTAT < 14.4$ & $DIS \geq 1.5$
r_9	$6.94 \leq RM < 7.45$
r_{10}	$6.94 \leq RM < 7.45$
r_{11}	$6.94 \leq RM < 7.45$ & $LSTAT < 9.7$
r_{12}	$6.94 \leq RM < 7.45$ & $LSTAT \geq 9.7$

measures of nitrogen oxides (NOX), particulate concentrations (PART) and proximity to the Charles River (CHAS), the socio-economic variables proportion of black population (B), property tax rate (TAX), proportion of lower status population (LSTAT), crime rate (CRIM), pupil teacher ratio (PTRATIO), proportion of old buildings (AGE), the average number of rooms (RM), area proportion zoned with large lots (ZN), the weighted distance to the employment centers (DIS) and an index of accessibility to key infrastructure (RAD). The dependent variable is the median housing value in the area.

Using equation (2.1) for example, r_{11} can be expressed as

$$r_{11}(\boldsymbol{x}) = \prod_{k \in Q_{11}} I(x_k \in s_{k,11}) = I(6.94 \leq RM < 7.45)\, I(LSTAT < 9.7).$$

This rule is true for areas with relatively large houses with between 6.94 and 7.45 rooms and less than 9.7% lower status population. The mth tree consists of $2(u_m - 1)$ rules, where u_m denotes the number of terminal nodes. Therefore $\sum_{m=1}^{M} 2(u_m - 1)$ rules can be extracted from a tree ensemble of size M.

2.2. *Collinearity structure of trees.* The generated rules will be combined in a linear model and collinearity is a concern. For example, the two first child nodes in each tree are perfectly negative correlated. Furthermore, each parent node is perfectly collinear with its two child nodes, as it is their union. One common way to deal with the collinearity problem is to include the terminal nodes only. This approach also reduces the number of rules and therefore simplifies computations. We have nevertheless chosen to consider all possible rules including also nonterminal ones, but to randomly select one of the two child nodes at each split. The reason for also including nonterminal nodes is three-fold. First, even though each parent node in a tree can be reconstructed as a linear combination of terminal nodes, when using regularization this equivalence no longer holds. Second, our complexity penalizing prior in Section 3.3 is partly based on the number of splits to measure the complexity of a rule, and will therefore shrink the several complex child nodes needed to approximate a simpler parent node. Third, the interpretation of the model is substantially simplified if the model can select a simple parent node instead of many complex child nodes.

2.3. *Generating an informative and diverse rule ensemble.* Any tree method can be used to generate decision rules. Motivated by the experiments in Friedman and Popescu (2003), Rulefit uses gradient boosting for rule generation [Friedman and Popescu (2008)]. Gradient boosting [Friedman (2001)] fits each tree iteratively on the pseudo residuals of the current ensemble in an attempt to correct mistakes made by the previous ensemble. This procedure introduces a lot of dependence between the members of the ensemble, and many of the produced rules tend to be informative only when combined to an ensemble. It might therefore not be possible to remove a lot of the decision rules without destroying this dependency structure.

Random Forest on the other hand generates trees independently from all previous trees [Breiman (2001)]. Each tree tries to find the individually best partitioning, given a random subset of observations and covariates. Random Forest will often generate rules with very similar splits, and the random selection of covariates forces it to often generate decision rules based on uninformative predictors. Random Forest will therefore produce more redundant and uninformative rules compared to gradient boosting, but the generated rules with strong predictive power are not as dependent on the rest of the ensemble.

Since the rules from boosting and Random Forest are very different in nature, it makes sense to use both types of rules to exploit both methods' advantages. This naturally leads to a larger number of candidate rules, but the generalized horseshoe shrinkage proposed in Section 3.2 and 3.3 can very effectively handle redundant rules. Traditional model combination methods usually use weighting schemes on the output of different ensemble methods [Rokach (2010)]. In contrast we combine the extracted rules from the individual trees. To the best of our knowledge this combination of individual weak learners from different ensemble methods is novel and fits nicely in the RuleFit framework with horseshoe regularization, as explained in the Introduction.

The tuning parameters used in the tree generation determine the resulting decision rules. The most impactful is the tree-depth, controlling the complexity of the resulting rules. We follow Friedman and Popescu (2008) with setting the depth of tree m to

$$td_m = 2 + \lfloor \varphi \rfloor, \tag{2.2}$$

where $\lfloor x \rfloor$ is the largest integer less or equal than x and φ is a random variable following the exponential distribution with mean $L-2$. Setting $L=2$ will produce only tree stumps consisting of one split. With this indirect specification the forest is composed of trees of varying depth, which allows the model to be more adaptive to the data and makes the choice of a suitable tree depth less important. We use this approach for both boosted and random forest trees.

Another important parameter is the minimum number of observations in a node $n_{\min}$. A too small $n_{\min}$ gives very specific rules and the model is likely to capture spurious relationships. Using $n_{\min} = N^{\frac{1}{3}}$ as a default setting has worked well in our experiments, but if prior information about reasonable sizes of subgroups in the data is available the parameter can be adjusted accordingly. Another choice is to determine $n_{\min}$ by cross validation.

In the following, all other tuning parameters, for example, the shrinkage parameter in gradient boosting or the number of splitting covariates in the Random Forest, are set to their recommended standard choices implemented in the R-packages *randomForest* and *gbm*.

3. Ensembles and rule based horseshoe regularization. This section discusses the *regularization step* of HorseRule and present a new horseshoe shrinkage prior tailored specifically for covariates in the form of decision rules.

3.1. *The ensemble.* Once a suitable set of decision rules is generated, they can be combined in a linear regression model of the form

$$y = \alpha_0 + \sum_{l=1}^{m} \alpha_l r_l(\mathbf{x}) + \varepsilon.$$

As $r_l(\mathbf{x}) \in \{0, 1\}$ they already have the form of dummy variables and can be directly included in the regression model. A simple but important extension is to also include linear terms

$$y = \alpha_0 + \sum_{j=1}^{p} \beta_j x_j + \sum_{l=1}^{m} \alpha_l r_l(\mathbf{x}) + \varepsilon. \tag{3.1}$$

This extension addresses the difficulty of rule and tree based methods to approximate linear effects. Splines, polynomials, time effects, spatial effects or random effects are straightforward extensions of equation (3.1).

Friedman and Popescu (2008) do not standardize the decision rules, which puts a higher penalty on decision rules with a smaller scale. To avoid this behavior, we scale the predictors to have zero mean and unit variance.

3.2. *Bayesian regularization through the horseshoe prior.* A large set of candidate decision rules is usually necessary to have a high enough chance of finding good decision rules. The model in (3.1) will therefore always be high dimensional and often $p + m > n$. Many of the rules will be uninformative and correlated with each other. Regularization is therefore a necessity.

RuleFit uses L1-regularized estimates, which corresponds to an a posterior mode estimator under a double exponential prior in a Bayesian framework [Tibshirani (1996)]. As discussed in the Introduction, the global shrinkage effect of L1-regularization can be problematic for rule covariates. L1-regularization is well known to lead to both shrinkage and variable selection. There now exist implementations of RuleFit that use the elastic net instead of L1-Regularization, which can lead to improved predictive performance [Zou and Hastie (2005)], however elastic net still only uses one global shrinkage parameter.

Another common Bayesian variable selection approach is based on the spike-and-slab prior [George and McCulloch (1993), Smith and Kohn (1996)]

$$\beta_j \sim w \cdot N\big(\beta_j; 0, \lambda^2\big) + (1 - w) \cdot \delta_0, \tag{3.2}$$

where δ_0 is the Dirac point mass function, $N(\beta_j; 0, \lambda^2)$ is the normal density with zero mean and variance λ^2, and w is the prior inclusion probability of predictor x_j. Discrete mixture priors enjoy attractive theoretical properties, but need to explore a model space of size $2^{(p+m)}$, which can be problematic when either p or m are large. The horseshoe prior by Carvalho, Polson and Scott (2009, 2010) mimics the behavior of the spike-and-slab but is computationally more attractive. The

regression model with the original horseshoe prior for linear regression is of the form

$$y|\mathbf{X}, \boldsymbol{\beta}, \sigma^2 \sim \mathcal{N}_n(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}_n), \tag{3.3}$$

$$\beta_j|\lambda_j, \tau^2, \sigma^2 \sim \mathcal{N}(0, \lambda_j\tau^2\sigma^2), \tag{3.4}$$

$$\sigma^2 \sim \sigma^{-2}\, d\sigma^2, \tag{3.5}$$

$$\lambda_j \sim \mathcal{C}^+(0, 1), \tag{3.6}$$

$$\tau \sim \mathcal{C}^+(0, 1), \tag{3.7}$$

where $\mathcal{C}^+(0, 1)$ denotes the standard half-Cauchy distribution. We use horseshoe priors on both linear [the β's in equation (3.1)] and rule terms [the α's in equation (3.1)]. The horseshoe shrinkage for β_j is determined by a local shrinkage parameter $\lambda_j > 0$ and a global shrinkage parameter $\tau > 0$. This is important since it allows aggressive shrinking of noise covariates through small values of τ, while allowing individual signals to have large coefficients through large λ_j. Carvalho, Polson and Scott (2010) show that the horseshoe is better at recovering signals than the Lasso, and the models obtained from the horseshoe are shown to be almost indistinguishable from the ones obtained by a well defined spike-and-slab prior.

3.3. *Horseshoe regularization with rule structure.* The original horseshoe assigns the same prior distribution to all regression coefficients, regardless of the rule's complexity (number of splits in the tree) and the specificity (number of data points that fulfill the rule). Similar to the tree structure prior in BART, we therefore modify the horseshoe prior to express the prior belief that rules with high length (many conditions) are less likely to reflect a true mechanism. In addition, we also add the prior information that very specific rules that are satisfied by only a few data points are also improbable a priori. The rule support $s(r_l) \in (0, 1)$ is given by $s(r_j) = N^{-1}\sum_{i=1}^{N} r_j(\mathbf{x}_i)$. Note that a support of 95% can also be interpreted as 5%. Therefore we express the specificity of a rule through $\min(1 - s(r_j), s(r_j))$ instead. These two sources of prior information are incorporated by extending the prior on λ_j to

$$\lambda_j \sim \mathcal{C}^+(0, A_j),$$

with

$$A_j = \frac{(2 \cdot \min(1 - s(r_j), s(r_j)))^{\mu}}{(l(r_j))^{\eta}}, \tag{3.8}$$

where $l(r_j)$ denotes the length of rule j defined as its number of conditions. With increasing number of conditions the prior shrinkage becomes stronger, as well as with increasing specificity. The hyperparameter μ controls the strength of our

belief to prefer general rules that cover a lot of observations and η determines how strongly we prefer simple rules. The response y should be scaled when using the rule structure prior since the scale of β depends on the scale of y.

The rule structure for A_j in equation (3.8) is designed such that $A_j = 1$ for rules with support 0.5 and length 1, as the ideal. Since $\lim_{\mu\to 0,\eta\to 0} A_j = 1$, our rule structure prior approaches the standard horseshoe prior for small μ and η. The rule structure prior gives a gentle push towards simple and general rules, but its Cauchy tails put considerable probability mass on nonzero values even for very small A_j; the data can therefore overwhelm the prior and keep a complex and specific rule if needed.

A model with many complex specific rules may drive out linear terms from the model, thereby creating an unnecessarily complicated model. We therefore use a standard prior with $A = 1$ for linear terms, and set the parameters μ and η to values larger than 0, which has the effect of giving linear effects a higher chance of being chosen a priori. When p is small it may also be sensible to use no shrinkage at all on the linear effects, and this is also allowed in our Gibbs sampling algorithm in Section 3.4. The hyperparameters μ and η can be chosen guided by theoretical knowledge about what kind of rules and linear effects are reasonable for a problem by hand, or determined via cross validation. As a default choice $(\mu, \eta) = (1, 2)$ worked well in our experiments, penalizing rule complexity heavily and low rule support moderately.

3.4. *Posterior inference via Gibbs sampling.* Posterior samples can be obtained via Gibbs sampling. Sampling from the above hierarchy is expensive, as the full conditionals of λ_j and τ do not follow standard distributions and slice sampling has to be used. Makalic and Schmidt (2016) propose an alternative Horseshoe hierarchy that exploits the following mixture representation of a half-Cauchy distributed random variable $X \sim \mathcal{C}^+(0, \Psi)$:

$$X^2|\psi \sim \mathcal{IG}\left(\frac{1}{2}, \frac{1}{\psi}\right), \tag{3.9}$$

$$\psi \sim \mathcal{IG}\left(\frac{1}{2}, \frac{1}{\Psi^2}\right), \tag{3.10}$$

which leads to conjugate conditional posterior distributions. The sampling scheme in Makalic and Schmidt (2016) samples iteratively from the following set of full conditional posteriors:

$$\boldsymbol{\beta}|\cdot \sim \mathcal{N}_p(\mathbf{A}^{-1}\mathbf{X}^T\mathbf{y}, \sigma^2\mathbf{A}^{-1}),$$

$$\sigma^2|\cdot \sim \mathcal{IG}\left(\frac{n+p}{2}, \frac{(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})}{2} + \frac{\boldsymbol{\beta}^T {\boldsymbol{\Lambda}_*}^{-1}\boldsymbol{\beta}}{2}\right),$$

$$\lambda_j^2|\cdot \sim \mathcal{IG}\left(1, \frac{1}{\nu_j} + \frac{{\beta_j}^2}{2\tau^2\sigma^2}\right),$$

$$\tau^2|\cdot \sim \mathcal{IG}\left(\frac{p+1}{2}, \frac{1}{\rho} + \frac{1}{2\sigma^2}\sum_{j=1}^{p}\frac{{\beta_j}^2}{{\lambda_j}^2}\right),$$

$$\nu_j|\cdot \sim \mathcal{IG}\left(1, \frac{1}{A^2} + \frac{1}{\lambda_j^2}\right),$$

$$\rho|\cdot \sim \mathcal{IG}\left(1, 1 + \frac{1}{\tau^2}\right),$$

with $\mathbf{A} = (\mathbf{X}^T\mathbf{X} + {\mathbf{\Lambda}_*}^{-1})$, $\mathbf{\Lambda}_* = \tau^2\mathbf{\Lambda}$, $\mathbf{\Lambda} = \text{diag}({\lambda_1}^2, \ldots, {\lambda_p}^2)$.

3.5. *Computational considerations*. The computational complexity of HorseRule can be mainly composed in rule generation and weight learning. The computational cost will thereby always be higher than using boosting or Random Forest alone. This speed disadvantage is partly mitigated by the fact that the HorseRule performs well also without cross-validation.

We have used the R implementations gbm and randomForest here. These algorithms do not scale well for large N and p and become a bottleneck for $N > 10{,}000$. This can be easily remedied by migrating the rule generation step to Xtreme Gradient Boosting (XGBoost) [Chen and Guestrin (2016)] or lightGBM [Ke et al. (2017)] that are magnitudes faster for big datasets.

Compared to Bayesian tree learning procedures such as BART or the recently proposed Dirichlet Adaptive Regression Trees (DART) [Linero (2018)], no Metropolis–Hastings steps are necessary to learn the tree structure in HorseRule; HorseRule uses only Gibbs sampling on a regularized linear model with rule covariates, which scales linearly with the number of observations [Makalic and Schmidt (2016)]. Sampling 1000 draws from the posterior distribution in the HorseRule model for the Boston housing data used in Section 4.7 takes about 90 seconds on a standard computer. The complexity of the Horseshoe sampling depends mostly on the number of linear terms and decision rules, and increases only slowly with N. Li and Yao (2014) suggest a computational shortcut where a given β_j is sampled in a given iteration only if the corresponding scale $(\lambda_j \cdot \tau)$ is higher than a threshold. The λ_j needs to be sampled in every iteration to give every covariate the chance of being chosen in the next iteration. We have implemented this approach and seen that it can give tremendous computational gains, but we have not used it when generating the results here since the effects it has on the invariant distribution of the MCMC scheme needs to be explored further. Finally, for very large N ($> 10{,}000$) the linear algebra operations in the Gibbs sampling can become time consuming, and GPU acceleration can be used to speed up sampling [Terenin, Dong and Draper (2016)].

3.6. *Sampling the splitting points*. The BART model can be seen as the sum of trees with a Gaussian prior on the terminal node values

$$\mu_j \sim \mathcal{N}\left(0, \frac{0.5}{\tau\sqrt{k}}\right),$$

where k denotes the number of trees. BART uses a fixed regularization parameter τ and samples the tree structure, while HorseRule uses a fixed rule structure and adapts to the data through sampling the shrinkage parameters λ_j and τ. Using a fixed tree structure offers dramatic computational advantages, as no Metropolis–Hastings updating steps are necessary, but the splits are likely to be suboptimal with respect to the whole ensemble.

As shown in Section 4, both HorseRule and BART achieve great predictive performance through different means, and a combination in which both shrinkage and tree structure are sampled in a fully Bayesian way could be very powerful, but computational very demanding. An intermediate position is to keep the splitting variables fixed in HorseRule, but to sample the splitting points. We have observed that HorseRule often keeps very similar rules with slightly different splitting points in the ensemble, which is a discrete approximation to sampling the splitting points. Hence this could also improve interpretability since a large number of rules with nearby splitting points can be replaced by a single rule with an estimated splitting point. It is also possible to replace many similar rules with suitable basis expansions, such as cubic terms or splines.

4. Results. This section starts out with a predictive comparison of HorseRule against a number of competitors on 16 benchmark datasets. The following subsections explore several different aspects of HorseRule on simulated and real data to evaluate the influence of different components of the model. Section 4.2 contrasts the ability of RuleFit and HorseRule to recover a true linear signal in models with additional redundant rules. The following subsection uses two real datasets to demonstrate the effect of having linear effects in HorseRule, and the advantage of using horseshoe instead of L1 for regularization. Section 4.4 addresses that HorseRule uses the training data both to generate the rules and for learning the weights. Section 4.5 explores the role of the rule generating process in HorseRule, and Section 4.6 the sensitivity to the number of rules. Finally in Sections 4.7 and 4.8 we showcase HorseRule's ability to make interpretable inference from data in different domains.

4.1. *Prediction performance comparison on* 16 *datasets*. We compare the predictive performance of HorseRule with competing methods on 16 regression datasets. The datasets are a subset of the datasets used in Chipman, George and McCulloch (2010). From the 23 datasets that were available to us online we excluded datasets that lacked a clear description of which variable to use as response, or which data preprocessing has to be applied to get to the version described in

TABLE 2
Summary of the 16 *regression datasets used in the evaluation. N, Q and C are the number of observations, quantitative and categorical predictors, respectively*

Name	N	Q	C	Name	N	Q	C
Abalone	4177	7	1	Diamond	308	1	3
AIS	202	11	1	Hatco	100	6	4
Attend	838	6	3	Heart	200	13	3
Baskball	96	4	0	Fat	252	14	0
Boston	506	13	0	MPG	392	6	1
Budget	1729	10	0	Ozone	330	8	0
CPS	534	7	3	Servo	167	2	2
CPU	209	6	1	Strike	625	4	1

Chipman, George and McCulloch (2010). Since both RuleFit and HorseRule assume Gaussian responses, we also excluded datasets with clearly non-Gaussian response variables, for example count variables with excessive number of zeros. HorseRule can be straightforwardly adapted by using a negative-binomial data augmentation scheme [Makalic and Schmidt (2016)], but we leave this extension for future work. Table 2 displays the characteristics of the datasets.

We compare HorseRule to RuleFit [Friedman and Popescu (2008)], Random Forest [Breiman (2001)], Bayesian Additive Regression Trees (BART) [Chipman, George and McCulloch (2010)], Dirichlet Adaptive Regression Trees (DART) [Linero (2018)], a recent variant of BART that uses regularization on the input variables, and XGBoost [Chen and Guestrin (2016)] a highly efficient implementation of gradient boosting.

We use 10-fold cross validation on each dataset and report the relative RMSE (RRMSE) in each fold; RRMSE for a fold is the RMSE for a method divided by the RMSE of the best method on that fold. This allows us to compare performance over different datasets with differing scales and problem difficulty. We also calculate a worst RRMSE (wRRMSE) on the dataset level, as a measure of robustness. wRRMSE is based on the maximal difference across all datasets between a method's RRMSE and the RRMSE of the best method for that dataset; hence a method with low wRRMSE is not far behind the winner on any dataset. We also calculate the mean RRMSE (mRRMSE) as the relative RMSE on dataset level averaged over all datasets.

To ensure a fair comparison we use another (nested) five-fold cross validation in each fold to find good values of the tuning parameters for each method. For BART and Random Forest the cross-validation settings from Chipman, George and McCulloch (2010) are used. DART is relatively independent of parameter tuning, through the usage of hyperpriors, so we only determine the optimal number of trees. For RuleFit we cross-validate over the number of rules and the depth of the trees, as those are the potentially most impactful parameters. The shrinkage τ

TABLE 3
Settings for the compared methods

Method	Parameter settings
HR-default	Ensemble: GBM+RF; $L = 5$; $(\mu, \eta) = (1, 2)$.
HR-CV	Ensemble: GBM+RF; $L = (2, 5, 8)$; $(\mu, \eta) = ((0, 0), (0.5, 0.5), (1, 2))$.
RuleFit	$k = 500, 1000, \ldots, 5000$; $L = (2, 5, 8)$.
Random Forest	Fraction of variables used in each tree $= (0.25, 0.5, 0.75, 1, \sqrt{p}/p)$.
BART	$(\gamma, q) = ((3, 0.9), (3, 0.99), (10, 0.75))$; $\tau = 2, 3, 5$; number of trees: 50, 200.
DART	Number of trees: 50, 100.
XGBoost	Number of trees: 50, 100, 200, 350, 500; $\nu = 0.1, 0.05, 0.01$; tree depth: 4, 6, 8.

in RuleFit is determined by the model internally. XGBoost has many parameters that can be optimized, we chose the number of trees, the shrinkage parameter and the tree depth as the most important. For HorseRule we use cross-validation to identify suitable hyperparameters (μ, η) as well as the tree depth. We also run a HorseRule version with the proposed standard settings without cross-validation. Table 3 summarizes the settings of all methods.

We first compare the three different HorseRule versions. Figure 2 shows the predictive performance of the HorseRule models over $10 \cdot 16 = 160$ dataset and cross-validation splits. While the $(\mu, \eta) = (1, 2)$ already performs better than the prior without rule structure $[(\mu, \eta) = (0, 0)]$, cross-validation of (μ, η) helps to improve performance further.

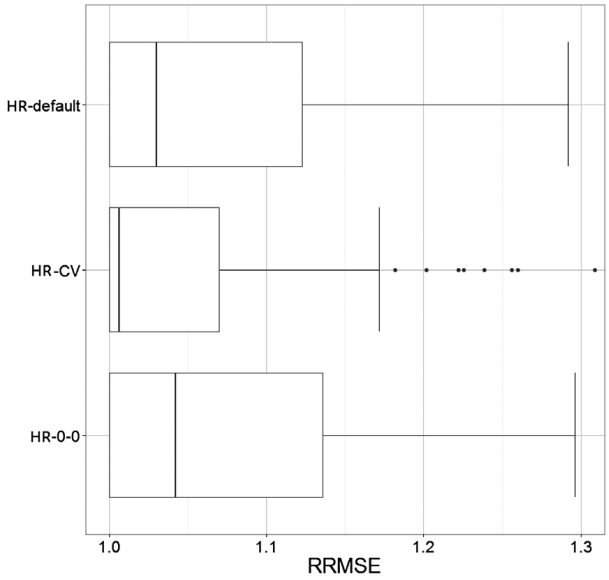


FIG. 2. *RRMSE comparison of the different HorseRule versions across all folds.*

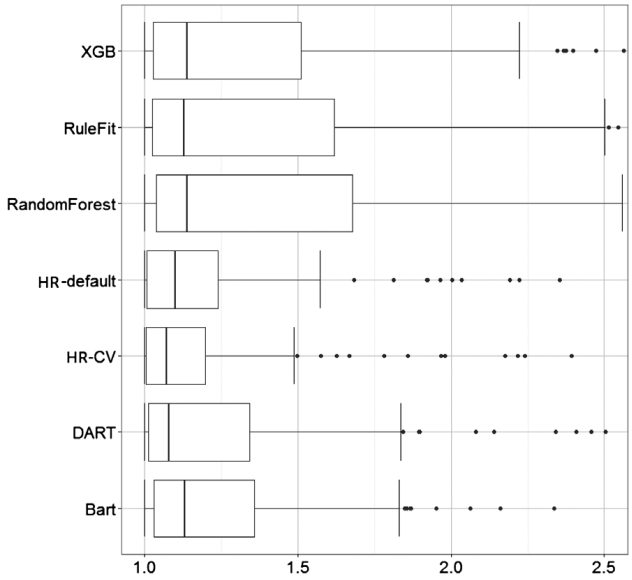


FIG. 3. *RRMSE comparison of HorseRule with competing methods across all folds.*

Figure 3 and Table 4 show that HorseRule has very good performance across all datasets and folds, and the median RRMSE is smaller than its competitors. DART also performs well and is second best in terms of median RRMSE. HorseRule-default is the third best method for the median and best for the mean, which is quite impressive since it does not use cross-validation.

Table 5 summarizes the performance on the dataset level. DART is the best model on $7/16$ datasets and has the best average rank. HorseRule-CV is the best method on $5/16$ datasets and has a slightly worse rank than DART. The last rows of Table 5 displays the wRRMSE and mRRMSE over all datasets for each method; it shows that whenever HorseRule is not the best method, it is only marginally behind the winner. This is not true for any of the other methods which all perform substantially worse than the best method on some datasets. RuleFit performs the

TABLE 4
RRMSE distribution over the 160 *crossvalidation folds of the competing methods*

	25%-Quant	**Median**	**Mean**	**75%-Quant**
XGBoost	1.02	1.139	1.496	1.509
RuleFit	1.026	1.129	1.426	1.618
RandomForest	1.039	1.137	1.508	1.677
HR-default	1.007	1.101	**1.247**	1.238
HR-CV	**1.004**	**1.072**	1.262	**1.198**
DART	1.012	1.080	1.376	1.342
BART	1.030	1.131	1.377	1.357

TABLE 5

Cross-validated prediction performance for the 16 *regression datasets. Each entry shows the RMSE and in parentheses the rank on this dataset. The best result is marked in bold*

	BART	**RForest**	**RuleFit**	**HorseRule**	**HorseRule-CV**	**DART**	**XGBoost**
Abalone	2.150 (7)	2.119 (3)	2.139 (5)	2.115(2)	**2.114** (1)	2.129 (4)	2.147 (6)
AIS	1.144 (4)	1.247 (7)	1.207 (6)	0.713 (2)	**0.699** (1)	1.061 (3)	1.188 (5)
Attend	394,141 (5)	411,900 (7)	**345,177** (1)	398,485 (6)	365,010 (2)	370,006 (4)	367,231 (3)
Baskball	0.087 (3)	0.086 (2)	0.088 (4)	0.088 (4)	0.092 (7)	**0.083** (1)	0.089 (6)
Boston	2.867 (2)	3.153 (7)	3.037 (6)	2.940 (4)	2.926 (3)	**2.819** (1)	2.97 (5)
Budget	0.039 (2)	**0.038** (1)	0.061 (7)	0.041 (4)	0.042 (5)	0.056 (6)	0.039 (2)
CPS	4.356 (3)	4.399 (6)	4.386 (5)	**4.348** (1)	4.370 (4)	4.353 (2)	4.448 (7)
CPU	41.52 (4)	54.08 (6)	54.50 (7)	**36.03** (1)	37.47 (3)	42.87 (5)	36.75 (2)
Diamond	215.0 (3)	465.9 (7)	233.7 (4)	184.5 (2)	**171.27** (1)	245.8 (5)	343.6 (6)
Hacto	0.453 (7)	0.311 (6)	0.297 (5)	0.261 (2)	**0.260** (1)	0.264 (3)	0.269 (4)
Heart	8.917 (2)	9.048 (3)	9.349 (7)	9.241 (5)	9.070 (4)	**8.869** (1)	9.310 (6)
Fat	1.306 (6)	1.114 (2)	1.173 (3)	1.264 (5)	1.245 (4)	**1.072** (1)	1.329 (7)
MPG	2.678 (3)	2.692 (5)	2.672 (2)	2.714 (6)	2.689 (4)	**2.642** (1)	2.750 (7)
Ozone	4.074 (3)	4.061 (2)	4.189 (7)	4.120 (4)	4.165 (5)	**4.054** (1)	4.174 (6)
Servo	0.588 (5)	0.486 (3)	0.502 (4)	0.409 (2)	**0.403** (1)	0.671 (6)	0.719 (7)
Strikes	458.4 (7)	453.7 (5)	447.7 (3)	449.2 (4)	447.2 (2)	**447.1** (1)	456.6 (6)
Av. Rank	3.9375	4.5625	4.8750	3.5625	3	**2.9375**	5.3125
wRRMSE	1.742	2.720	1.726	1.179	**1.160**	1.666	2.006
mRRMSE	1.128	1.250	1.182	1.051	**1.035**	1.141	1.201

best on $1/16$ datasets, and the median RRMSE is slightly lower than for Random Forest and XGBoost. XGBoost has the hightest median RRMSE and rank in this experiment. This is probably due to the fact, that all methods except Random Forest rely to a certain degree on boosting and improve different aspects of it, making it a hard competition for XGBoost.

To summarize, the results show that HorseRule is a highly competitive method with a very stable performance across all datasets. The rule structured prior was found to improve predictive performance, and performs well also without time-consuming cross-validation of its hyperparameters.

4.2. *Regularization of linear terms and rules—RuleFit vs. HorseRule.* This subsection uses simulated data to analyse the ability of HorseRule and RuleFit to recover the true signal when the true relationship is linear and observed with noise. The data is generated with $X_i \sim \mathcal{N}(0, 1), i = 1, \ldots, 100$, $Y = 5X_1 + 3X_2 + X_3 + X_4 + X_5 + \varepsilon$ and $\varepsilon \sim \mathcal{N}(0, 1)$. The first five predictors thus have a positive dependency with y of varying magnitude while the remaining 95 covariates are noise. Table 6 reports the results from 100 simulated datasets. RMSE measures the discrepancy between the fitted values and the true mean for unseen test data. RuleFit and HorseRule model use 500 rules in addition to the linear terms. The best

TABLE 6
Simulation study. The true effect is linear

	$RMSE$			$\Delta\beta_{\text{true}}$			$\Delta\beta_{\text{noise}}$		
	$n = 100$	$n = 500$	$n = 1000$	$n = 100$	$n = 500$	$n = 1000$	$n = 100$	$n = 500$	$n = 1000$
OLS	3.23	1.10	1.06	1.25	0.19	0.14	2302	3.78	2.54
Horseshoe Regression	1.14	1.01	1.01	0.40	0.18	0.13	1.72	0.70	0.49
HorseRule $\alpha = 0, \beta = 0$	1.54	1.02	1.01	1.99	0.39	0.29	2.74	0.22	0.15
HorseRule $\alpha = 1, \beta = 2$	1.25	1.02	1.01	1.15	0.37	0.28	3.14	0.37	0.24
RuleFit $k = 2000$	1.84	1.23	1.15	3.58	1.42	1.05	1.18	0.91	0.99

model in RMSE is as expected the Horseshoe regression without any rules. The OLS estimates without any regularization struggles to avoid overfitting with all the unnecessary covariates and does clearly worse than the other methods. HorseRule without the rule structure prior outperforms RuleFit, but adding a rule structured prior gives an even better result. The differences between the models diminishes quickly with the sample size (since the data is rather clean), the exception being RuleFit which improves at a much lower rate than the other methods. Table 6 also breaks down the results into the ability to recover the true linear signal, measured by $\Delta\beta_{\text{true}} = |(\beta_1, \beta_2, \beta_3, \beta_4, \beta_5) - (5, 3, 1, 1, 1)|_1$, and the ability to remove the noise covariates, measured by $\Delta\beta_{\text{noise}} = |(\beta_6, \ldots, \beta_{100}) - (0, \ldots, 0)|_1$. We see that the HorseRule's horseshoe prior is much better at recovering the true linear signal compared to RuleFit with its L1-regularization. OLS suffers from its inability to shrink away the noise.

Even though such clear linear effects are rare in actual applications, the simulation results in Table 6 shows convincingly that HorseRule will prioritize and accurately estimate linear terms when they fit the data well. This is in contrast to RuleFit which shrinks the linear terms too harshly and compensates the lack of fit with many rules. HorseRule will only try to add nonlinear effects through decision rules if they are really needed.

4.3. *Influence of linear terms in HorseRule, and regularizing by horseshoe instead of L*1. In this section we analyze to what extent HorseRule's good performance depends on having linear terms in the model, and how crucial the horseshoe regularization is for performance. We demonstrate the effect of these model specification choices on the two datasets Diamonds and Boston. The Diamonds dataset was selected since HorseRule is much better than its competitors on that dataset. The Boston data was chosen since it will be used for a more extensive analysis in

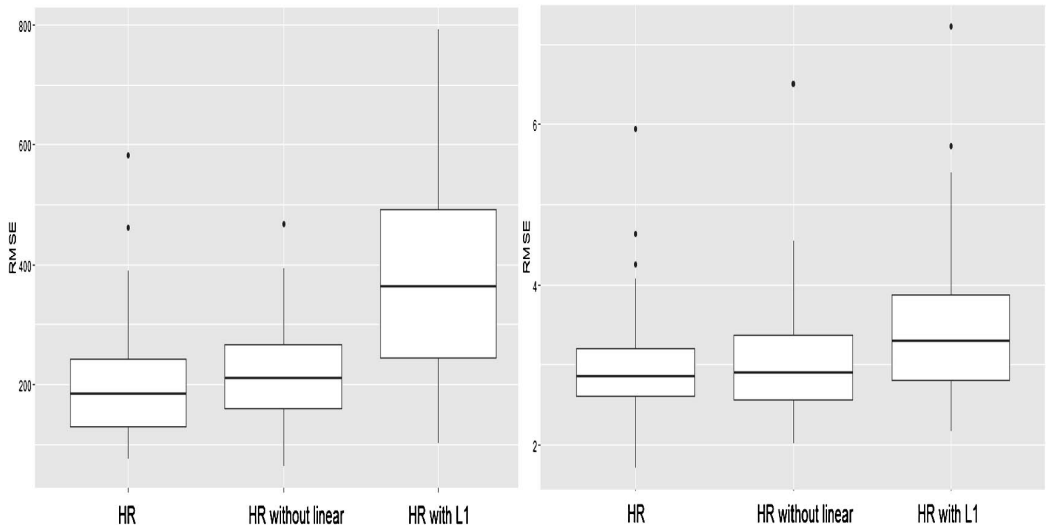


FIG. 4. *RMSE on the Diamond* (*left*) *and the Boston* (*right*) *dataset when linear terms are removed and when using* $L1$ *regularization instead of horseshoe*.

Section 4.7. Figure 4 shows the RMSE distribution over the folds used in 10-fold cross-validation. The results are replicated 10 times using different random seeds. The results show that the aggressive shrinkage offered of the horseshoe prior is essential for HorseRule; changing to L1 increases RMSE, especially for the Diamond data. Note that the L1-version is not entirely identical to RuleFit, as RuleFit uses different preprocessing on rules and only boosting generated rules [Friedman and Popescu (2008)]. Figure 4 also shows that adding linear terms gives small decrease of RMSE, but seems less essential for HorseRule's performance.

4.4. *Influence of the two-step procedure*. One concern of our two-step procedure is that the same training data is used to find rules and to learn the weights. This double use of the data can distort the posterior distribution and uncertainty estimates. It should be noted however that the rule generation uses only random subsets of data, which mitigates this effect to some extent. It is also important to point out that the predictive results presented in this paper are always on an unseen test set so this is not an issue for the performance evaluations.

One way to obtain a more coherent Bayesian interpretation is to split the training data in two parts: one part for the rule generation and one part for learning the weights. We can view this as conditionally coherent if the rule learned from the first part of the data is seen as prior experience of the analyst in analyzing the second part of the data. An obvious drawback with such an approach is that less data can be used for learning the model, which will adversely affect predictive performance. Table 7 displays how predictive performance on the Diamonds ($N = 308$) and Boston ($N = 506$) data deteriorates from a 50/50 split of the training data. Both these datasets are small and we have also included the moderately large Abalone

TABLE 7
Median RMSE for different splitting strategies

	Diamond	**Boston**	**Abalone**
All data	184.6	2.851	2.115
50/50 split	283.7	3.555	2.136

data ($N = 4177$); for this dataset the data splitting has essentially no effect on the performance. Hence, data-splitting may be an attractive option for moderately large and large data if proper Bayesian uncertainty quantification is of importance.

4.5. *Influence of the rule generating process.* In this section we analyze the influence of different rule generating processes on model performance for the Diamond dataset with ($N = 308$ and $p = 4$) and the Boston housing data ($N = 506$ and $p = 13$).

In each setting 1000 trees with an average tree depth of $L = 5$ are used, using different ensemble strategies for the rule generation:

1. Random Forest generated rules plus linear terms.
2. Gradient boosting generated rules plus linear terms.
3. A combination of 30% of the trees from Random Forest and 70% from gradient boosting plus linear terms.

The results are shown in Figure 5. As expected the error-correcting rules found by gradient boosting outperforms randomly generated rules from Random For-

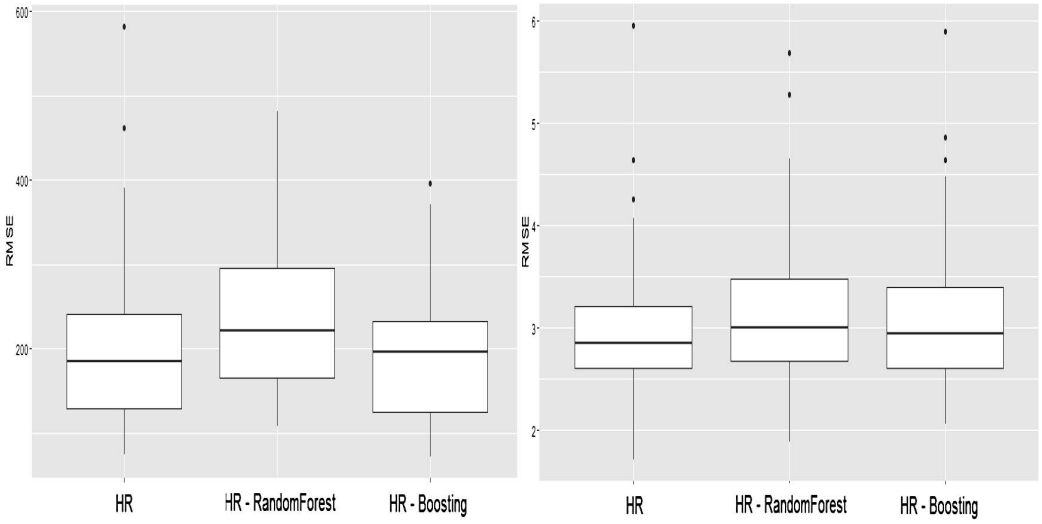


FIG. 5. *RMSE on the Diamond (left) and the Boston (right) dataset for different rule generating strategies.*

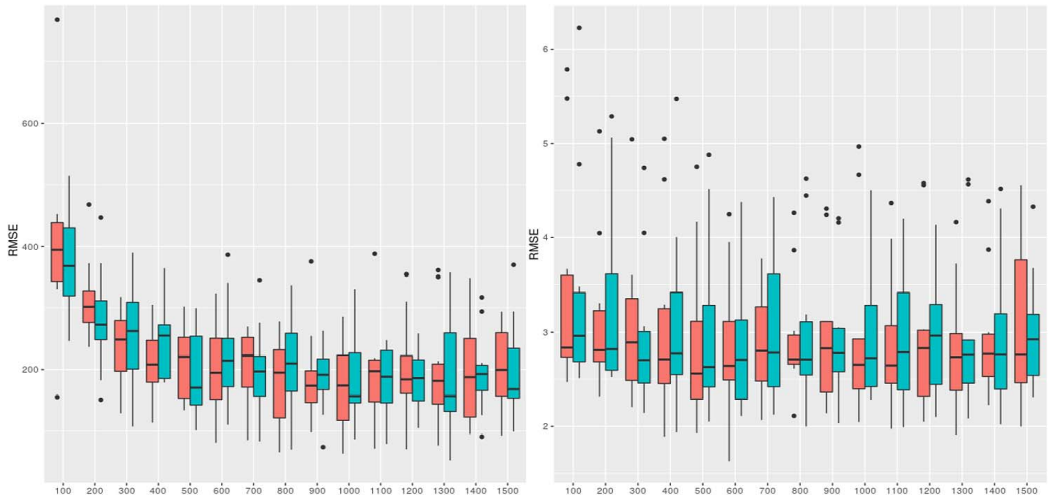


FIG. 6. *RMSE depending on the number of trees on the Diamond* (*left*) *and Boston* (*right*) *dataset for* $(\mu, \eta) = (0, 0)$ (*red*) *and* $(\mu, \eta) = (1, 2)$ (*blue*).

est. However, combining the two types of rules leads to a lower RMSE on both datasets. In our experiments it rarely hurts the performance to use both type of rules, and on some datasets it leads to a dramatically better prediction accuracy. The mixing proportion for the ensemble methods can also be seen as a tuning parameter to give a further boost in performance.

4.6. *Influence of the number of rules*. Another parameter that is potentially crucial is the number of trees used to generate the decision rules. In gradient boosting limiting the number of trees (iterations) is the most common way to control overfitting. Also in BART the number of trees has a major impact on the quality and performance of the resulting ensemble [Chipman, George and McCulloch (2010)]. The same is expected for RuleFit, as it uses L1-regularization; with an increasing number of rules the overall shrinkage τ increases, leading to an over-shrinkage of good rules.

To investigate the sensitivity of HorseRule to the number of trees, we increase the number of trees successively from 100 to 1500 in the Boston and Diamond datasets. This corresponds to $500, 550, \ldots, 5 \cdot 1500 = 7500$ decision rules before removing duplicates. We also test if the rule structured prior interacts with the effect of the number of trees by running the model with $(\mu, \eta) = (0, 0)$ and $(\mu, \eta) = (1, 2)$. Figure 6 shows the performance of HorseRule as a function of the number of trees used to extract the rules. Both HorseRule models are relatively insensitive to the choice of k, unless the number of trees is very small. Importantly, no overfitting effect can be observed, even when using an extremely large number of 1500 trees on relatively small datasets ($N = 308$ and $N = 506$ observations, respectively). We use 1000 trees as a standard choice, but a small number of trees

TABLE 8
The 10 *most important rules in the Boston housing data*

Rule		5% I	$\bar{I}$	95% I	$\bar{\beta}$
1	$RM \leq 6.97$ $LSTAT \leq 14.4$	0.96	0.99	1.00	24.1
2	$RM \leq 6.97$ $DIS > 1.22$ $LSTAT \leq 14.4$	0.77	0.89	1.00	−21.9
3	$LSTAT \leq 4.66$	0.00	0.27	0.51	12.35
4	$TAX \leq 416.5$ $LSTAT \leq 4.65$	0.00	0.21	0.43	−10.46
5	$NOX \leq 0.59$	0.00	0.12	0.21	−2.94
6	$NOX \leq 0.67$ $RM > 6.94$	0.00	0.10	0.33	3.87
7	$NOX > 0.67$	0.00	0.11	0.37	−3.24
8	$LSTAT > 19.85$	0.00	0.15	0.53	−3.18
9	$linear : AGE$	0.00	0.09	0.15	−0.03
10	$linear : RAD$	0.00	0.07	0.19	0.10

can be used if computational complexity is an issue, with little to no expected loss in accuracy.

4.7. *Boston housing*. In this section we apply HorseRule to the well known Boston Housing dataset to showcase its usefulness in getting insights from the data. For a detailed description of the dataset see Section (2.1). The HorseRule with default parameter settings is used to fit the model. Table 8 shows the 10 most important effects. Following Friedman and Popescu (2008), the importance of a linear term is defined as

$$I(x_j) = |\beta_j| \mathrm{sd}(x_j),$$

where $\mathrm{sd}(\cdot)$ is the standard deviation, and similarly for a predictor from a decision rule

$$I(r_l) = |\alpha_l| \mathrm{sd}(r_l).$$

We use the notation I_j when it is not important to distinguish between a linear term and a decision rule. For better interpretability we normalize the importance to be in $[0, 1]$, so that the most important predictor has an importance of 1. Table 8 reports the posterior distribution of the normalized importance (obtained from the MCMC draws) of the 10 most important rules or linear terms. The most important single variable is LSTAT, which appears in many of the rules, and as a single variable in the third most important rule. Note also that LSTAT does not appear as a linear predictor among the most important predictors.

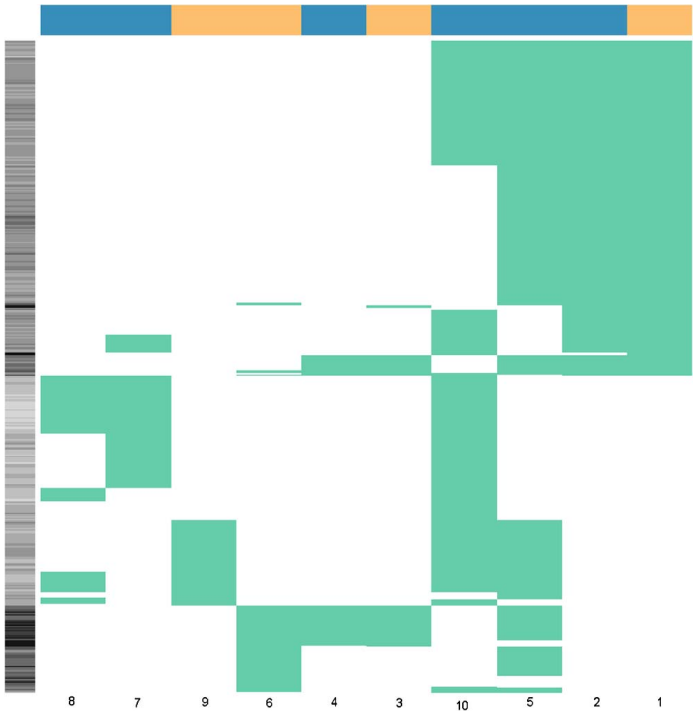


FIG. 7. *RuleHeat for the Boston housing data. See the text for details.*

To interpret the more complex decision rules in Table 8 it is important to understand that decision rules in an ensemble have to be interpreted with respect to other decision rules, and in relation to the data points covered by a rule. A useful way to explore the effects of the most important rules is what we call a *RuleHeat* plot, see Figure 7 for an example for the Boston housing data. The horizontal axis lists the most important decision rules and the vertical axis the N observations. A square is green if $r_l(\mathbf{x}) = 1$. The grayscale on the bar to the left indicates the outcome (darker for higher price) and the colorbar in the top of the figure indicates the sign of the covariate's coefficient in the model (sand for positive). RuleHeat makes it relatively easy to to find groups of similar observations, based on the rules found in HorseRule, and to assess the role a rule plays in the ensemble. For example, Figure 7 shows that the two most important rules differ only in a few observations. The two rules have very large coefficients with opposite signs. Rule 1 in isolation implies that prices are substantially higher when the proportion of lower status population is low ($LSTAT \leq 14.4$) for all but the very largest houses ($RM \leq 6.97$). However, adding Rule 2 essentially wipes out the effect of Rule 1 ($24.1 - 21.9 = 2.2$) except for the six houses very close to the employment centers ($DIS < 1.22$) where the effect on the price remains high.

Similarly to the variable importance in Random Forest and RuleFit, we can calculate a variable input importance for the HorseRule model. The importance of

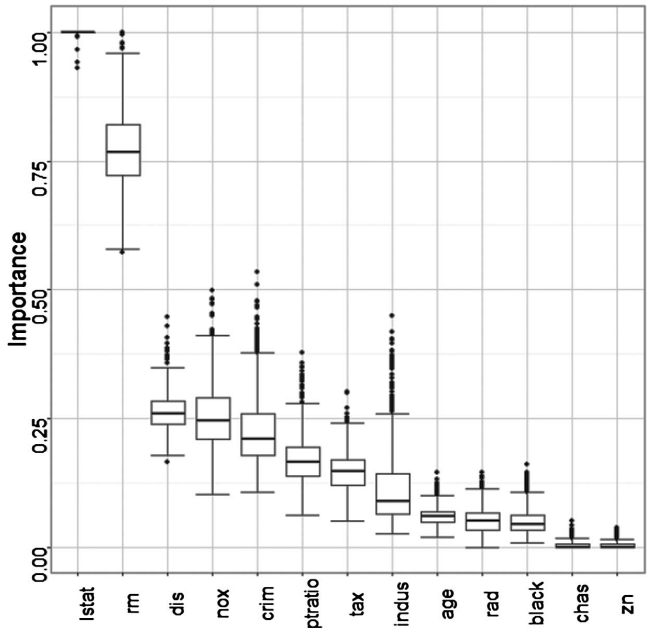


FIG. 8. *Posterior distribution of the input variable importance for the* 13 *covariates.*

the jth predictor given the data is defined as [Friedman and Popescu (2008)]

$$J(x_j) = I(x_j) + \sum_{l: j \in Q_l} I(r_l)/|Q_l|,$$

where the sum runs over all rules where x_j is one of the predictors used to define the rule. Note how the importance of the rules are discounted by the number of variables involved in the rule, $|Q_l|$. Figure 8 shows the posterior distribution of $J(x_j)$ for the 13 covariates. LSTAT is the most important covariate with median posterior probability of 1 and very narrow posterior spread, followed by RM which has a median posterior importance of around 0.75. The importance of some variables, like NOX and INDUS, has substantial posterior uncertainty whereas for other covariates, such as AGE, the model is quite certain that the importance is low (but nonzero).

The overlapping rules, as well as similar rules left in the ensemble in order to capture model uncertainty about the splitting points make interpretation somewhat difficult. One way to simplify the output from HorseRule is to use the *decoupling shrinkage and summary* (DSS) approach by Hahn and Carvalho (2015). The idea is to reconstruct the full posterior estimator $\hat{\beta}$ with a 1-norm penalized representation, that sets many of the coefficients to exactly zero and also merges together highly correlated coefficients. We do not report systematic tests here, but in our experiments using DSS with a suitable shrinkage parameter did not hurt the predictive performance, while allowing to set a vast amount of coefficients to zero. Using HorseRule followed by DSS on the Boston housing data leaves 106 nonzero

TABLE 9
The ten most important rules in Boston data after DSS

Rule		$\bar{I}$	$\bar{\beta}$
1	$RM \leq 7.13$	1.00	−3.47
2	$RM \leq 6.98$ $PTRATIO \leq 18.7$ $LSTAT > 5.95$	0.97	−2.36
3	$LSTAT > 18.75$	0.81	1.80
4	$linear : RAD$	0.80	0.10
5	$RM \leq 7.437$ $LSTAT \leq 7.81$	0.79	−2.03
6	$NOX \leq 0.62$ $RM \leq 7.31$	0.70	−1.64
7	$RM \leq 7.1$ $RAD \leq 4.5$ $LSTAT \leq 7.81$	0.68	−2.47
8	$NOX > 0.59$	0.63	−1.47
9	$linear : LSTAT$	0.58	−0.09
10	$linear : AGE$	0.58	−0.02

coefficients in the ensemble. The 10 most important rules can be seen in Table 9. We can see that the new coefficients are now less overlapping. The relatively small number of rules simplify interpretation. Posterior summary for regression with shrinkage priors is an active field of research [see, e.g., Nalenz and Villani (2018), Piironen and Vehtari (2017) and Puelz, Hahn and Carvalho (2017) for interesting approaches] and future developments might help to simplify the rule ensemble further.

4.8. *Logistic regression on gene expression data*. Here we analyze how HorseRule can find interesting pattern in classification problems, specifically in using gene expression data for finding genes that can signal the presence or absence of cancer. Such information is extremely important since it can be used to construct explanations about the underlying biological mechanism that lead to mutation, usually in the form of gene pathways. Supervised gene expression classification can also help to design diagnostic tools and patient predictions, that help to identify the cancer type in early stages of the disease and to decide on suitable therapy [Van't Veer et al. (2002)].

Extending HorseRule to classification problems can be easily done using a latent variable formulation of, for example, the logistic regression. We chose to use the Pólya–Gamma latent variable scheme by Polson, Scott and Windle (2013). Methodological difficulties arise from the usually small number of available samples, as well as high number of candidate genes, leading to an extreme $p \gg n$

TABLE 10
Accuracy in training and test set for the prostate cancer data

	BART	**Random Forest**	**RuleFit**	**HorseRule**
CV-Accuracy	0.900	0.911	0.831	0.922
CV-AUC	0.923	0.949	0.953	0.976
Test-Accuracy	0.824	0.971	0.941	0.971
Test-AUC	1	0.991	0.995	1

situation. We showcase the ability of HorseRule to make inference in this difficult domain on the Prostate Cancer dataset, which consists of 52 cancerous and 50 healthy samples ($n = 102$). In the original data $p = 12{,}600$ genetic expressions are available, which can be reduced to 5966 genes after applying the preprocessing described in Singh et al. (2002). Since spurious relationships can easily occur when using higher order interactions in the $p \gg n$ situation, we use the hyperparameters $\mu = 2$ and $\eta = 4$ to express our prior belief that higher order interactions are very unlikely to reflect any true mechanism.

Table 10 shows that HorseRule has higher accuracy and significantly higher AUC than the competing methods. We also test the methods on an unseen test dataset containing 34 samples not used in the previous step. All methods have lower error here, implying that the test data consists of more predictable cases. The difference is smaller, but HorseRule performs slightly better here as well.

The 10 most important rules for HorseRule are founds in Table 11. It contains eight rules with one condition and only two with two conditions, implying that there is not enough evidence in the data for complicated rules to overrule our prior specification. All of the most important rules still contain 0 in their 5% posterior importance distribution, implying that they are eliminated by the model in at least 5% of the samples; the small sample size leads to nonconclusive results.

Figure 9 shows the input variable importance of the 50 most important genes. In this domain the advantage of having estimates of uncertainty can be very beneficial, as biological follow up studies are costly and the probability of spurious relationships is high. In this data the genes 37,639_*at* and 556_*s*_*at* contain an importance of 1 in their 75% posterior probability bands. The gene 37,639_*at* was found in previous studies to be the single gene most associated with prostate cancer [Yap et al. (2004)]. However, gene 556_*s*_*at*, which makes up the most important Rule 1, was only found to be the ninth important in previous studies on the same data using correlation based measures [Yap et al. (2004)]. So, while this gene is individually not very discriminative (77% accuracy), it becomes important in conjunction with other rules. This is also borne out in the RuleHeat plot in Figure 10. The outcome is binary, and the vertical bar to the left is red for cancer and black for healthy. RuleHeat shows that Rule 1 covers all except one cancer tissue together with a number of normal tissues, and would therefore probably not be found to be

TABLE 11
10 *most important rules in the cancer data*

Rule		**5%** $\bar{I}$	$\bar{I}$	**95%** $\bar{I}$	$\bar{\beta}$
1	556_*s*_*at* ≤ 1.55	0	0.33	1	3.10
2	34,647_*at* ≤ -1.18	0	0.15	1	-1.78
	37,639_*at* ≤ 1				
3	37,478 > -0.32	0	0.18	0.91	1.42
4	38,087_*s*_*at* ≤ 0.83	0	0.23	1	1.81
5	34,678_*at* > 0.38	0	0.19	0.88	-1.58
6	1243_*at* ≤ 0.35	0	0.15	0.66	1.19
7	37,639_*at* ≤ 1	0	0.13	0.80	-1.10
8	33,121_*g*_*at* ≤ 0.672	0	0.10	0.82	-1.09
	960_*g*_*at* > 0.378				
9	41,706_*at* ≤ 1.33	0	0.15	0.79	-1.13
10	39,061_*at* > 0.31	0	0.1	0.52	-1.03

significant using traditional tests in logistic regression. Its importance arises from the combination with the other rules, especially Rule 2, Rule 7 and Rule 8, that are able to correct the false positive predictions using Rule 1 alone.

To illustrate HorseRule's potential for generating important insights from interaction rules, we present the subspaces of the two most important interaction rules in Figure 11 and Figure 12. Again healthy tissues are colored black and cancerous red. The first interaction looks somewhat unnatural. The gene 37,639_*at* is individ-

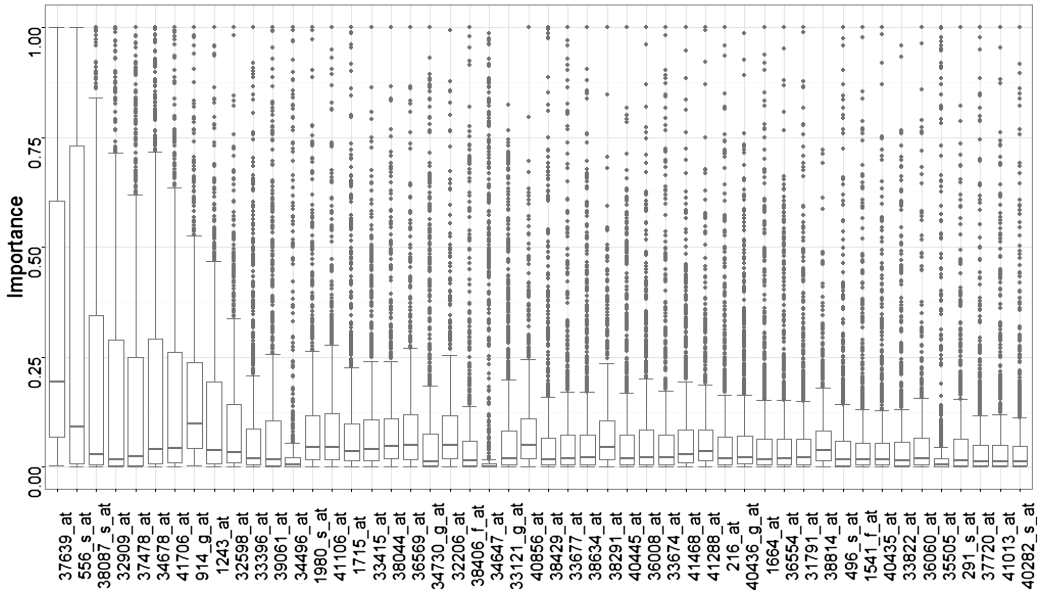


FIG. 9. *Posterior distribution of the input variable importance of the* 50 *most influential covariates.*

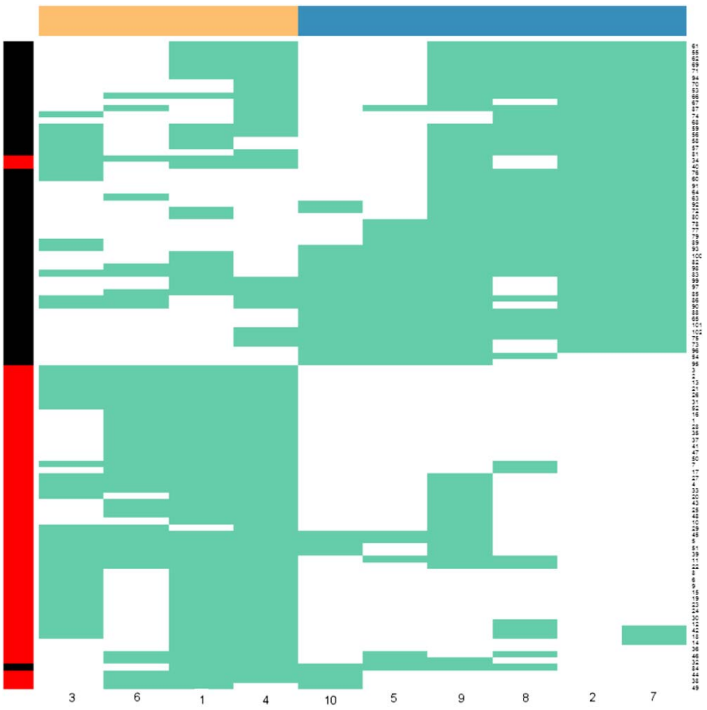


FIG. 10. *RuleHeat for the prostate cancer data. Cancer tissues are colored in red, healthy in black.*

ually seen to be a strong classifier where higher values indicate cancer. This rule is also individually represented as Rule 7. The second split on 34,647_*at* < -1.18 corrects three misclassified tissues by the first split alone. This rule probably only works well in the ensemble but may not reflect a true mechanism. The second interaction effect is more interesting. It seems that healthy tissues have lower values in the expression of 33,121_*g*_*at* and higher values in the expression of 960_*g*_*at*. This rule might reflect a true interaction mechanism and could be worth analysing further.

Overall, this shows that HorseRules nonlinear approach with interacting rules complements the results from classical linear approaches with new information. Decision rules are especially interesting for the construction of gene-pathways [Glaab, Garibaldi and Krasnogor (2010)], diagnostic tools and identification of targets for interventions [Slonim (2002)].

5. Conclusions. We propose HorseRule, a new model for flexible nonlinear regression and classification. The model is based on RuleFit and uses decision rules from a tree ensemble as predictors in a regularized linear fit. We replace the L1-regularization in RuleFit with a horseshoe prior with a hierarchical structure especially tailored for a situation with decision rules as predictors. Our prior shrinks complex (many splits) and specific (small number of observations satisfy the rule) rules more heavily a priori, and is shown to be efficient in removing noise without

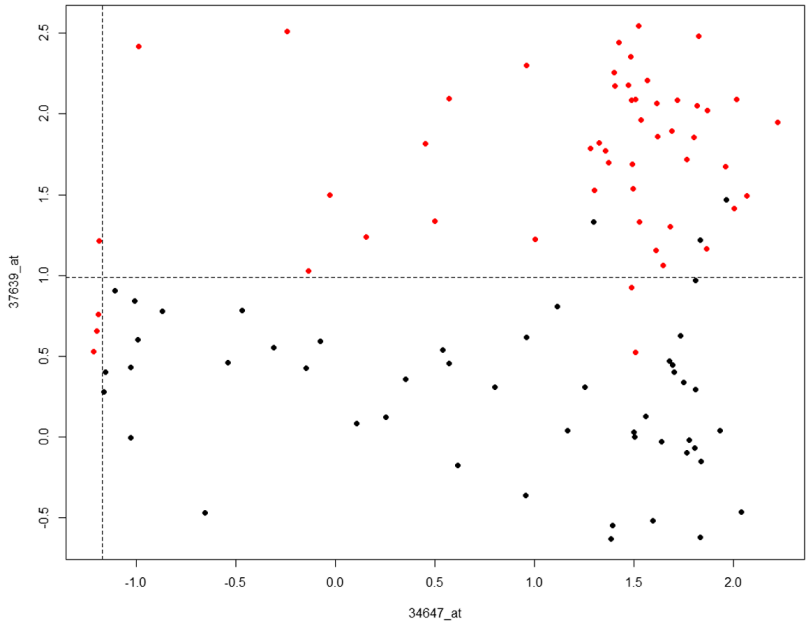


FIG. 11. *Scatterplot for Genes* 37,639*_at and* 34,647*_at. Healthy samples in black and cancerous samples in red. Rule* 2 *is defined by the bottom right quadrant.*

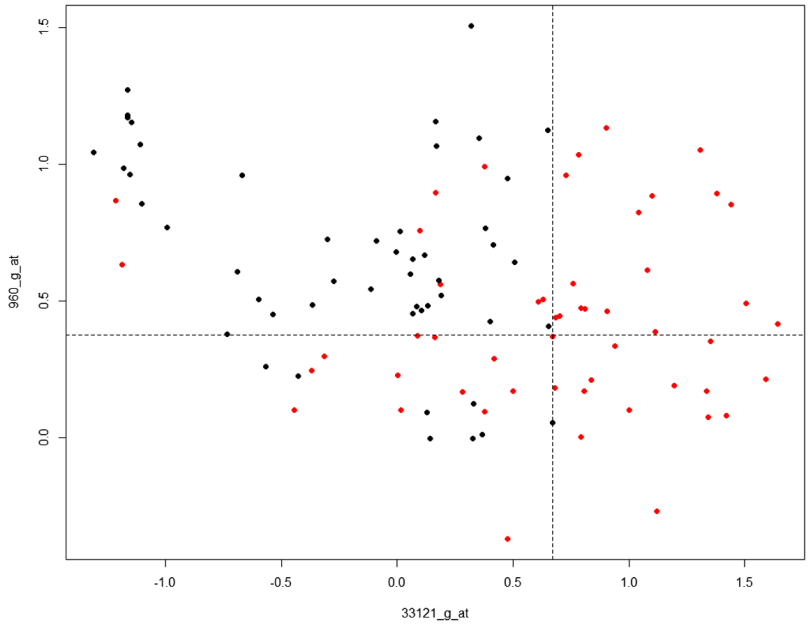


FIG. 12. *Scatterplot for Genes* 33,121*_g_at and* 960*_g_at. Healthy samples in black and cancerous samples in red. Rule* 8 *is defined by the top left quadrant.*

tampering with the signal. The efficient shrinkage properties of the new prior also makes it possible to complement the rules from boosting used in RuleFit with an additional set of rules from random forest. The rules from Random Forest are not as tightly coupled as the ones from boosting, and are shown to improve prediction performance compared to using only rules from boosting.

HorseRule is shown to outperform state-of-the-art competitors like RuleFit, BART and Random Forest in an extensive evaluation of predictive performance on 16 widely used datasets. Importantly, HorseRule performs consistently well on all datasets, whereas the other methods perform quite poorly on some of the datasets. We explored different aspect of HorseRule to determine the underlying factors behind its success. We found that the combination of mixing rule from different tree algorithms and the aggressive but signal-preserving horseshoe shrinkage are essential, but that the addition of linear terms seems less important. Our experiments also show that the predictive performance of HorseRule is not sensitive to its prior hyperparameters. We also demonstrate the interpretation of HorseRule in both a regression and a classification problem. HorseRule's use of decision rules as predictors and its ability to keep only the important predictors makes it easy to interpret its results, and to explore the importance of individual rules and predictor variables.

Acknowledgements. We are grateful to the two reviewers and the Associate Editor for constructive comments that helped to improve both the presentation and the contents of the paper.

SUPPLEMENTARY MATERIAL

The HorseRule R-package (DOI: 10.1214/18-AOAS1157SUPP; .pdf). Example code illustrating the basic features of our HorseRule package in R with standard settings. The package is available on CRAN at https://CRAN.R-project.org/package=horserule.

REFERENCES

BREIMAN, L. (1996). Stacked regressions. *Mach. Learn.* **24** 49–64.

BREIMAN, L. (2001). Random forests. *Mach. Learn.* **45** 5–32.

CARVALHO, C. M., POLSON, N. G. and SCOTT, J. G. (2009). Handling sparsity via the horseshoe. In *AISTATS* **5** 73–80.

CARVALHO, C. M., POLSON, N. G. and SCOTT, J. G. (2010). The horseshoe estimator for sparse signals. *Biometrika* **97** 465–480. MR2650751

CHEN, T. and GUESTRIN, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 785–794. ACM, New York.

CHIPMAN, H. A., GEORGE, E. I. and MCCULLOCH, R. E. (2010). BART: Bayesian additive regression trees. *Ann. Appl. Stat.* **4** 266–298. MR2758172

COHEN, W. W. (1995). Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning* (*ICML*'95) 115–123.

DEMBCZYŃSKI, K., KOTŁOWSKI, W. and SŁOWIŃSKI, R. (2010). ENDER: A statistical framework for boosting decision rules. *Data Min. Knowl. Discov.* **21** 52–90. MR2720513

FREUND, Y. and SCHAPIRE, R. E. (1996). Experiments with a new boosting algorithm. In *Machine Learning*: *Proceedings of the Thirteenth International Conference* **96** 148–156. Bari, Italy.

FRIEDMAN, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Ann. Statist.* **29** 1189–1232. MR1873328

FRIEDMAN, J. H. and POPESCU, B. E. (2003). Importance sampled learning ensembles. Technical report, Dept. Statistics, Stanford Univ., Stanford, CA.

FRIEDMAN, J. H. and POPESCU, B. E. (2008). Predictive learning via rule ensembles. *Ann. Appl. Stat.* **2** 916–954. MR2522175

FÜRNKRANZ, J. (1999). Separate-and-conquer rule learning. *Artif. Intell. Rev.* **13** 3–54.

GEORGE, E. I. and MCCULLOCH, R. E. (1993). Variable selection via Gibbs sampling. *J. Amer. Statist. Assoc.* **88** 881–889.

GLAAB, E., GARIBALDI, J. M. and KRASNOGOR, N. (2010). Learning pathway-based decision rules to classify microarray cancer samples.

HAHN, P. R. and CARVALHO, C. M. (2015). Decoupling shrinkage and selection in Bayesian linear models: A posterior summary perspective. *J. Amer. Statist. Assoc.* **110** 435–448. MR3338514

KE, G., MENG, Q., FINLEY, T., WANG, T., CHEN, W., MA, W., YE, Q. and LIU, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems* 3149–3157.

LI, L. and YAO, W. (2014). Fully Bayesian logistic regression with hyper-LASSO priors for high-dimensional feature selection. *J. Stat. Comput. Simul.* **88** 2827–2851. MR3827411

LINERO, A. R. (2018). Bayesian regression trees for high dimensional prediction and variable selection. *J. Amer. Statist. Assoc.* **113** 626–636. MR3832214

MAKALIC, E. and SCHMIDT, D. F. (2016). A simple sampler for the horseshoe estimator. *IEEE Signal Process. Lett.* **23** 179–182.

NALENZ, M. and VILLANI, M. (2018). Supplement to "Tree ensembles with rule structured horseshoe regularization." DOI:10.1214/18-AOAS1157SUPP.

PIIRONEN, J. and VEHTARI, A. (2017). Comparison of Bayesian predictive methods for model selection. *Stat. Comput.* **27** 711–735. MR3613594

POLSON, N. G., SCOTT, J. G. and WINDLE, J. (2013). Bayesian inference for logistic models using Pólya–Gamma latent variables. *J. Amer. Statist. Assoc.* **108** 1339–1349. MR3174712

PUELZ, D., HAHN, P. R. and CARVALHO, C. M. (2017). Variable selection in seemingly unrelated regressions with random predictors. *Bayesian Anal.* **12** 969–989. MR3724975

ROKACH, L. (2010). Ensemble-based classifiers. *Artif. Intell. Rev.* **33** 1–39.

SCHAPIRE, R. E. (1999). A brief introduction to boosting. In *IJCAI* 1401–1406.

SINGH, D., FEBBO, P. G., ROSS, K., JACKSON, D. G., MANOLA, J., LADD, C., TAMAYO, P., RENSHAW, A. A., D'AMICO, A. V., RICHIE, J. P. et al. (2002). Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell* **1** 203–209.

SLONIM, D. K. (2002). From patterns to pathways: Gene expression data analysis comes of age. *Nat. Genet.* **32 (Supp)** 502.

SMITH, M. and KOHN, R. (1996). Nonparametric regression using Bayesian variable selection. *J. Econometrics* **75** 317–343.

TERENIN, A., DONG, S. and DRAPER, D. (2016). GPU-accelerated Gibbs sampling. Preprint. Available at arXiv:1608.04329.

TIBSHIRANI, R. (1996). Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B* **58** 267–288. MR1379242

VAN'T VEER, L., DAI, H., VAN DE VIJVER, M. J., HE, Y. D., HART, A. A. M., MAO, M., PETERSE, H. L., VAN DER KOOY, K., MARTON, M. J., WITTEVEEN, A. T. et al. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature* **415** 530–536.

WOLPERT, D. H. (1992). Stacked generalization. *Neural Netw.* **5** 241–259.

YAP, Y., ZHANG, X., LING, M. T., WANG, X., WONG, Y. C. and DANCHIN, A. (2004). Classification between normal and tumor tissues based on the pair-wise gene expression ratio. *BMC Cancer* **4** 72.

ZOU, H. and HASTIE, T. (2005). Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **67** 301–320. MR2137327

DIVISION OF STATISTICS AND MACHINE LEARNING
DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE
LINKÖPING UNIVERSITY
SE-581 83 LINKÖPING
SWEDEN
E-MAIL: malte.nlz@googlemail.com
mattias.villani@liu.se
URL: http://mattiasvillani.com

Compressed Rule Ensemble Learning

Preprint – Under Review

Malte Nalenz
Department of Statistics
Ludwig-Maximilians-University
Munich

Thomas Augustin
Department of Statistics
Ludwig-Maximilians-University
Munich

Abstract

Ensembles of decision rules extracted from tree ensembles, like RuleFit, promise a good trade-off between predictive performance and model simplicity. However, they are affected by competing interests: While a sufficiently large number of binary, non-smooth rules is necessary to fit smooth, well generalizing decision boundaries, a too high number of rules in the ensemble severely jeopardizes interpretability. As a way out of this dilemma, we propose to take an extra step in the rule extraction step and compress clusters of similar rules into *ensemble rules*. The outputs of the individual rules in each cluster are pooled to produce a single soft output, which reflects the marginal smoothing behaviour of the original ensemble. The final model, that we call *Compressed Rule Ensemble* (CRE), fits a linear combination of ensemble rules. On a variety of datasets we show empirically that CRE is both sparse and accurate, carrying over the ensemble behaviour, while remaining interpretable. CRE delivers predictive performance on par with state-of-the-art tree ensemble methods but with a model size that is substantially smaller compared to previous rule ensemble approaches. Predictions can be explained by looking at the active ensemble rules, which allows external validation. We showcase that ensemble rules are also useful for a wider range of models that utilize decision rules extracted from tree ensembles.

1 Introduction

Ensemble methods that use decision trees as base learners are among the most popular and successful general purpose supervised learning methods. They can naturally adapt to non-linearities, capture interactions between features and often perform well off-the-shelf with little to no parameter tuning (Fernandez-Delgado et al., 2014). Most tree ensemble methods use re-sampling schemes in order to create trees that capture different aspects of the training data. This increased model variance in return leads to more stable, robust and accurate predictions compared to a single decision tree.

However, the increase in model complexity resulting from the ensemble approach is also a major downside. While a single decision tree is straightforward to interpret, a forest resulting from the combination of hundreds, deep and randomized, decision trees, can not be processed by the human mind, essentially turning the ensemble into a black box

model. Methods for analysing the behaviour of the forest exist, such as Variable Importance (Breiman, 2001) and recently proposed variants, such as SHAP-values (Lundberg and Lee, 2017), but the intuitive structure of the individual trees is lost. This makes it unclear, how exactly a decision is reached, which is a fact that is often not acceptable in high-stake situations, such as a medical treatment choice.

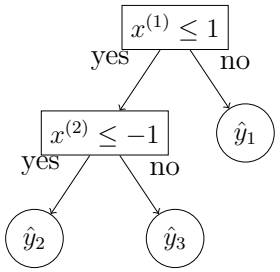
One approach towards interpretable machine learning models is to learn rule ensembles. As decision rules are composed of simple if-else statements, they are easier to interpret for humans, compared with deep decision trees. One such approach is RuleFit introduced by (Friedman and Popescu, 2008). Instead of learning decision rules directly, the candidate rules are extracted from decision forests and combined in a penalized linear model. The rationale of RuleFit is both simple and compelling: Tree ensemble methods often have remarkable accuracy. However, their greedy learning procedure produces overly complicated models. By regularizing away the unnecessary complexity, RuleFit promises a step towards a favourable accuracy-complexity trade-off.

We argue that rule ensemble approaches suffer from competing interests: In order to provide smooth decision boundaries, a property essential for good generalization in ensembles (Bühlmann and Yu, 2002; Bühlmann, 2012), a sufficiently large number of slightly different – potentially overlapping – rules need to be selected for the final ensemble, which in return harms the interpretability. We propose a way to solve this dilemma, based on the interpretation of ensemble learning as a smoothing of the hard thresholding behaviour of decision rules (Bühlmann and Yu, 2002). Instead of using the individual decision rules directly, we first identify clusters of similar conditions. To this end univariate clustering is performed on the splitpoints in each covariate. The resulting groups of similar conditions are then compressed into soft conditions, that we call *ensemble conditions*. By averaging the discrete outputs of the individual conditions, ensemble conditions produce a smooth output, that reflects the behaviour of the original forest method. Ensemble rule compression allows to carry over the smoothing behaviour of forest methods while sacrificing very little in terms of interpretability and reflecting the uncertainty about the 'true' splitpoint. We argue that often already a few compressed rules allow to capture and interpret the central behaviour of the forest, allowing a glimpse into the black box.

The structure of this paper is as follows. In section 2 we give an overview of existing rule ensemble approaches, and in section 3 we review the RuleFit approach and introduce notations. Section 4 introduces *compressed rule ensembles (CRE)*, that combine ensemble rules, based on ensemble conditions, with the RuleFit approach. We also showcase that ensemble rules are useful in other rule ensemble frameworks. Section 5 presents our experiments on classification tasks. Section 6 concludes.

2 Related work

Several different ways have been proposed to (greedily) induce decision rule ensembles. Classical approaches include the divide and conquer algorithms, that sequentially induces non-overlapping rules (Cohen, 1995; Fürnkranz, 1999), and boosted decision rules (Freund and Schapire, 1996; Weiss and Indurkhya, 2000; Dembczyński et al., 2008) that use re-weighting schemes to induce rules that iteratively reduce the error from the current



$\phi(c)$	s	v	t
$I(x^{(1)} \leq 1)$	1	1	1
$I(x^{(1)} > 1)$	2	1	1
$I(x^{(1)} \leq 1)$	3	1	1
$I(x^{(2)} \leq -1)$	3	2	-1
$I(x^{(1)} \leq 1)$	4	1	1
$I(x^{(2)} > -1)$	4	2	-1

Figure 1: Left: Binary decision tree with 3 leafs, 1 internal node and the root node. Right: Further decomposition of the decision rules into the elementary conditions. Multiple conditions per rule are combined with the logical AND.

ensemble.

RuleFit (Friedman and Popescu, 2008) combines candidate rules in a penalized linear model. This two-step formulation of rule learning allows the application of standard statistical learning methods. In its original formulation rules are extracted from gradient boosted decision trees (Friedman, 2002), but also other forest types have been explored (Nalenz and Villani, 2018; Fokkema, 2020). Inducing decision rules jointly with learning the weight coefficients was explored in (Jawanpuria et al., 2011) and (Wei et al., 2019). Using quadratic programming to select the final ruleset was explored in (Meinshausen, 2010).

Another interesting way to combine decision rules was recently proposed with SIRUS (Bénard et al., 2021), where paths are extracted from an adapted version of random forests: the data is quantile transformed beforehand, to limit the possible splitpoints in trees, allowing to identify frequent pattern across trees. The most common decision rules are simply averaged to produce a prediction, without the need of a linear combination, which improves the model stability significantly.

3 Predictive rule ensembles

Given the N training examples $(y_i, x_i), i = 1, \ldots, N$, with generic variables y and x, where y is either discrete or numeric and $x = (x_1, ..., x_p) \in \mathbb{R}^p$ is the p-dimensional covariate vector, with the j'th component of x denoted as $x^{(j)}$, we seek to find a function, that allows to predict y from x. In the context of predictive rule ensembles and assuming a regression task we look at the class of generalized additive models

$$y = \sum_{h=1}^{H} \alpha_h r_h(x), \quad r_h \in \{0, 1\} \tag{1}$$

where decision rules $r_h(x)$ are used as basis functions, weighted by the coefficients α_h. Instead of learning decision rules directly from the data, the RuleFit framework, takes a two-step procedure.
First, a (greedy) tree ensemble is generated. (Friedman and Popescu, 2008) use gradient boosting to generate the set of trees. As boosting shows great accuracy in many tasks, it

is reasonable to assume that the model is able to find interesting subspaces, defined by decision rules. The decision trees are then decomposed into its defining decision rules and harvested across the whole ensemble. In our approach we decompose the rules further into their elementary conditions. s denotes the index of the rule that the condition originates from, v the index of the covariate used for the comparison and t the splitpoint. Figure 1 shows an example of this decomposition and the introduced notation. A condition is thus defined by the triplet $c_a = (s_a, v_a, t_a), a = 1, ..., A$, where A is the total number of conditions collected from the forest and $M = |\{s\}|$ is the number of decision rules. Note that in this step only the different paths to all nodes are stored, not the values of the leaf nodes. The full rule is the conjunction of its individual conditions. The hard-thresholding split function ϕ is given by

$$\phi(x, v, t) = I(x^{(v)} < t) \quad \text{or} \tag{2}$$

$$\phi(x, v, t) = I(x^{(v)} \geq t) = 1 - I(x^{(v)} < t) \tag{3}$$

depending on the direction that is encoded in r_h and assuming numerical features. As the second step the decision rules are included, together with linear terms, as 0-1 features in a linear regression model. Using the above notations, the full rules $r_h \in \{0, 1\}$ are obtained by taking the product of the conditions that are part of rule h,

$$r_h(x) = \prod_{a:s_a=h} \phi(x, v_a, t_a). \tag{4}$$

As the original forest contains a large number of rules, L1-penalization (Tibshirani, 1996) is used to shrink the large set of candidate rules down to the truly predictive ones.

4 Compressed rule ensembles (CRE)

As in previous versions of RuleFit, as a first step a tree ensemble is generated. Either the random forest or gradient boosting framework can be applied. For computational efficiency we use XGBoost (Chen and Guestrin, 2016) to generate the rules. However, before transforming the rules directly into 0-1 features using (4), we take an additional step and compress groups of similar conditions into ensemble conditions.

4.1 Ensemble compression

When using re-sampling techniques as is commonly featured in both random forests and (stochastic) gradient boosting, the split points inside the forests will often appear in clusters. Depending on the sample that is seen by a tree and the weights in this iteration, the (greedy) tree induction algorithm will often chose similar trees with slightly different splitpoints. This is generally beneficial in terms of predictive performance, as it leads to a smooth decision boundary (Bühlmann and Yu, 2002), which stabilizes predictions. This also implies that when removing many rules in RuleFit we expect the decision boundaries to become non-smooth and the predictive performance to drop. The goal is therefore to preserve smoothness, but reshape it in a form that is accessible for human interpretation.

4.2 Clustering of similar conditions

To preserve the forests behaviour, we identify clusters of similar conditions that only differ in their exact splitpoint and combine their binary decision into a single smooth decision.

More formally, for each covariate $j, j = 1, ..., p$ we look at the vector of splitpoints $T^{(j)} = (t_a : v_a = j)$. This step collects all splitpoints from splits involving covariate j from all rules that were extracted from the original forest. Note that the splitpoints are taken from single condition rules or from more complicated rules, involving several conditions and other covariates. Also no attention is drawn to the depth of the rule in which the condition appears, as with the symmetry in the conjunctive form of decision rules, ordering is somewhat arbitrary. As we expect the clusters of splitpoints to be fairly obvious, we use k-means as a robust and well understood clustering method to find the clusters. We assume that the splitpoints will appear in a relatively small number of cluster. The k centers in the k-means algorithm are chosen to minimize the intra-cluster variation,

$$C(k, \mu, T^{(j)}) = \sum_{l=1}^{k} \sum_{\{z: g_z^{(j)} = l, t_z \in T^{(j)}\}} (t_z - \mu_l)^2 \tag{5}$$

where $z \in \{1, ..., Z = |T^{(j)}|\}$ is the index of splitpoints for covariate j, $g^{(j)} = (g_1, ..., g_Z)$ is the vector of clusterlabels for the splitpoints and $\mu = (\mu_1, ..., \mu_k)$ the vector of mean values of the k groups. For this one-dimensional clustering problem the *Ckmeans.1d.dp* algorithm (Wang and Song, 2011) can be applied, that uses dynamic programming to find the global optimal solution. If a certain splitpoint is very important in the prediction task, it will often appear in the vector $T^{(j)}$ and dominate the cluster solution in equation (5). This is a desired property, as it results in an implicit weighting of regions found important by the forest method. As the appropriate number of clusters for each covariate is unknown a-priori, we determine the optimal k using the AIC criterion with a pre-specified maximum number of clusters k_{max}. Other clustering algorithms that we considered were Gaussian-Mixture-Models and density based clustering methods, such as DBSCAN (Schubert et al., 2017). As the results were quite similar, we decided to stick with k-means as the most simple and robust approach. Note that the clustering is performed on the splitpoints found in the forest, never on the original data, making this step computationally cheap. For 500 trees the number of splits is typically $<<$ 1000 per covariate and therefore almost independent from N and only linear in p.

4.3 Combining multiple conditions into a soft condition.

Given the vectors of splitpoints for group l of covariate j from the clustering step, $T_l^{(j)} = (t_i \in T^{(j)} : g_i^{(j)} = l)$, we combine the individual conditions to ensemble conditions. The combined output of the ensemble condition is computed by average pooling of the outputs from the individual conditions. The soft output function for ensemble condition l becomes

$$\Phi(x, v, l) = |T_l^{(v)}|^{-1} \sum_{t \in T_l^{(v)}} \phi(x, v, t). \tag{6}$$

Averaging over several conditions turns the individual binary outputs $\phi(x, v, t) \in \{0, 1\}$ into a soft output $\Phi(x, v, l) \in [0, 1]$. In contrast to other soft decision rule approaches, such as Akdemir et al. (2013), our approach is non-parametric. Φ reflects the empirical distribution from the splits found in the forest and preserves the univariate behaviour of the full ensemble and compresses it in a single ensemble condition. Figure 2 shows the distribution of splitpoints for the Diabetes dataset from UCI repository (Dua and Graff,

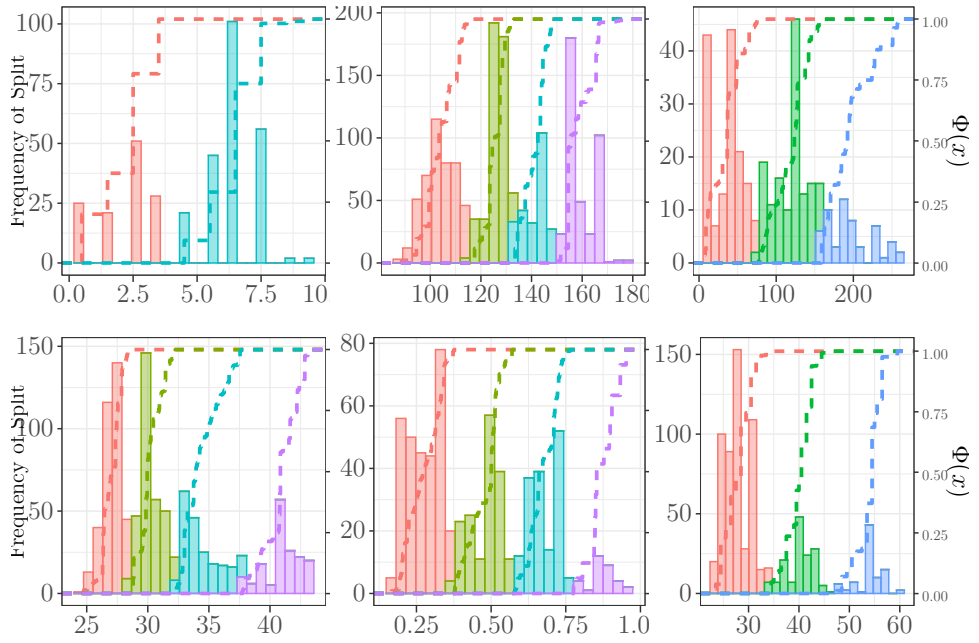


Figure 2: Distribution of splitpoints for the 6 most used covariates in the diabetes data set, using $k_{max} = 4$. Colours indicate the cluster solution from the *Ckmeans.1d.dp* algorithm. The dashed line shows the soft output $\Phi(x)$ for each cluster (compressed condition).

2017) and the clustering result using $k_{max} = 4$. We can see that the $\Phi(x)$ can follow arbitrary distributions. Also note the dense regions, forming clusters of splitpoints that are interesting for predictions. Lastly, if the underlying relationship is in fact a stepfunction, we expect the forest method to also be able to capture it and in return, the intervals of the clusters will become very narrow.

To finish this step, all original conditions are replaced by their corresponding ensemble conditions. Using ensemble conditions turns each binary rule r_h into a smooth rule $\mathcal{R}_h$ generalizing equation (4). The output of $\mathcal{R}_h$ is calculated via

$$\mathcal{R}_h(x) = \prod_{a:s_a=h} \Phi(x, v_a, g_a) \in [0, 1]. \tag{7}$$

As all conditions in each cluster have the same output for any given x_i, this allows to remove a large number of redundant rules.

4.4 Finding a sparse set

Given the ensemble rules, the second step combines them to a reduced ensemble. We investigate two ways of rule aggregation, weighting and averaging.

Linear Weighting Following RuleFit, the ensemble rules are included, together with linear terms, in the (generalized) linear regression model:

$$F(x) = \sigma(\beta_0 + \sum_{j=1}^{p} \beta_j x_j + \sum_{h=1}^{H} \alpha_h \mathcal{R}_h(x)), \tag{8}$$

where σ is a link function. As in (Friedman and Popescu, 2008) the rule terms are not scaled, leading to a higher penalty on rules with low support. However one property specific to compressed rules is that rule support decreases slower with additional conditions, leading to lower penalization of complicated rules. As complicated rules are highly undesireable in terms of interpretability, we counteract this effect by decreasing the scale of each $\mathcal{R}_h$ proportional to the number of conditions involved, via

$$\mathcal{R}_h^*(x) = \frac{\mathcal{R}_h(x)}{length(\mathcal{R}_h)^\eta}, \quad \eta > 0, \tag{9}$$

where η is a parameter that controls the amount of extra penalty for the number of conditions involved and $length(\mathcal{R}_h)$ is the number of conditions. This is similar to the rule structured prior used in (Nalenz and Villani, 2018). Penalizing depth was also found an effective way to promote simplicity in (Wei et al., 2019; Chipman et al., 2010). We found $\eta = 0.5$ to work well as a default choice, but η can also be guided by prior knowledge, about the complexity of the underlying relationship or tuned via cross validation. The weights are found by solving the L1-regularized regression

$$\{\alpha^*, \beta^*, \beta_0^*\} = \arg\min_{\beta_0,\beta,\alpha} \left[L(y, F(x)) + \lambda \left(\sum_{j=1}^{p} |\beta_j| + \sum_{h=1}^{H} |\alpha_h| \right) \right], \tag{10}$$

with L being an appropriate loss function. A big advantage of the linear model approach is its easy interpretability. Following (Friedman and Popescu, 2008), we can rank rules and linear terms by their (rescaled) effect size $|\alpha^*|$ and $|\beta^*|$ respectively as a measure of importance. In our experiments we use the `R`-package (R Core Team, 2021) *glmnet* (Friedman et al., 2010) and the penalty parameter λ is chosen via cross-validation (CV). A popular choice is to use λ_{1se} the highest λ value within one standard deviation of the minimum, in order to promote sparsity, which is also used for CRE.

Averaging An alternative to the linear combination (8) is to simply count the number of occurrences of each smooth rule $\mathcal{R}_h$ and average over the most frequent rules. For each rule the associated prediction values for cases that are covered/not covered by a rule, μ_+, μ_- respectively are the weighted mean on the training data

$$\mu_{+,h} = \frac{1}{\sum_{i=1}^{N} \mathcal{R}_h(x_i)} \sum_{i=1}^{N} y_i \mathcal{R}_h(x_i), \tag{11}$$

$$\mu_{-,h} = \frac{1}{\sum_{i=1}^{N} (1 - \mathcal{R}_h(x_i))} \sum_{i=1}^{N} y_i (1 - \mathcal{R}_h(x_i)), \tag{12}$$

assuming $y \in \{0, 1\}$, which is a soft version of the SIRUS algorithm. Predictions for the whole ensemble rule are obtained via

$$\hat{y}_{i,h} = \mathcal{R}_h(x_i, v, g) \cdot \mu_{+,h} + (1 - \mathcal{R}_h(x_i, v, g)) \cdot \mu_{-,h}. \tag{13}$$

The output of the whole ensemble is then simply the average of the K most frequent compressed rules $\mathcal{R}_h$. Adopting the SIRUS approach (Bénard et al., 2021) to use compressed conditions instead of normal decision rules goes together quite naturally with the idea of ensemble compression, avoiding any data discretization. We find the core idea of SIRUS particularly interesting, as it can be seen as a proxy of how good a whole tree ensemble can be summarised by a small number of ensemble rules.

4.5 The effect of of ensemble compression

Choice of k_{max} The inverse k_{max}^{-1} can be interpreted as compression rate. Setting $k_{max} = 1$ compresses all splitpoints per covariate into a single group and results in a monotonic transformation of the covariate, based on the distribution of the splitpoints. In this setting only monotonic effects can be captured and no change of sign is possible. Increasing k_{max} allows changes in sign and magnitudes of the effects, therefore finding different regions of interest. As $k \to Z$, where Z is the number of splitpoints in this covariate, our model approaches the original RuleFit model. Using ensemble compression also acts as a regularizer, as it makes it harder to overfit on individual rules, but has to take into account the general pattern found by the forest. We found a relatively small value of k_{max} (e.g. $k_{max} = 4$) is usually a good choice as discussed below.

Computational cost. The number of unique conditions is reduced to a maximum of k_{max} distinct ensemble conditions for each covariate, which can be significantly lower than the number of distinct original conditions. This also leads to a much smaller number of unique rules in the linear modelling step, which is important, as the design matrix in equation (10) is of size $(n, p + H)$. As duplicates and colinear terms can be safely removed, this effectively lowers the computational cost and memory usage significantly, and decreasing k_{max} lowers the computation time considerably. On datasets where the number of predictive covariates is relatively small, as is in the Diabetes data, around 60 % of $\mathcal{R}_h$ can be removed from the initial set (with $k_{max} = 4$). If the splits distribute more evenly over a large number of covariates, the reduction is still notable but less pronounced.

5 Results

In this section we test our method empirically. The goal is to show that CRE is able to produce both accurate and small models, due to the smooth boundaries introduced by the ensemble compression. R-code to reproduce all results will be made available upon publication.

5.1 Experimental setup

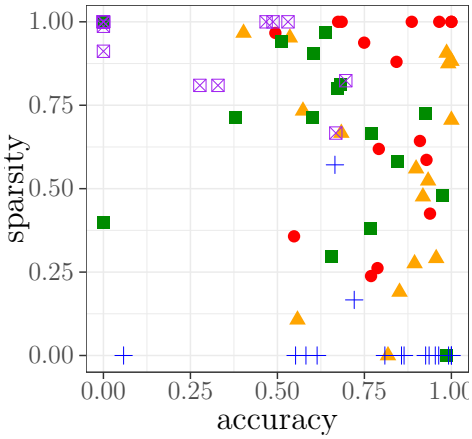
For comparison we use 16 binary classification datasets from the UCI repository (Dua and Graff, 2017). We chose datasets that consist of mostly numerical covariates and require minimal preprocessing. A detailed description of selection criteria and preprocessing, algorithm settings and additional results can be found in the supplementary material (SM). We limit the experiments to binary classification but note that CRE can also be extended to regression, multi-label and multi-target classification (Aho et al., 2012).

Table 1: Accuracy measured in AUC for the competing methods on the 16 benchmark datasets.

dataset	CRE_S	$CRE_{k:2}$	$CRE_{k:4}$	$CRE_{k:6}$	CRE_{RF}	PRE	RF	RuleFit	SIRUS	XGB
Australian	0.901	0.937	0.939	0.944	0.935	0.930	0.936	0.938	0.921	0.939
Banknote	0.986	1	1	1	1	1.000	1.000	1.000	0.972	1
Biodeg	0.871	0.929	0.931	0.930	0.919	0.915	0.938	0.924	0.840	0.932
Blood Transf	0.732	0.730	0.750	0.748	0.741	0.724	0.668	0.751	0.708	0.746
Diabetes	0.823	0.830	0.830	0.831	0.832	0.829	0.825	0.840	0.807	0.841
Haberman	0.712	0.683	0.670	0.680	0.621	0.676	0.685	0.708	0.651	0.688
Heart	0.898	0.910	0.896	0.896	0.888	0.877	0.905	0.897	0.899	0.906
ILPD	0.709	0.728	0.723	0.718	0.704	0.735	0.752	0.706	0.729	0.724
Ionosphere	0.955	0.963	0.964	0.965	0.954	0.965	0.981	0.968	0.941	0.970
Liver	0.649	0.666	0.679	0.644	0.667	0.623	0.564	0.657	0.644	0.654
Parkinsons	0.906	0.945	0.959	0.950	0.968	0.857	0.953	0.960	0.888	0.962
Pop Failure	0.907	0.947	0.945	0.952	0.946	0.947	0.920	0.925	0.889	0.946
Sonar	0.863	0.923	0.927	0.910	0.925	0.875	0.949	0.915	0.829	0.940
Spambase	0.963	0.985	0.986	0.986	0.985	0.980	0.987	0.985	0.933	0.988
WBCD	0.991	0.992	0.993	0.991	0.993	0.992	0.992	0.989	0.981	0.995
Wilt	0.952	0.990	0.992	0.991	0.993	0.991	0.990	0.993	0.901	0.990
Mean Rank	7.688	4.875	4.125	4.750	5.188	7	4.875	4.812	8.750	2.938
Mean ΔAUC	0.033	0.012	0.011	0.014	0.018	0.027	0.019	0.012	0.051	0.008

Table 2: Left: Number of coefficients selected for the final model. Right: Figure 3: Normalized Accuracy vs. Normalized Sparsity where 1 is the best on each dataset and 0 the worst. $CRE_{k:2}$ (red circle), $CRE_{k:4}$ (orange triangle), PRE (green square), RuleFit (blue cross), SIRUS (purple squares).

dataset	$CRE_{k:2}$	$CRE_{k:4}$	$CRE_{k:6}$	CRE_{RF}	PRE	RuleFit	SIRUS
Australian	18	26	32	19	20	40	15
Banknote	11	21	29	18	45	45	14
Biodeg	52	62	69	51	50	106	22
Blood Transf	7	8	8	6	9	22	6
Diabetes	9	18	23	10	28	36	9
Haberman	2	3	3	2	4	23	6
Heart	13	17	18	18	22	28	18
ILPD	10	10	15	9	10	68	8
Ionosphere	33	40	43	22	23	27	15
Liver	7	9	10	8	8	24	10
Parkinsons	22	25	27	20	14	35	18
Pop Failure	29	38	40	27	25	46	17
Sonar	50	61	61	42	31	54	19
Spambase	95	112	119	121	75	149	22
WBCD	31	32	36	24	28	36	15
Wilt	16	23	29	23	55	91	17
Mean Rank	2.75	4.44	5.47	2.88	3.81	6.62	2.03



5.2 Competing methods

As a black box baseline with generally strong predictive performance we include random forests and gradient boosting. Random forest (RF) is run with default settings using the original `randomForest` R-package (Breiman, 2001). Gradient boosting, implemented with the `xgboost` R-package (Chen and Guestrin, 2016), is more dependent on parameter tuning. We use model based optimization with `mlrMBO` (Bischl et al., 2017) inside each fold, in order to find reasonable parameters and ensure a fair comparison.

We compare against two versions of RuleFit, both implemented with the `pre` R-package

(Fokkema, 2020). RuleFit uses normal CART trees as base learners and parameter settings that most closely resemble the original version of RuleFit. In order to determine a reasonable tree depth, we use 5-fold CV inside each fold. PRE is a more interpretable setting proposed in (Fokkema, 2020) that uses λ_{1se} and an average treedepth of 3 (without tuning).

SIRUS is built using the `sirus` (Bénard et al., 2021) R-package, with the number of rules determined using the CV strategy proposed by the authors and implemented in the package.

The CRE based models use gradient boosted trees from `xgboost` to generate the rules. Different degrees of compression are tested, using $k_{max} = \{2, 4, 6\}$ denoted as $CRE_{k:k_{max}}$. Only CRE_{RF} uses random forests to generate the rules, as a way to measure the influence of the tree generating process and $k_{max} = 4$. All CRE models use $\eta = 0.5$ (cf. (9)) to promote taking in less complex rules. No parameters, for the rule generation or η, are tuned. Better predictive performance may be reached, but in this article we are interested in the 'out-of-the-box' performance. We also test compressed rules with averaging of the rules, which resembles the SIRUS approach. To estimate the influence of the rule compression, CRE_S uses on each dataset the average number of rules used by SIRUS, leading to the overall same model complexity as SIRUS. We expect the same number of ensemble rules to generalize better compared to normal rules.

Accuracy We report accuracy as measured by the area under the curve (AUC). Table 2 shows the results over the 16 datasets, together with the mean rank and average deviation from the best AUC value over all datasets. In line with our expectation, and previously reported results, a well tuned XGB model is on average the most accurate. $CRE_{k:4}$ achieves the second best rank, outperforming all rule based competitors, the vanilla random forest and the tuned RuleFit model. It is interesting to note that $CRE_{k:2}$ is still competitive in terms of accuracy to random forest and RuleFit and outperforming most of the other rule based competitors. Using a higher compression parameter $CRE_{k:6}$ is not beneficial in the analysed datasets. This can be contributed to the regularizing effect of ensemble compression, making $k_{max} = 4$ our recommended default choice for prediction. SIRUS, CRE_S and PRE are on average less accurate. As seen by the average deviation from the best method, CRE models and RuleFit are on average not much behind the best method, implying a stable performance. The same is not true for SIRUS, CRE_S and PRE, which sacrifice on average a notable amount of accuracy.

Model Complexity Table 2 shows the number of selected rules and linear terms. In terms of model complexity SIRUS, $CRE_{k:2}$, CRE_{RF} and PRE produce the smallest models, with SIRUS being the winner. However, as discussed above, SIRUS and PRE have to sacrifice an substantial amount of accuracy to achieve this goal, whereas $CRE_{k:2}$ on most datasets produces similarly sparse models, while remaining competitive in predictive performance. $CRE_{k:4}$ takes in slightly more rules, but also produces reasonably small models on most datasets, whereas also showing strong accuracy. RuleFit produces overall the largest models.

Accuracy vs. Sparsity We conclude that CRE produces models that are both accurate and sparse, whereas all of the competing methods have to compromise either

Table 3: Model output for the Diabetes data.

Rule	β
Intercept	−1.41
age $\geq$ [21.5;26.5] $\wedge$ BMI $\geq$ [21.75;28.15]	0.57
age $\geq$ [27.5;34.5]	0.47
BMI $\geq$ [21.75;28.15]	0.38
BMI <[28.45;32.35] $\wedge$ preg <[5.5;6.5]	−0.26
BMI <[38.45;45.45] $\wedge$ pedi <[0.6;0.9]	−0.26
BMI $\geq$ [21.75;28.15] $\wedge$ pedi $\geq$ [0.14;0.37]	0.22
linear: plas	0.15
plas <[115.5;141.5] $\wedge$ preg <[5.5;6.5]	−0.09
BMI <[38.45;45.45] $\wedge$ plas <[142;188.5]	−0.06
BMI $\geq$ [28.45;32.35] $\wedge$ pedi $\geq$ [0.38;0.59]	0.06
preg $\geq$ [5.5;6.5]	0.05
BMI <[38.45;45.45] $\wedge$ plas <[142;188.5] $\wedge$ preg <[7.5;8.5]	−0.05
BMI <[28.45;32.35] $\wedge$ preg <[7.5;8.5]	−0.05
preg <[7.5;8.5] $\wedge$ skin $\geq$ [3.5;11.5]	−0.02

aspect. This trade-off can be seen in Figure 3, where only $CRE_{k:2}$ and $CRE_{k:4}$ are able to consistently achieve good accuracy and sparsity (top right quadrant). This is enabled through the usage of ensemble rules, that allow smooth decision boundaries even for extremely sparse solutions. Ensemble compression also improves the predictive performance of the SIRUS framework, when using the same amount of rules. CRE produces more accurate models, when combined with gradient boosting, while producing more sparse solution, when using random forest to generate the rules. While in this study a well tuned XGBoost model is the most accurate, CRE is on average not much worse while producing very well interpretable models.

5.3 Interpretation

Finally, we showcase how CRE can be used for vivid interpretation. Here we focus on the literal interpretation of the rules, as they are the main advantage of rule ensembles. The following table shows an output of $CRE_{k:4}$ for the Diabetes dataset [1]:

It becomes immediatly obvious that diabetes is strongly connected to age and BMI and its interaction. BMI appears to be the most important covariate, appearing in almost all of the ensemble rules, often in combination with different covariates. The rule $BMI \geq [21.75; 28.15]$ also demonstrates the usefulness of ensemble rules. In this region the risk of diabetes starts to increase, but no single split value would describe the relationship well and would be rather arbitrary. The distribution of split points and $\Phi(x)$ can be visualised, as shown in Figure 2.

CRE can also give an explanation of how a prediction is produced. Table 4 shows the output for a 'close call' observation with $(age, BMI, pedi, preg, plas) = (32, 23.3, 0.67, 8, 62.1)$. It is interesting to take a closer look at the rule $age \geq [27.5, 34.5]$. If this ensemble rule

[1] A description of the covariates can be found in the supplementary material. Y = 1: diabetes positive.

Table 4: CRE prediction explanation.

Rule	β	$\mathcal{R}(x)$	$\beta\mathcal{R}(x)$
linear: plas	0.023	62.1	1.440
Intercept	−1.410	1	−1.410
age $\geq$ [27.5;34.5]	0.470	0.450	0.210
BMI <[38.45;45.45]	−0.260	0.410	−0.110
$\wedge$ pedi <[0.6;0.9]			
preg $\geq$ [5.5;6.5]	0.050	1	0.050
BMI <[38.45;45.45]	−0.060	0.200	−0.010
$\wedge$ plas <[142;188.5]			
BMI $\geq$ [21.75;28.15]	0.380	0.020	0.010
age $\geq$ [21.5;26.5]	0.570	0.020	0.010
$\wedge$ BMI $\geq$ [21.75;28.15]			

fully fires ($\mathcal{R}(x) = 1$, cf. equation (4)) the risk of diabetes increases by $\exp(0.47) = 1.6$. In this example about half the split points in the ensemble rule fire, leading to an increase in risk of $\exp(0.21) = 1.24$. If instead hard rules were used, the rule could only give the full risk increase or none at all. While the last rule contributes little to the current prediction, it is still interesting: If the covariate BMI increases this rule will fire more strongly and the diabetes risk will increase. Instead of giving all or nothing decisions, CRE allows to spot grey areas, that are interesting for interventions.

6 Conclusion and Future Directions

We proposed a framework to compress decision tree ensembles into smooth decision rules. Combining ensemble conditions with the RuleFit approach leads to simpler and more robust models, while being competitive in terms of predictive performance. We argue, that the increase in complexity, due to smooth decision rules, does not harm interpretability. On the contrary, it resembles human intuition, so that the interpretation reflects the models uncertainty better.

We expect CRE to be more stable than RuleFit, as the ensemble rules are less dependent on the specific data sample and are more consistent between runs. However, in this paper we were unable to test the stability empirically, as to the best of our knowledge no suitable stability measure exists. The approach in (Bénard et al., 2021) requires discretizing the data, which does not make sense for CRE. Suitable stability measures would be highly desirable, for future work.

Compressed rules may also be interesting to approximate a forest by means of a simpler model. To this end, rule compression can be used to get an insight in the inner workings of a forest, by extracting the most common paths in the forest, as was showcased by the combination with the SIRUS approach.

Supplement A – Data Gathering and Preprocessing

Datasets, used in the section 5 and the diabetes dataset in section 4 of the paper are taken from the UCI machine learning repository (Dua and Graff, 2017). These criteria are:

- In this article we only consider binary classification.
- We chose datasets with mostly numerical features or features with low cardinality.
- Only datasets with low number of missing values are considered to minimize algorithm differences in missing value handling.

This criteria are set in order to make preprocessing as minimal as possible. Mostly numerical features are chosen for two reasons: (1) Ensemble compression only works on numerical features. (2) Tested algorithms have different ways to deal with discrete features, therefore we want to limit the influence of the implementation on the results. The preprocessing takes the following steps:

- Missing values are mean-imputed.
- Categorical features are simply transformed to numerical features, using the factor levels. (only in the Australian dataset).
- Dummy covariates are left as they are.
- For the liver dataset the Covariate "drinks number" is used to generate the classes, as in (**?**).

Generally, first the datasets were selected and the preprocessing fixed, then we ran the experiments and no further datasets were excluded.

Supplement B – Algorithm Settings

The exact settings and software used to allow reproducibility of results in section 4 are stated below:

- RuleFit: we use the R-package **pre** (Fokkema, 2020) to build the RuleFit model. For reasons of comparability we use boosted CART trees to generate the rules, but note that using the conditional random forest method to generate the trees might improve performance, as shown in Fokkema (2020). Other settings are set to the ones in Friedman and Popescu (2008). The most impactful parameter, *treedepth* is determined via internal 5-fold CV trying the values $1, 2, 3, 4, 5$. λ is taken as minimal value from the sequence, promoting accurate models.
- PRE is built using the default setting of **pre**, which was shown in (Fokkema, 2020) to provide a good trade-off between accuracy and interpretability.
- RandomForest (RF) are built using the R-package **randomForest** (Breiman, 2001). The number of features sampled at each split is left to default ($\lfloor\sqrt{p}\rfloor$) and normal bootstrapping used for resampling. RF is used as a out-of-the-box baseline.

- XGBoost (XGB) is tuned via Bayesian Optimization, as it relies much more on suitable parameters, which is done with the R-package **mlrMBO** (Bischl et al., 2017). The learning rate is considered between $[0.005, 0.1]$, covariates per tree between $[0.7, 1]$, subsample per tree between $[0.2, 1]$ and the *maxdepth* of trees as $\{1, 2, 3, 4, 5, 6\}$. The budget is set to 20 and the remaining values to default.

Supplement C – Additional Results

The following graphs shows a graphical visualisation of the results presented in section 4 of the paper:

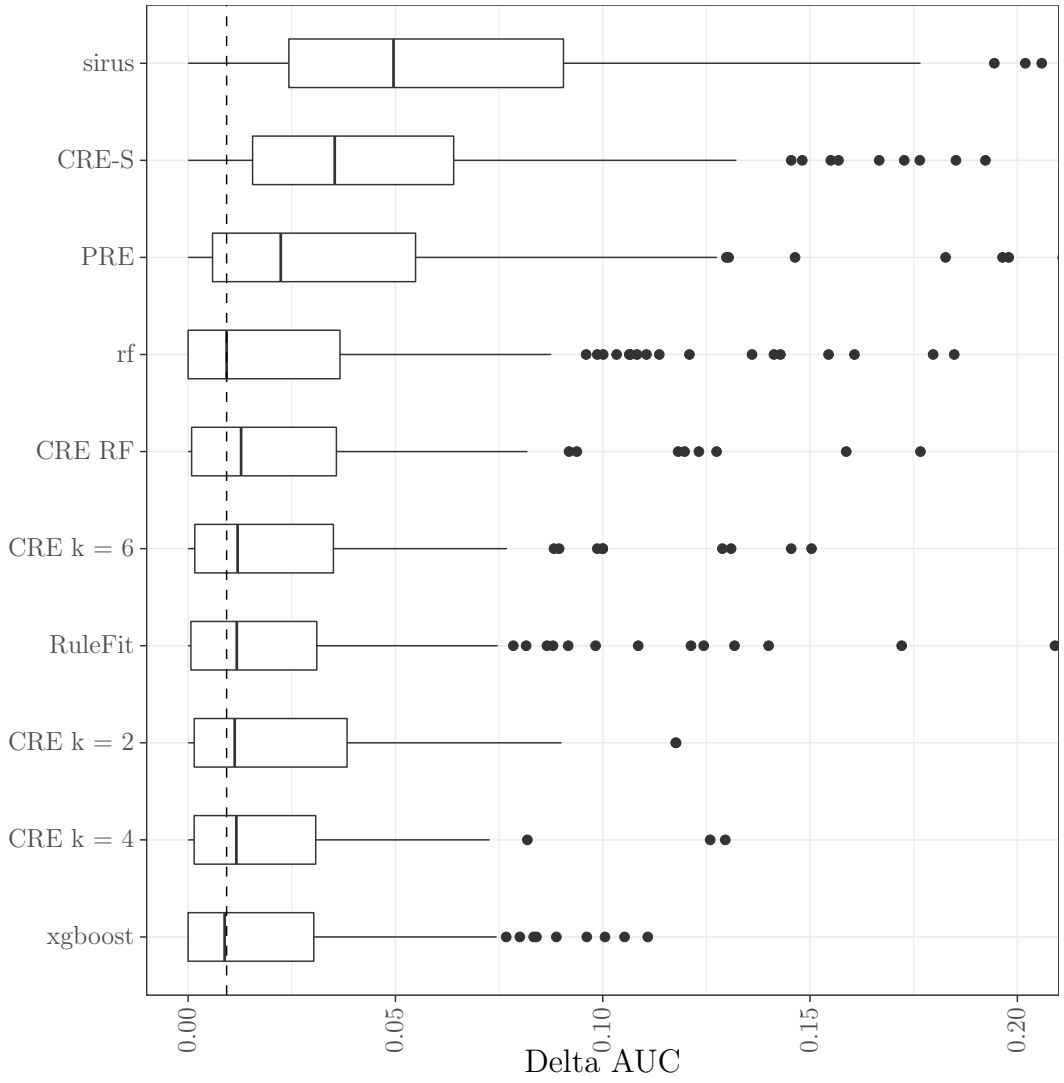


Figure 3: ΔAUC for the competing methods. The best performing method will have $\Delta AUC = 0$ in the given fold. The methods are ordered by the mean ΔAUC.

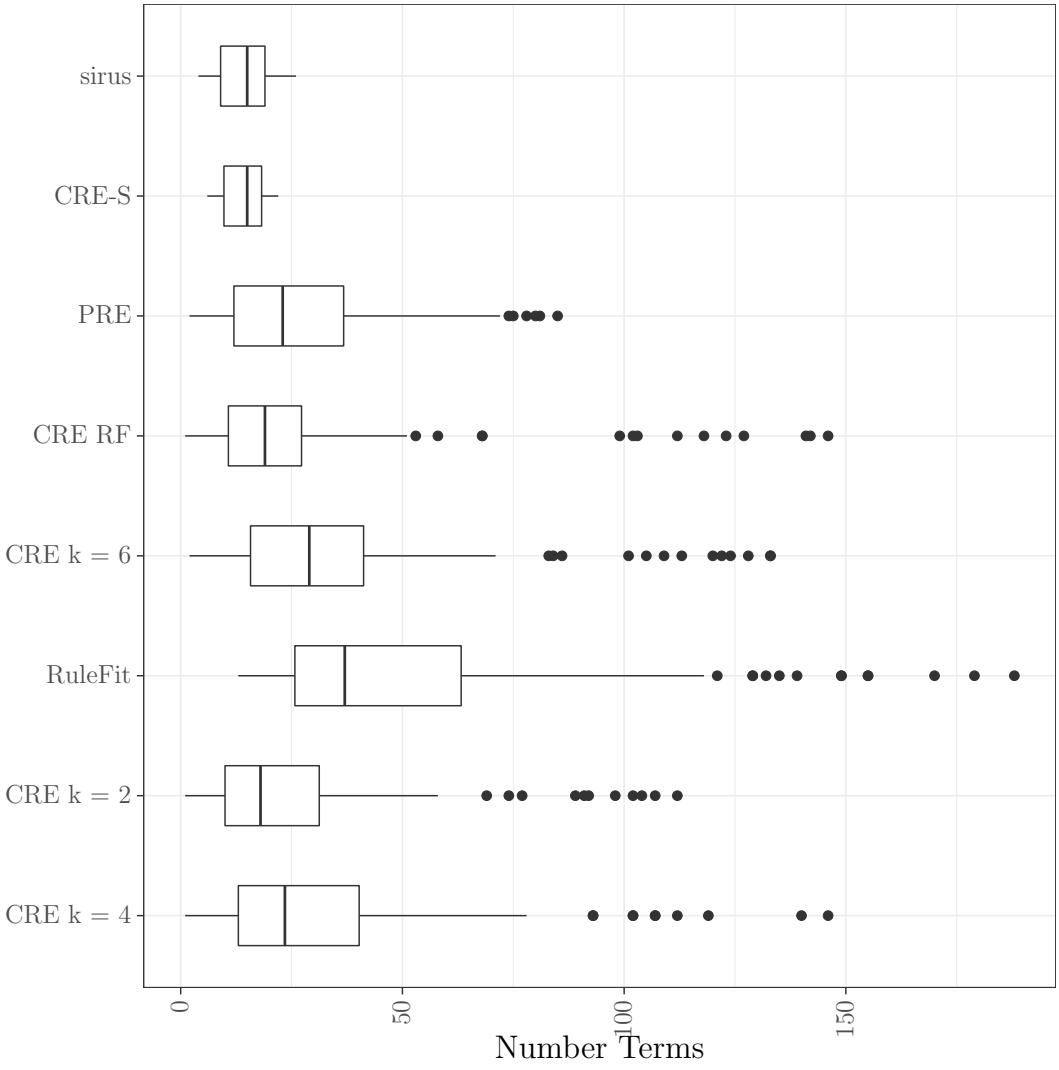


Figure 4: Number of non-zero coefficients (rules or linear terms).

Table 5: Performance meassured by accuracy over the 16 datasets.

dataset	CRE-S	$CRE_{k:2}$	$CRE_{k:4}$	$CRE_{k:6}$	CRE_{RF}	PRE	RF	RuleFit	SIRUS	XGB
Australian	0.841	0.864	0.862	0.868	0.864	0.861	0.865	0.870	0.829	0.868
Banknote	0.905	0.999	0.999	0.999	0.999	0.989	0.993	0.996	0.899	0.998
Biodeg	0.663	0.855	0.860	0.866	0.862	0.855	0.868	0.873	0.767	0.863
Blood Transf	0.762	0.765	0.761	0.762	0.767	0.762	0.751	0.781	0.762	0.789
Diabetes	0.651	0.776	0.758	0.752	0.762	0.752	0.768	0.769	0.698	0.758
Haberman	0.735	0.735	0.732	0.735	0.735	0.735	0.725	0.712	0.735	0.732
Heart	0.786	0.822	0.802	0.815	0.819	0.785	0.809	0.829	0.822	0.838
ILPD	0.714	0.715	0.714	0.705	0.722	0.714	0.705	0.696	0.714	0.700
Ionosphere	0.840	0.920	0.932	0.935	0.937	0.937	0.934	0.920	0.883	0.926
Liver	0.569	0.609	0.615	0.612	0.621	0.583	0.539	0.580	0.565	0.600
Parkinsons	0.809	0.897	0.912	0.907	0.907	0.856	0.902	0.902	0.866	0.922
Pop Failure	0.915	0.952	0.948	0.952	0.944	0.944	0.922	0.948	0.915	0.946
Sonar	0.732	0.857	0.842	0.838	0.856	0.785	0.842	0.847	0.756	0.842
Spambase	0.860	0.946	0.952	0.952	0.947	0.942	0.953	0.946	0.857	0.957
WBCD	0.939	0.967	0.967	0.961	0.967	0.963	0.960	0.965	0.942	0.970
Wilt	0.946	0.983	0.982	0.986	0.984	0.984	0.982	0.986	0.946	0.986
Mean Rank	8.562	4.031	5.281	4.250	3.562	6.469	6.312	4.625	8.062	3.844
Delta Best	0.071	0.009	0.011	0.010	0.007	0.022	0.018	0.011	0.053	0.007

Although we believe accuracy to be less informative compared to AUC, we also provide tabular results of the accuracy. The results are quite similar to the AUC results. Noteworthy difference is $CRE-S$ which performs worse when meassured in accuracy, implying that the prediction outputs are not well calibrated. Another noteworthy difference is, that using the accuracy as meassure, CRE_{RF} shows the overall strongest performance.

Supplement D – Dataset Description of the Diabetes Data

To show the easy interpretability of CRE, we use in Section 4 the freely available Pima Diabetes data set. For a more detailed description, see (**?**). The full names of covariates are:

- `preg`: Number of times pregnant
- `plas`: Plasma glucose concentration a 2 hours in an oral glucose tolerance test
- `pres`: Diastolic blood pressure (mm Hg)
- `skin`: Triceps skin fold thickness (mm)
- `insu`: 2-Hour serum insulin (mu U/ml)
- `mass`: Body mass index (weight in kg/(height in m)^2) [2]
- `pedi`: Diabetes pedigree function
- `age`: Age (years)
- `y`: Class variable (0 or 1) (1 = Diabetes)

[2] Also referred to as BMI in the main paper, due to better understandability.

References

Aho, T., Ženko, B., Džzeroski, S., Elomaa, T., and Brodley, C. (2012). Multi-target regression with rule ensembles. *Journal of Machine Learning Research*, 13.

Akdemir, D., Heslot, N., and Jannink, J.-L. (2013). Soft rule ensembles for supervised learning. *stat*, 1050:22.

Bénard, C., Biau, G., Da Veiga, S., and Scornet, E. (2021). Sirus: Stable and interpretable rule set for classification. *Electronic Journal of Statistics*, 15:427–505.

Bischl, B., Richter, J., Bossek, J., Horn, D., Thomas, J., and Lang, M. (2017). *mlrMBO: A Modular Framework for Model-Based Optimization of Expensive Black-Box Functions*.

Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.

Bühlmann, P. (2012). Bagging, boosting and ensemble methods. In *Handbook of Computational Statistics*, pages 985–1022. Springer.

Bühlmann, P. and Yu, B. (2002). Analyzing bagging. *The Annals of Statistics*, 30:927–961.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4:266–298.

Cohen, W. W. (1995). Fast effective rule induction. In *Machine learning proceedings 1995*, pages 115–123. Elsevier.

Dembczyński, K., Kotłowski, W., and Słowiński, R. (2008). Maximum likelihood rule ensembles. In *Proceedings of the 25th International Conference on Machine Learning*, pages 224–231.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Fernandez-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research*, 15:3133–3181.

Fokkema, M. (2020). Fitting prediction rule ensembles with R package pre. *Journal of Statistical Software*, 92:1–30.

Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, volume 96, pages 148–156.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33:1–22.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38:367–378.

Friedman, J. H. and Popescu, B. (2008). Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2:916–954.

Fürnkranz, J. (1999). Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13:3–54.

Jawanpuria, P., Jagarlapudi, S. N., and Ramakrishnan, G. (2011). Efficient rule ensemble learning using hierarchical kernels. In *Proceedings of the 28th International Conference on Machine Learning*, pages 161–168.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30:4765–4774.

Meinshausen, N. (2010). Node harvest. *The Annals of Applied Statistics*, 4:2049–2072.

Nalenz, M. and Villani, M. (2018). Tree ensembles with rule structured horseshoe regularization. *The Annals of Applied Statistics*, 12:2379–2408.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Schubert, E., Sander, J., Ester, M., Kriegel, H.-P., and Xu, X. (2017). DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)*, 42:1–21.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58:267–288.

Wang, H. and Song, M. (2011). Ckmeans.1d.dp: optimal k-means clustering in one dimension by dynamic programming. *The R journal*, 3:29.

Wei, D., Dash, S., Gao, T., and Gunluk, O. (2019). Generalized linear rule models. In *International Conference on Machine Learning*, pages 6687–6696.

Weiss, S. M. and Indurkhya, N. (2000). Lightweight rule induction. In *Proceedings of the 17th International Conference on Machine Learning*, pages 1135–1142.

Malte Nalenz and Thomas Augustin

Cultivated Random Forests: Robust Decision Tree Learning through Tree Structured Ensembles

Technical Report Number 240, 2021
Department of Statistics
University of Munich

http://www.statistik.uni-muenchen.de

Cultivated Random Forests: Robust Decision Tree Learning through Tree Structured Ensembles

Technical Report

Malte Nalenz
Department of Statistics
Ludwig-Maximilians-University
Munich

Thomas Augustin
Department of Statistics
Ludwig-Maximilians-University
Munich

Abstract

We propose a robust decision tree induction method that mitigates the problems of instability and poor generalization on unseen data. In the spirit of model imprecision and robust statistics, we generalize decision trees by replacing internal nodes with two types of ensemble modules that pool together a set of decisions into a soft decision: (1) option modules consisting of all reasonable variable choices at each step of the induction process, (2) robust split modules including all elements of a neighbourhood of an optimal split-point as reasonable alternative split-points. We call the resulting set of trees *cultivated random forest* as it corresponds to an ensemble of trees which is centered around a single tree structure, alleviating the loss of interpretability of traditional ensemble methods. The explicit modelling of non-probabilistic uncertainty about the tree structure also provides an estimate of the reliability of predictions, allowing to abstain from predictions when the uncertainty is too high. On a variety of benchmark datasets, we show that our method is often competitive with random forests, while being structurally substantially simpler and easier to interpret.

1 Introduction

Decision trees are one of the most common prediction methods. Their popularity mostly stems from their interpretability and methodological simplicity. Decision tree successively partition the covariate space into smaller subspaces that are purer with respect to the target values y. Most practical algorithms, such as CART [6] and C45 [27], use a greedy procedure that chooses the covariate and split-point with the largest gain in purity at each step. Decision trees are adaptive to arbitrary underlying functions and can perform well in several domains. A major downside of decision trees is their instability with respect to small perturbations of the training data, see already [4]. Slight changes in the training set can lead to entirely different tree structures, raising suspicion about the validity of their implied interpretations as well as their generalizibility to unseen data.
The instability can be traced back to the all-in decision at each node [8]. Adding or removing observations might lead to the choice of a different splitting point or even different variable to split on. Through the recursive structure, all decisions depend on the

previous ones. Thus small changes in the top layer of the tree can lead to dramatically different subtrees. Decisions can only partially be reversed post-hoc through pruning away dubious subtrees, making individual choices very influential.
Ensemble methods such as bagging [4], random forests [5] and boosting [16] solve the instability and generalizibility issues at the cost of giving up the interpretational simplicity: Instead of a single tree model, a sequence of trees is generated, each built on alterations of the original data. The final prediction is then the a combination of these individually weak decision trees.
In this article, we take a conceptually different approach. Instead of trying to find an optimal single tree or generating an ensemble of multiple decision trees, we model the uncertainty about the tree structure directly. To this end we introduce ensemble modules that pool a set of decisions into a soft decision. Ensemble modules capture the uncertainty about both the variable to use and the choice of an exact splitting position. The resulting model, that we call *cultivated random forest* (CRF), corresponds to an ensemble of trees, carrying over the desired stability and generalizability of ensemble methods. However, through a notion of neighbourhood interpretability is preserved. In many domains such as the clinical, the ability to inspect what a prediction is based upon is crucial in order to reveal spurious or nonsensical relationships [9], potential gender and racial biases [10] and give practitioners the option to intervene with the decision system in a guided way. This is also important to build acceptance from practioners. Additionally, through the framework of model imprecision, CRF is able to give an estimate of the reliability of predictions, that can be used to abstain from a predictions, if the uncertainty is too high. This is especially important, when the decision system is integrated in a larger work-flow and also alternative means of decision exist such as domain experts. With this, CRF offers a good trade-off between high accuracy, interpretability and accountability.
In section 2 basic notations and principles of decision tree learning are recalled. In section 3 we introduce ensemble modules and the resulting CRF model. Benchmark results on several binary classification datasets are shown in section 4, and section 5 concludes.

2 Decision tree learning

Decision Trees. Decision trees use a graph of decision rules to map a p-dimensional covariate vector $\mathbf{x}_i = (x_{i,1}, ..., x_{i,p})$ to a decision about the target value y_i. In the following all covariates are assumed to be numeric and the target to be binary thus $y \in \{0, 1\}$. Trees consist of a root node, a set of internal nodes and a set of leaf nodes. Starting from the root node where the whole dataset $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ consisting of N covariate vectors is present, subsets of observations are moved to its childnodes based on decision rules, recursively partitioning $\mathcal{X}$ into smaller rectangles. Here only univariate binary decisions of the form $d(\mathbf{x}, t_0, j) = I(x_j \leq t_0)$ are considered, thus each decision leads to exactly two childnodes. For ease of notation $d(\mathbf{x}, t_0, j)$ will be in the following oversimplified as $d(\mathbf{x})$ and assumed that t_0 and j are attached.
In this article, we also utilize the idea of fractional observerations [27, 29]: if a decision can not be made with certainty, observations are split up into fractions and moved to both childnodes. Each observation is attached a value $w_{i,l} \in [0, 1]$ that represents the fraction of the i'th observations that is present in the l'th node. Once a leafnode is reached, a decision is made based on its attached prediction value $\hat{y}$ for the target variable y, typically either the majority class or class distribution. A leaf node can be written as a

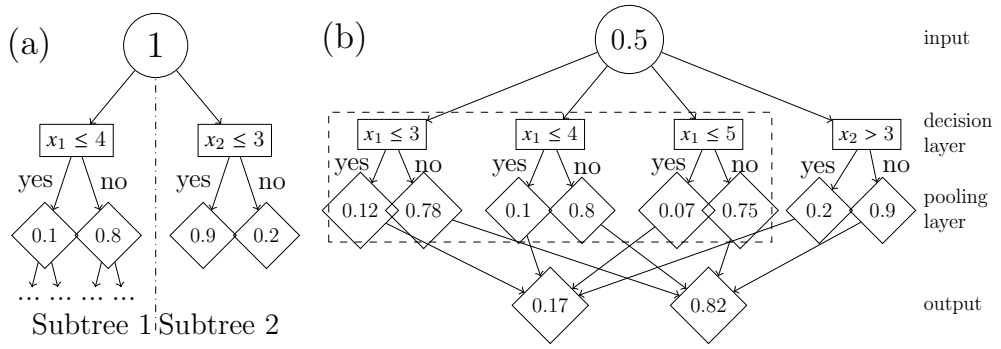


Figure 1: Option Tree vs. Ensemble module. Decisions are shown as rectangles, implied current leaf nodes as diamonds. The numbers inside the diamonds reflect the target distribution $p(y = 1|\mathbf{x})$. (a) Option Tree. Different subtrees follow from each option split (b) Ensemble module, consisting of a set of decisions followed by pooling of the resulting leafnodes with similar target distribution. The decision nodes inside the dashed rectangle can be summarized as a single robust split module.

product of decision rules by $\mathcal{L}(\mathbf{x}, \mathcal{D}_\mathcal{L}) = \prod_{d \in \mathcal{D}_\mathcal{L}} d(\mathbf{x})$, where $\mathcal{D}_\mathcal{L} = \{d_1, ..., d_l\}$ is the path of l decisions, that have to be fulfilled [1] to reach this node [17].

Tree induction. Given a training sample where we are given $\mathcal{X}$ and the labels y, the goal is to build a decision tree that is able to classify unseen samples. Many different tree inducers have been proposed, for an overview we refer to [30]. Usually, at each step during training the decision that minimizes some measure of impurity in the implied childnodes is chosen and applied to the data points, partitioning the training data. In this article, the Gini impurity is used as in CART [6]. This recursive process is repeated until no split reduces impurity further or a stopping criterion is reached.

Decision tree instability. Instability in decision tree learning is a well known problem [20, 4]. At each node a single decision is required, while there can be considerable uncertainty about the correct choice. This dilemma leads to a high degree of instability. In this article, we focus on:

Variable uncertainty: At each node, a binary decision tree needs to decide on exactly one covariate for further partitioning. If the implied purity of several covariates is similar, this all-in approach neglects the uncertainty about our choice.

Splitting point uncertainty: Given the covariate to split on, a cut-point t_0 needs to be chosen. If the impurity surface is flat, a sharp decision is not justified. This uncertainty translates to a lack of smoothness that is found in decision trees [20].

Other sources of uncertainty include paramter uncertainty, such as the correct maximum tree depth, and the choice of the best subtree. This is typically addressed through pruning techniques. Good overviews can be found in [23, 30].

Ensemble methods and option trees. Random forests address the aforementioned stability issues with the bagging of randomized trees. In the standard version each tree is build independently on a bootstrap sample of $\mathcal{X}$ using only a subset of covariates. The predictions of this sequence of trees are combined through averaging or voting. Ensemble

[1] If the path to this leaf node goes to the left side at a node j we take $d_j(\mathbf{x})$ into $\mathcal{D}_\mathcal{L}$ while we take $1 - d_j(\mathbf{x})$ if it goes to the right. More details can be found in the supplementary materials.

methods work well in practice because they reduce several problems of single decision trees: they introduce smoother decision boundaries, mitigate the variable selection uncertainty and lead to better generalization performance.
Another approach to address the variable uncertainty is to use multiple decisions at each node if the implied purity is similar. This was first introduced by Buntime [8] as option trees. In Fig.1 (a), an option tree is shown. At each node several decisions are allowed. The resulting subtrees are subsequently grown and evaluated separately and the final prediction is an aggregate over all subtrees. The option tree approach was combined in [15] with boosting into the alternating decision tree (ADT) model. ADTs were shown to posses decent predictive performance and relatively small model sizes. An interesting property is that option trees and ADT can be seen as structurally sparse representation of an ensemble. As a part of the structure is shared by all subtrees, a whole ensemble of trees can be described by a single tree structure [14].

3 Cultivated Random Forest

In this work, ensemble learning is viewed from the point of robust statistics and model imprecision. Instead of a single model, we are looking at a set of models that correspond to a set of different choices in the model construction process. Typically, in robust statistical models, these are distributional assumptions or priors in the Bayesian setting. In the context of decision trees, model imprecision was applied to the probability distributions in the leafnodes to robustify entropy based splits [22, 2]. Here we instead use this framework to express our uncertainty about the structure of the tree itself and capture the uncertainty involved with the choices made during the tree induction process. To this end, we generalize the decision tree model by replacing internal nodes with *ensemble modules* $\mathcal{M} = \{d_1, ..., d_h\}$ that consist of the set of h decisions that are reasonable at a given step of the induction process. To preserve the binary tree structure, the decisions are then pooled and observations split up into fractions. Usually the left child node is the True part of the decision rule. For ensemble modules we require the decisions to be directed as in [32]. In binary classification we define the right childnode to have the higher implied target probability $p(y = 1|x)$. For multinomial and other target distributions, more sophisticated merging algorithms are required, as in [26][31]. For the case that all decisions are weighted equally and we use the average as pooling function, the fraction going to the left childnode is given by $\psi(\mathbf{x}, \mathcal{M}) = |\mathcal{M}|^{-1} \sum_{d \in \mathcal{M}} d(\mathbf{x})$. If only a fraction of an observation is present in this leaf, we simply take fractions of this fraction.
The whole process is shown in Fig. 1 (b). A set of decisions is considered inside the ensemble module and then pooled into two child nodes. Note that the decision $x_2 > 3$ is directed, such that the right childnode has the higher target probability $p(y = 1|x)$. The name ensemble module stems from the insight that the fraction of an observation to be present in a given leaf node can be written by replacing the l binary decisions in the path to a leafnode with l ensemble modules $\mathcal{M}_{\mathcal{L}} = \{\mathcal{M}_1, ..., \mathcal{M}_l\}$, and pooling after each ensemble module, as

$$\mathcal{L}(\mathbf{x}, \mathcal{M}_{\mathcal{L}}) = \prod_{\mathcal{M} \in \mathcal{M}_{\mathcal{L}}} \left(\frac{1}{|\mathcal{M}|} \sum_{d \in \mathcal{M}} d(\mathbf{x}) \right) = \frac{1}{|D_\times|} \sum_{D_{\mathcal{L}} \in D_\times} \left(\prod_{d \in D_{\mathcal{L}}} d(\mathbf{x}) \right) = \frac{1}{|D_\times|} \sum_{D_{\mathcal{L}} \in D_\times} \mathcal{L}(\mathbf{x}, D_{\mathcal{L}}) \tag{1}$$

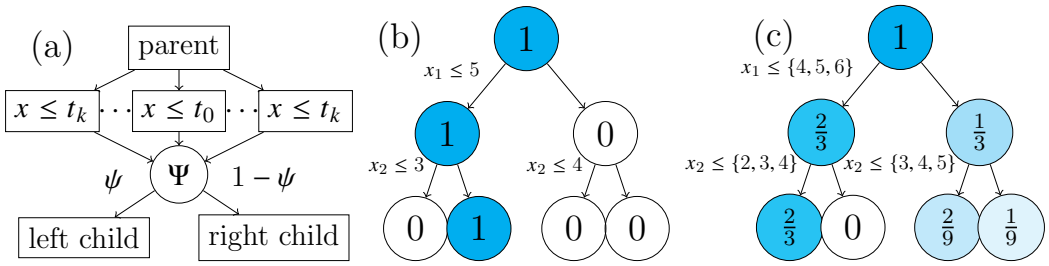


Figure 2: (a): Compact representation of a robust split module. (b,c) Example Classification of $(x_1, x_2) = (5, 4)$ assuming $\phi = 1$. (b) standard binary tree, (c) tree with robust split module. Nodes are drawn as circles. Numbers and colouring represent $w_{i,l}$, the fractions of the example observation that reach each node.

with $\mathcal{D}_\times = \{\mathcal{M}_1 \times ... \times \mathcal{M}_l\}$. So in fact, by pooling the decisions at each ensemble module, the fractional observations in each leafnode can be written as average over an ensemble of trees that is spanned by the Cartesian product of the ensemble modules. The derivation can be found in the supplementary materials. By that CRF is a structurally sparse representation of an ensemble of size $|\mathcal{D}_\times|$. The main difference to random forests is that the trees in the ensemble are not grown independently, but instead are chosen as all reasonable choices along the induction process. Individual trees are therefore not constructed as weak learners that are decorrelated in order to improve the final classifiers generalizibility. On the contrary, the trees share the largest part of the tree structure with other trees in the ensemble but deviate on average just in a few decisions. We expect that generalization performance will be lower in certain domains where the decorrelation aspect of random forest is important to capture all underlying mechanisms present in the data. But we argue that in many applications this aspect is overcompensated by the simplification and resulting interpretability. We introduce two types of ensemble modules, that will be defined formally in the next sections:

- *Option modules* that consist of all reasonable variable choices at the each respective step.
- *Robust split modules* that given the best split-point with respect to impurity for each covariate also consist of reasonable alternative thresholds.

Both module types of ensemble modules can be combined. This is shown in Fig.1 (b) by replacing the single decision that is part of the option module $x_1 \leq 4$ with a robust split module.

Robust split module. [7] show that an ensemble of decision trees using bagging without replacement can be described as a neighborhood around the optimal split-point. Imagine the simple example where we are given one numeric covariate x with associated labels y. The underlying true function is $y = I(x > t_0)$, however x is observed with noise. Depending on the degree of noise and the sample size, when using bootstrap samples of the original data, the decisions will be distributed around the true split point t_0. In this simple example the ensemble can be summarized as $P(t_o)$, where P is a unknown distribution function. Therefore, it should be possible to describe large parts of a bagged tree ensemble through the neighbourhood of splits.
The splits found in our induction process are unlikely to be optimal, so the theoretical results from [7], just discussed, can be only understood as a heuristic. As no prior

knowledge about the split's distribution is available we choose a non-parametric approach and use the closest points in the covariate space to construct a neighbourhood $\mathcal{T}(t_0)$ around t_0. Let $(x_{(i)}, w_{(i)})$ be the i'th *ordered* covariate value and its fraction present in the current node to split, then we define the robust split module as

$$\mathcal{T}(t_0) = \{t_{-j} = x_{(i-j)} \leq \ldots \leq t_{-1} = x_{(i-1)} \leq t_0 = x_{(i)} \leq t_1 = x_{(i+1)} \leq \ldots \leq t_m = x_{(i+m)}\}$$

with

$$j = (\arg\max_{\tilde{j}} \sum_{q=i-\tilde{j}}^{i-1} w_{(q)} < k) + 1 \qquad ; \qquad m = (\arg\max_{\tilde{j}} \sum_{q=i+1}^{i+\tilde{j}} w_{(q)} < k) + 1.$$

Intuitively, on both sides we take k "full" observations into the set. This expresses our assumption that we can not be too sure about the exact position of the split and should also consider all slightly different splits as equally likely, mimicking the behaviour of bagging. As split-points are constituted by observations, fractional observations directly imply fractional split-points. Let $\phi(t)$ denote the weight that can be interpreted as the "representation strength" defined by the point mass of the data points in the current leaf for a cut-point t, given by the recursive function

$$\phi(t) = \begin{cases} \sum_{i=1}^{N} w_{(i)} I(x_{(i)} = t), & \text{if } t \notin \{t_{-j}, t_m\} \\ k - \sum_{l=1}^{j-1} \phi_(t_{-l}), & \text{if } t = t_{-j} \\ k - \sum_{l=1}^{m-1} \phi_(t_l), & \text{if } t = t_m. \end{cases} \qquad (2)$$

In (2) an exception is made for the boarder cases t_{-j} and t_m. As those often can not be included fully they are simply asssigned the remaining of k on this side. This neighbourhood is constructed such that $\sum_{t \in \mathcal{T}(t_0)} \phi(t) \leq 2k + \phi(t_0)$ in each robust split module. For an observation reaching a robust split module l, the fraction that is moved to the left side is given by the gating function

$$\psi(\mathbf{x}, \mathcal{T}(t_0)) = \frac{1}{\sum_{t \in \mathcal{T}(t_0)} \phi(t)} \sum_{t \in \mathcal{T}(t_0)} \phi(t) I(x \leq t).$$

The fraction present in the left childnode is then $w_{(i),left} = w_{(i)} \psi(x_{(i)}, \mathcal{T}(t_0))$ and the fraction in the right childnode $w_{(i),right} = w_{(i)}(1 - \psi(x_{(i)}, \mathcal{T}(t_0)))$. This directly implies that for each observation the sum of the fraction over all (current) leaf nodes equals 1 at each moment in training and prediction. Cases close to the decision boundary will be present in both childnodes for further training. This is shown in Fig.2 for an example data point. Instead of being present in only one node, the data point is present in three current leafnodes. This reflects our uncertainty as the observation is close to the decision boundaries and slightly different model choices would have let to different decision.
During induction, the exact position of t_0 will therefore not influence the tree structure substantially, leading to more stable structures, as we withdraw from making a definite decision at this point. Importantly, as $\mathcal{T}$ is centered around t_0, the interpretation of the robust split module is similar to a common binary decision. When looking at a node instead of the sharp interpretation 'if $x_j \leq t_0$' we can interpret each robust split module as 'if x_j is less than around t_0' ($x_j \lessapprox t_0$). This offers a nice trade-off between smoothness and interpretability. The parameter k controls the degree of smoothness that

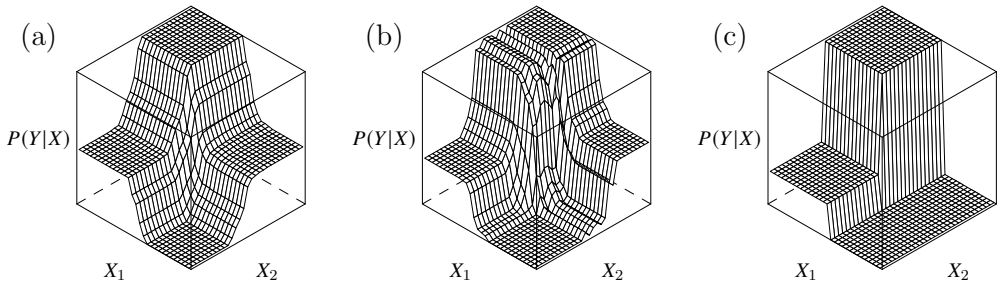


Figure 3: Decision Surface for the simulated dataset of (a) The first split using CRF, (b) full random forest model (c) CART tree with depth 2. Details can be found in the supplementary materials.

we enforce on the model. In the extreme case of $k = N$ the gating function ψ equals the univariate empirical cumulative distribution function. Our choice is motivated by the results in [7], who show that split-points using bagging without replacement lie within a $N^{1/3}$ neighborhood around the optimal t_0. Due to the suboptimality of found splits in practice, we found a slightly bigger neighbourhood of $k = (\sum_{i=1}^{N} w_{(i)})^{1/2}$ to work well. Note that k is not optimized in the tree induction.

Option module. To address the variable uncertainty, we introduce option modules, similar to option trees and ADT. Let h_j denote the weighted impurity using covariate j for splitting and h_{min} the minimal impurity value found in the current step. Then, for a given threshold η_l, for the next split all covariates x_j with $\{j|h_j \leq h_{min} + \eta_l\}$ are taken into the set as reasonable options. As the decisions made in the top layers of the tree are the most influential on the tree structure, the parameter η_l is set to diminish with increasing depth, here by $\eta_l = \eta_0/s_l{}^3$, where s_l is the tree depth in node l and η_0 is a pre-specified parameter. Thus in the top layers more covariates are taken into option modules, while in the bottom layers extra covariates are only added if the decision is very tight.[2] Option modules can easily be combined with robust split module using robust split modules instead of single splits for each covariate. h_j can then be set as average impurity of all elements in $\mathcal{T}(t_0)$. This approach shares similarity with the idea of inner ensembles, where bootstrap samples are used to decide on the best next *single* split [1, 21]. $\phi(\mathcal{T}(t_0))$ is then normalized to sum up to one, to give each covariate the same weight. The combination of the two types of modules is shown in Fig.3 on simulated data, where the two classes are drawn from 2-dimensional mixture of normals. In this example, CRF approximates the behaviour of random forest quite well with respect to the decision surface and the smoothness that is introduced. The smoothness stems from the robust split modules. Also both covariates are used in the first split, as the decision is tight. In contrast to multivariate split methods, such as oblique trees [25], the decisions are not jointly optimized. This can be seen as a

[2]We also tested using Hoeffding bounds to select alternative covariates [12, 24].The Hoeffding bound is inversely proportional to n and in our experiments too few covariates were considered in the top levels of the tree and too many in the bottom levels. However a more theoretically motivated choice of η that expresses the trade-off between reflecting all choices and the reduced interepretability would still be desirable.

form of regularization, as the decisions are required to be individually predictive. The decision surface for CART in Fig.3 (c) shows the lack of smoothness in standard decision trees, making it impossible to capture the underlying relationship in this simulation.

Predictions. For a leafnode the attached prediction value is set as the mean of $\mathbf{y}$ weighted by its fractions present in this node $\hat{y}_j = (\sum_{i=1}^{N} w_{i,l})^{-1} \sum_{i=1}^{N} w_{i,l} y_i$. At prediction time, test cases will have non-zero weights in several leafnodes. The output from our model for a given observation i and the set of m leafnodes is the set $\check{\mathcal{Y}} = \{(\hat{y}_1, w_{i,1}), ..., (\hat{y}_m, w_{i,m})\}$. For obtaining a real valued point estimate, we can simply use the weighted average over the set $\check{\mathcal{Y}}$ with $p(y|x_i) = \sum_{j=1}^{m} w_{i,j} \hat{y}_j$. Note that the size of $\check{\mathcal{Y}}$ is the number of leafnodes, not the number of trees that are represented by our ensemble. For interpretation, this allows to look at the leaf nodes with the highest fractions and have a symbolic description, what the prediction is based on. It can also be informative to look at the spread of $\check{\mathcal{Y}}$ as a measure for reliability of this prediction. A natural measure is the variance of the fractional predictions. The reliability reflects the degree of conflict between the decisions in the ensemble modules. If an observation is often close to the decision boundary, or if decisions inside option modules are contradictory, the prediction is found unreliable. Consider the example where half the fractions of an observations fall into leafs with prediction value 0 and the other half into leafs with prediction value 1. Both the variance and the final prediction for this observation will be 0.5. This prediction shows a high degree of uncertainty in two layers: about the predicted value $p(y|x) = 0.5$ and given $p(y|x)$ the stochastic uncertainty about the outcome. On the other hand, if an observation always falls into leafs with predicted values of 0.5, we are quite certain that we should predict 0.5 and the uncertainty concerns only the outcome. A nice property for set-valued predictions is the option to abstain from a prediction if the uncertainty is too high [11]. This is important in practice, when a high cost is associated with a wrong prediction. For example in clinical applications it might be better to remeasure covariates in case of potential measurement error or consider a further test altogether if the model prediction is unreliable for a given patient. Also expert knowledge should be taken into account for uncertain predictions, if available.

4 Experiments

To test the predictive capabilities of our proposed method, we carried out 10-fold cross validation on two sets of data sets:

Gene expression data. The goal is to predict a binary disease outcome, based on the genetic expression profile. These data sets are characterised by an extreme $p >> n$ situation with thousands of covariates and small sample sizes.

Binary classification benchmark datasets. The datasets are taken from the UCI repository [13]. Data are coming from various domains, including small and medium sized data sets with varying number of covariates, and varying degrees of target imbalance.

Evaluation settings. All datasets together with seeds (that were generated randomly for the experiments) and a data description are available in the supplementary material to enhance an easy reproducibility. As some of the datasets are imbalanced, we chose the area under the ROC curve (AUC) as evaluation metric. We compare CRF against random forest as the ensemble benchmark and to CART as the baseline for an interpretable model. All methods were run with standard settings from the R-libraries randomForest and rpart respectively [28]. We test 4 different versions of CRF:

	Dataset	random forest	CRF-full	CRF-split	CRF-option	CRF-shallow	CART
	Colon [3]	0.850	**0.887**	**0.874**	0.808	**0.887**	0.859
Gene	DLBC [18]	0.951	**0.955**	0.901	0.78	**0.955**	0.721
Expression	Leukaemia [19]	**1**	**1**	**1**	**1**	**1**	0.838
	Prostate [18]	0.955	**0.968**	0.938	0.847	**0.968**	0.851
	Australian Credit	0.932	**0.936**	0.921	**0.928**	**0.937**	0.903
Benchmark	Banknote	1.000	**0.998**	**0.998**	**0.999**	**0.995**	0.966
Data	Blood Transfusion	0.685	**0.750**	**0.731**	**0.691**	**0.745**	0.729
	Climate Model	0.930	**0.940**	**0.920**	0.909	**0.936**	0.771
	Diabetes	0.828	**0.831**	0.814	0.800	**0.830**	0.797
	EEG-Eye-State	0.985	0.935	0.940	0.934	0.794	0.724
	Haberman	0.682	**0.724**	**0.696**	**0.681**	**0.723**	0.626
	Indian Liver	0.752	**0.749**	0.724	0.709	**0.749**	0.667
	Ionosphere	0.982	0.957	0.940	0.960	0.949	0.905
	Magic	0.937	0.920	0.901	0.904	0.887	0.808
	Parkinsons	0.980	0.950	0.949	0.932	0.953	0.890
	QSAR Biodeg	0.933	**0.923**	0.900	0.878	0.906	0.838
	Spambase	0.986	**0.980**	0.971	0.973	0.962	0.894
	SPECTF	0.850	0.837	0.764	0.724	**0.841**	0.721
	Steel Plates	0.992	**1.000**	**1.000**	**1.000**	**0.987**	1.000
	Vertebral	0.995	0.958	0.956	0.966	0.947	0.927
	Wisconsin Breast	0.992	**0.992**	**0.987**	**0.991**	**0.992**	0.948
	Wilt	0.987	**0.987**	**0.989**	**0.984**	0.958	0.960

Table 1: Average AUC using 10-fold CV over 22 Datasets. Results are marked in bold where versions of CRF performs comparably or better than random forest.

1. *CRF-full* with both types of ensemble modules with a max-depth of 14.
2. *CRF-split* using only neighborhood moduleswith a max-depth of 14.
3. *CRF-option* using only option modules with a max-depth of 14.
4. *CRF-shallow* with a max-depth of 6 leading to a maximum of 126 nodes/ensemble modules and both types of ensemble modules.

Each version uses a minimum node size for splitting of 6, a data depenendet parameter $k = \sqrt{n_l}$, where n_l is the sum of weights in node l and $\gamma_0 = 0.3$. All algorithms could be tuned, so we believe it to be a fair comparison to run them with standard settings, especially as random forests are known to be quite robust with respect to the parameter choices. More details about the different implementations can be found in the supplementary materials.

Predictive performance. Table 1 shows the AUC of the competing methods using the above setting. All versions of CRF outperform CART on all tested datasets. On 16 out of the 22 tested Datasets CRF-full performs comparably or better than Random Forest, if one is willing to trade off 0.01 in AUC. In 9 Datasets CRF-full shows slightly better performance. Performance is especially good in small data sets, where the uncertainty in the tree induction process is high. Noteworthy is the good performance on the gene expression datasets. CRF is able to account for the extremely high uncertainty due to the small samples and has s generalization performance on par with random forest, whereas CART clearly struggles. Also on difficult and perhaps noisy data sets such as Blood Transfusion and Haberman with overall low AUC values, CRF shows strong performance. Here the decorrelated trees in random forests might become too weak, leading to suboptimal ensemble performance. On 5 of the datasets CRF performs significantly

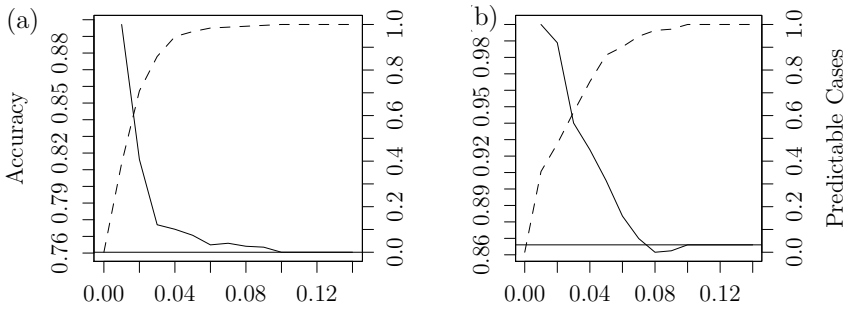


Figure 4: Accuracy, given the option to abstain from a vote, with varying thresholds the prediction spread $\sigma(p(y|\mathbf{x}))$. The dashed line shows the proportion that can be reliably predicted and the full line the accuracy for the predicted cases. (a) Diabetes, (b) Parkinsons.

worse than random forest. A likely explanation is that for these datasets the optimal decision surface is truly multimodal, which CRF in its current state is unable to capture well.

Influence of treedepth. CRF-shallow performs almost identical to CRF-full on most datasets. However on some datasets performance drops significantly, implying that deeper trees are necessary in some of the datasets. This result is also interesting, as it suggests that over-fitting is not a huge problem in CRF. Note that no form of pruning is applied on the deeper trees.

Robust split modules. Using only the robust split module deteriorates the AUC on most datasets, but still outperforms CART on all tested datasets. On 9 datasets the performance is similar to random forest. Note that CRF-split offers almost the same easy interpretability as a standard decision tree.

Option modules. CRF-option still improves on standard CART in most datasets, but the lack of smoothness provided by the robust split modules decreases performance significantly. It is also notable that on some datasets the decrease is quite big, especially the genetic datasets with small sample sizes. The uncertainty about the split-point is here the highest and neglecting it results in worse generalization performance due to instability. Also the setting of η and choice of η_0 might be suboptimal on these datasets.

Abstaining from predictions. Fig.4 shows the accuracy if the classifier is given the option to abstain if the prediction spread $\sigma(p(y|\mathbf{x})) > t$ for different thresholds t. On the Diatebes and Parkinsons data, abstaining from about 40% of the predictions gives an substantial increase of 8% in accuracy. For the abstained predictions other means of decisions (such as expert opinions) might be better suited, or data recollected in case of possible meassurement error. Together with the interpretability aspect, this makes our method especially well suited for medical and clinical applications.

To summarise, in this section we showed, that CRF can perform remarkably well on many datasets despite restricting the ensemble to a neighborhood around a single tree.

5 Conclusion

We introduced a new framework for the construction of tree ensembles: instead of growing a sequence of trees, we construct a set of trees corresponding to reasonable choices in the construction process. The resulting cultivated random forests (CRF) are structurally simpler than regular random forests, which can be beneficial both for diagnosing their validity as well as memory efficiency. The empirical results suggest that CRF is competitive to random forests on many of the tested data sets. An further advantage is that CRF gives an estimate of the reliability of each prediction that can be used to abstain from predictions. This might also be interesting in further research, when building ensembles of randomized CRF, where weak learners can abstain from predictions. We believe that the framework of model imprecision in decision tree learning is well worth exploring further, as it is flexible and the CRF can be generated "on the fly". Next steps include more data adaptive ways to construct neighbourhoods, as well as exploring weighting schemes when pooling covariates in option modules. Also suitable pruning methods should be investigated, as it is likely to limit the complexity further and might lead to an increase in accuracy. A shortcoming of our method, that we will adress in the future, is the implicit assumption of uni-modal impurity surfaces in the covariates. While being beneficial for interpretation this might harm predictitve performance for those data sets, where this assumption does not hold.

6 Broader Impact

We believe that our framework of tree structured ensemble learning makes a step towards much needed transparency in machine learning. As machine learning emerges in more and more areas of daily life it affects large parts of society directly. Black-box models may be used to predict insurance claims, calculate credibility scores for credit applicants or in prosecution of potential criminals, with potentially negative consequences for individuals. In our opinion the possibility to give a reasoning behind a prediction should be a minimal requirement in these areas. The same is true in clinical applications, where statistical models might decide on the optimal treatment and consequences of error may be fatal. Here practitioners should need to have the possibility to have insight in the reasoning behind a prediction, in order to challenge its validity. Also the ability to estimate its own reliability becomes more and more important in order to build trust in the predictions made by machine learning models. Here our approach is only a first step and more sophisticated ways should be explored to estimate a model's reliability.

References

[1] Houman Abbasian, Chris Drummond, Nathalie Japkowicz, and Stan Matwin. Inner ensembles: Using ensemble methods inside the learning algorithm. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 33–48. Springer, 2013.

[2] Joaquin Abellan and Serafin Moral. Maximum of entropy for credal sets. *International journal of uncertainty, fuzziness and knowledge-based systems*, 11(05):587–597, 2003.

[3] Uri Alon, Naama Barkai, Daniel A Notterman, Kurt Gish, Suzanne Ybarra, Daniel Mack, and Arnold J Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the national academy of sciences*, 96(12):6745–6750, 1999.

[4] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[5] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[6] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[7] Peter Bühlmann and Bin Yu. Analyzing bagging. *The annals of statistics*, 30(4):927–961, 2002.

[8] Wray Buntine. Learning classification trees. *Statistics and computing*, 2(2):63–73, 1992.

[9] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1721–1730, 2015.

[10] Danton S Char, Nigam H Shah, and David Magnus. Implementing machine learning in health care - addressing ethical challenges. *The new england journal of medicine*, 378(11):981, 2018.

[11] Giorgio Corani, Joacquin Abellán, Andres Masegosa, Serafin Moral, and Marco Zaffalon. Classification. In Thomas Augustin, Frank PA Coolen, Gert De Cooman, and Matthias CM Troffaes, editors, *Introduction to imprecise probabilities*, pages 916–954. Wiley, 2014.

[12] Pedro Domingos and Geoff Hulten. Mining high-speed data streams. In *Proceedings of the sixth ACM SIGKDD international conference on knowledge discovery and data mining*, pages 71–80, 2000.

[13] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[14] Eibe Frank, Michael Mayo, and Stefan Kramer. Alternating model trees. In *Proceedings of the 30th annual ACM symposium on applied computing*, pages 871–878, 2015.

[15] Yoav Freund and Llew Mason. The alternating decision tree learning algorithm. In *International conference of machine learning*, volume 99, pages 124–133, 1999.

[16] Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

[17] Jerome H Friedman and Bogdan E Popescu. Predictive learning via rule ensembles. *The annals of applied statistics*, 2(3):916–954, 2008.

[18] Enrico Glaab, Jaume Bacardit, Jonathan M Garibaldi, and Natalio Krasnogor. Using rule-based machine learning for candidate disease gene prioritization and sample classification of cancer gene expression data. *PLOS one*, 7(7), 2012.

[19] Todd R Golub, Donna K Slonim, Pablo Tamayo, Christine Huard, Michelle Gaasenbeek, Jill P Mesirov, Hilary Coller, Mignon L Loh, James R Downing, Mark A Caligiuri, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.

[20] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009.

[21] Igor Ibarguren, Jesús M Pérez, Javier Muguerza, Ibai Gurrutxaga, and Olatz Arbelaitz. Coverage-based resampling: Building robust consolidated decision trees. *Knowledge-based systems*, 79:51–67, 2015.

[22] Carlos J Mantas and Joaquín Abellán. Credal-C4.5: Decision tree based on imprecise probabilities to classify noisy data. *Expert systems with applications*, 41(10):4625–4637, 2014.

[23] John Mingers. An empirical comparison of pruning methods for decision tree induction. *Machine learning*, 4(2):227–243, 1989.

[24] Zahra Mirzamomen and Mohammad Reza Kangavari. A framework to induce more stable decision trees for pattern classification. *Pattern analysis and applications*, 20(4):991–1004, 2017.

[25] Sreerama K Murthy, Simon Kasif, and Steven Salzberg. A system for induction of oblique decision trees. *Journal of artificial intelligence research*, 2:1–32, 1994.

[26] John C Platt, Nello Cristianini, and John Shawe-Taylor. Large margin dags for multiclass classification. In *Advances in neural information processing systems*, pages 547–553, 2000.

[27] J Ross Quinlan. *C4. 5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[28] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019.

[29] Brian D. Ripley. *Pattern recognition and neural networks*. Cambridge University Press, 1996.

[30] Lior Rokach and Oded Maimon. Top-down induction of decision trees classifiers-a survey. *IEEE Transactions on systems, man, and cybernetics, part C (applications and reviews)*, 35(4):476–487, 2005.

[31] Jamie Shotton, Toby Sharp, Pushmeet Kohli, Sebastian Nowozin, John Winn, and Antonio Criminisi. Decision jungles: Compact and rich models for classification. In *Advances in neural information processing systems*, pages 234–242, 2013.

[32] Albrecht Zimmermann. Ensemble-trees: Leveraging ensemble power inside decision trees. In *International conference on discovery science*, pages 76–87. Springer, 2008.

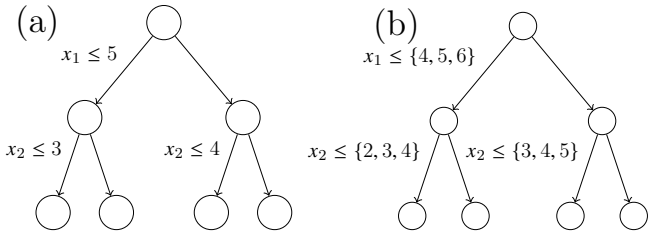


Figure 5: Example for the paths to leafnodes

Supplement A - Details of Cultivated Random Forest

A leaf node can be written as a product of decision rules by $\mathcal{L}(\mathbf{x}, \mathcal{D}_\mathcal{L}) = \prod_{d \in \mathcal{D}_\mathcal{L}} d(\mathbf{x})$, where $\mathcal{D}_\mathcal{L} = \{d_1, ..., d_l\}$ is the path of l decisions, that have to be fulfilled. For the 4 leafnodes shown in Fig. 1 (a) the corresponding paths $\mathcal{D}_\mathcal{L}$ are:

1. $\{d(\mathbf{x}, 5, 1), d(\mathbf{x}, 3, 2)\}$
2. $\{d(\mathbf{x}, 5, 1), 1 - d(\mathbf{x}, 3, 2)\}$. Note $1 - d(\mathbf{x}, 3, 2)$ corresponds to the decision $I(x_2 > 3)$.
3. $\{1 - d(\mathbf{x}, 5, 1), d(\mathbf{x}, 4, 2)\}$
4. $\{1 - d(\mathbf{x}, 5, 1), 1 - d(\mathbf{x}, 4, 2)\}$

If instead ensemble modules are used as shown in Fig. 1 (b) the corresponding paths $\mathcal{M}_\mathcal{L}$ are:

1. $\{\psi(x, \mathcal{M}_1), \psi(\mathbf{x}, \mathcal{M}_2)\}$
2. $\{\psi(x, \mathcal{M}_1), 1 - \psi(\mathbf{x}, \mathcal{M}_2)\}$
3. $\{1 - \psi(x, \mathcal{M}_1), \psi(\mathbf{x}, \mathcal{M}_3)\}$
4. $\{1 - \psi(x, \mathcal{M}_1), 1 - \psi(\mathbf{x}, \mathcal{M}_3)\}$

with $\mathcal{M}_1 = \{d(\mathbf{x}, 4, 1), d(\mathbf{x}, 5, 1), d(\mathbf{x}, 6, 1)\}$, $\mathcal{M}_2 = \{d(\mathbf{x}, 2, 2), d(\mathbf{x}, 3, 2), d(\mathbf{x}, 4, 2)\}$ and $\mathcal{M}_3 = \{d(\mathbf{x}, 3, 2), d(\mathbf{x}, 4, 2), d(\mathbf{x}, 5, 2)\}$. Then we can write the fraction present in a given

leafnode as:

$$\begin{aligned}
\mathcal{L}(\mathbf{x}, \mathcal{M}_{\mathcal{L}}) &= \prod_{\mathcal{M} \in \mathcal{M}_{\mathcal{L}}} \left(\frac{1}{|\mathcal{M}|} \sum_{d \in \mathcal{M}} d(\mathbf{x}) \right) \\
&= \frac{1}{\prod_{\mathcal{M} \in \mathcal{D}_{\mathcal{L}}} |\mathcal{M}|} \left(\sum_{d_1 \in \mathcal{M}_1} d_1(\mathbf{x}) \cdot \sum_{d_2 \in \mathcal{M}_2} d_2(\mathbf{x}) \cdot \ldots \cdot \sum_{d_l \in \mathcal{M}_l} d_l(\mathbf{x}) \right) \\
&= \frac{1}{|D_{\times}|} \sum_{d_1 \in \mathcal{M}_1, \ldots, d_l \in \mathcal{M}_l} d_1(\mathbf{x}) \cdot \ldots \cdot d_l(\mathbf{x}) \\
&= \frac{1}{|D_{\times}|} \sum_{D \in D_{\times}} \left(\prod_{d \in D} d(\mathbf{x}) \right) \\
&= \frac{1}{|D_{\times}|} \sum_{D_{\mathcal{L}} \in D_{\times}} \mathcal{L}(\mathbf{x}, D_{\mathcal{L}})
\end{aligned}$$

with $\mathcal{D}_{\times} = \{\mathcal{M}_1 \times \ldots \times \mathcal{M}_l\}$.

Supplement B - Simulation

The data used for the illustrative example in Figure 3 of the main paper was generated as following: for $i = 1, \cdots, 300$:

$y_i \sim \mathcal{B}(0.5)$

$x_i | y_i = 1 \sim \mathcal{N}_2\left(\begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 0.75^2 & 0 \\ 0 & 0.75^2 \end{pmatrix} \right)$

$x_i | y_i = 0 \sim \mathcal{N}_2\left(\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0.75^2 & 0 \\ 0 & 0.75^2 \end{pmatrix} \right)$

The simulation is used to highlight the smoothness of the decision boundaries of different classifiers, when the true underlying relationship requires smooth boundaries.

Supplement C - Benchmark Experiments

Data

The UCI data sets were selected under following criteria:

- Public availability.
- Binary classification.
- Mostly numeric or categorical features with low cardinality.

The last criteria has the following reasoning. One of the main improvements of CRF compared to CART is the robust split module, that requires numerical attributes. Hence categorical attributes are transformed to dummy variables, while random forests and CART have different ways built in to handle categorical attributes with larger cardinality. However the CRF framework can straightforwardly be extended to more naturally handling

of categorical features (for example using the approach from CART), which will be one of the next steps in further research. So to achieve a good comparison of the main improvements of CRF we focus on datasets with mostly numeric covariates.
For the genetic datasets fully preprocessed datasets were chosen, so that no further preprocessing needed to be applied.

Dataset	Covariates	N	Class 1	Class 2
Australian	14	690	383	307
Banknote	4	1372	762	610
Blood Transfusion	4	748	570	178
Climate Model	20	540	46	494
Diabetes	8	768	500	268
EEG-Eye-State	14	14980	8257	6732
Haberman	3	306	225	81
Indian Liver	10	583	416	167
Ionosphere	34	351	126	225
Magic	10	19020	12332	6688
Parkinsons	22	195	48	147
QSAR Biodeg	42	1055	699	356
Spambase	57	4601	2788	1813
SPECTF	44	267	55	212
Steel Plates	33	1941	1268	673
Vertebral	6	620	420	200
Wilt	5	4839	4578	261
Wisconsin Breast Cancer	30	569	357	212

Table 2: Characteristics of the 18 binary classification benchmark data sets from the UCI.

Dataset	Covariates	N	Class 1	Class 2
Colon	2000	62	40	22
DLBC	2647	77	19	58
Leukaemia	7128	72	47	25
Prostate	2135	102	50	52

Table 3: Characteristics of the 4 gene expression data sets.

Table 2 shows characteristics of the UCI datasets and Table 3 for the genetic datasets. The datasets cover a wide range of domains as well as varying size and number of covariates. Also different situations of class imbalance are covered. The folder of data sets used is attached. The target variable can be found in the last column of each data set.

Implementations

We used R (v. 3.5.3) to perform the benchmarking. To this end 10-fold crossvalidation was performed, using the same seed for data splitting for each method.

Random Forest Random forest was fit using the ***randomForest*** (v. 4.6-14) package from CRAN. We ran the algorithm with standard settings, which corresponds to 500 trees that are grown until purity and $p/3$ covariates, where p is the number of covariates in total, tried at each node.

CART For fitting the CART algorithm, we used the ***rpart*** (v. 4.1-15) package from CRAN. The standard settings are a minimum number of observations for splitting of 20 and a maximum depth of 30. To prevent overfitting a pre-pruning mechanism is build in: a split is not made if the impurity meassure is reduced by less than 0.01. We also tried cost-complexity pruning implemented in rpart, however the results got worse on average, compared to prepruning.

Cultivated Random Forest We implemented a prototype version in R. All code to run the experiments together with a full implementation of CRF will be published online at the time of publication, as described above.

Expert RuleFit: Complementing Rule Ensembles with Expert Knowledge

Luisa Ebner[1], Malte Nalenz[2][0000−0003−3439−4469], Annette ten Teije[2][0000−0002−9771−8822], Frank van Harmelen[1][0000−0002−7913−0048], and Thomas Augustin[2][0000−0002−1854−6226]

[1] Vrije Universiteit Amsterdam, de Boelelaan 1081a, Amsterdam, NL
[2] University of Munich, Ludwigstr. 33, Munich, Germany
ebner.luisa@gmx.de
{Frank.van.Harmelen,Annette.ten.Teije}@vu.nl
{Malte.Nalenz,Thomas.Augustin}@stat.uni-muenchen.de

Abstract. Machine learning algorithms have great potential to enhance clinical diagnosis and treatment. Yet, their overall performance is limited by the quality and quantity of available training data, while their adoption is limited by the level of trust ascribed by human experts. Injecting additional knowledge obtained from existing literature or from human expertise into the machine learning algorithm is widely seen as a solution to both of these problems. Yet, few implementations of expert-guided machine learning exist to date. We present Expert RuleFit (ERF), an approach to integrate expert knowledge in the form of rules and linear terms into an existing method for rule learning (RuleFit). A customized regularization strategy allows us to consider the different strengths of expert knowledge. For an empirical evaluation, we trained ERF models on a diabetes dataset for which we acquired expert rules from medical guidelines and expert interviews. We show that our ERF method enriches or replaces potentially spurious correlations learned from a patient sample with expert-derived, validated domain knowledge without sacrificing predictive performance. The integration of different knowledge sources makes the ERF model a promising tool for learning accurate, explainable and trustworthy medical decision rules.

Keywords: Decision rules · Rule learning · Explainable Machine Learning.

1 Introduction

Machine Learning (ML) systems offer great potential in medicine to provide healthcare improvements. Their ability to learn from data without explicit human guidance provides an attractive solution to the problems of manual knowledge acquisition encountered in the development of rule-based expert systems [16]. Considering the complexity and dynamics of medical knowledge, inductive learning is essential for successful decision support systems [16]. However, it is

not to be forgotten that rule-based expert systems [2] have two clear advantages over ML systems.

First, expert systems allow for the integration of and reasoning with various sources of expert knowledge, ranging from personal assessments to factual textbook knowledge. ML algorithms, to the contrary, are dependent on training examples as the only source of information. To generalize well to unseen cases, ML requires sufficient data to represent the population as a whole. Besides the number of training examples, this depends on the amount of information present in the recorded attributes, the amount of noise and the presence of hidden confounders. Due to the high cost and effort of information acquisition, privacy concerns and an intrinsic uncertainty of medical data, clinical datasets are often characterized by few examples, many missing values and insufficient task-relevant input attributes. Then, ML models suffer from limited generalizability. Indeed, significant performance decline is often observed when a model trained on data from e.g. one hospital is used to predict patient outcomes from another.

Second, expert systems draw upon expert-derived knowledge (e.g. in form of rules) to perform reasoning. As a result, expert system recommendations come with explanations that resemble human knowledge and reasoning in structure and vocabulary. The state-of-the-art in ML, to the contrary, often trades explainability for predictive accuracy. In safety-critical applications, this may diminish human trust and chances for system adoption. Without the possibility of expert validation, high performance on test sets derived from the same distribution as the training set is often considered as evidence that real knowledge has been captured by a model. This is dangerous because, in practice, ML cannot guarantee reasonable patterns. Lacking any general domain knowledge, it cannot be ruled out that ML algorithms make mistakes that would appear trivial to a human [11].

A solution to both problems of generalizability and explainability is to incorporate prior knowledge. As an additional source of information, it allows ML algorithms to better generalize to unseen cases while allowing human experts to better understand and validate recommendations. We meet this challenge proposing Expert RuleFit, a classification method that combines the strengths of inductive ML and expert rule-based reasoning. Expert RuleFit injects expert knowledge in the form of rules and linear terms into the existing rule ensemble method RuleFit [9]. We use the term expert knowledge to refer to any form of knowledge that experts consider state of the art and that they formulate to the best of their knowledge. As such, expert knowledge is somehow validated, e.g. through expert reasoning and academic studies or practically from experience or usage. The rules allow to stratify a patient population into task-relevant subpopulations, while the linear terms allow to express correlations between patient attributes and the target. Furthermore, by means of a tailored regularization strategy, our approach allows experts to specify *confirmed* knowledge to certainly enter the final prediction model as well as *optional* knowledge to be promoted over data rules through a customized penalization strategy. By adding expert knowledge in the form of rules to a data-generated rule set, we increase

the explainability and trustworthiness of ML results, to meet the high demands on medical decision support systems.

The contribution of this paper is a novel approach to combine rule *learning* with rule based *expertise*, implemented as an extension of the existing rule ensemble RuleFit and illustrated by a use case of diabetes diagnosis. We start in section 2 with a brief discussion of related work. Section 3 describes the existing RuleFit method and discusses its limitations in medical application contexts. Section 4 presents Expert RuleFit as an expert-guided RuleFit extension to obtain the benefits of expert knowledge inclusion. Section 5 compares Expert RuleFit with the conventional RuleFit method on a use case of diabetes diagnosis. Finally, section 6 concludes and points out research paths for future work.

2 Related Work

Knowledge representation in the form of rules has a long tradition in medical AI, in particular in rule-based expert systems [2]. More recently, the integration of symbolic prior knowledge into the process of learning from examples has been considered in hybrid systems [18]. In pursuit of theory-guided data science and scientific consistency in machine learning, research interest is increasingly devoted to combinations of data- and knowledge driven approaches. Under the umbrella term *informed machine learning*, the recent survey paper [17] provides a structured overview on many different ML learning algorithms that can be enriched with prior knowledge. A taxonomy classifies them according to the *source* of knowledge, its *representation* and its *integration* into the ML process. The majority of research work is concerned with the use of symbolic knowledge in neural networks. [20] and [4] use logical formulas to guide the output of deep neural networks as logical constraints in loss functions. In [14], knowledge graphs enhance deep neural networks with rules about relations between instances. In contrast, Expert RuleFit does not add rule-based knowledge to deep learning, but to a rule learning engine. Using the terminology from the informed ML taxonomy [17], we use expert knowledge as knowledge source, rules as knowledge representation and integrate knowledge directly into the learning algorithm. One particularly related approach is Expert-Augmented Machine Learning (EAML) [11], where domain experts use an online platform to assess the relative risk of subpopulations defined by RuleFit rules and the difference between their assessment and the empirical risk is considered as part of a novel regularization strategy for RuleFit. Whereas EAML first derives knowledge from data and then has the same evaluated by experts post hoc, our approach allows to formulate knowledge a priori which is then taken into account by the rule-learning algorithm. This allows human knowledge to be explicitly integrated to co-define model components.

3 Context

RuleFit is a rule ensemble method proposed by Friedman and Popescu [9]. In general, ML ensembles solve prediction problems by combining the predictions of

several classifiers. That is to say, multiple classifiers are trained to solve the same problem and consequently, their individual results are aggregated in the form of a generalized, linear model to form one joint prediction model [1]. The ensemble members – commonly referred to as base learners – are potentially different functions of different subsets of predictor attributes derived from the training data [9]. The rationale of ensembling is performance improvement by variance reduction. Given a labelled dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, RuleFit derives regression and classification ensembles whose accuracy is competitive with state-of-the-art methods such as random forests [13]. The rule learning ability avoids knowledge acquisition, enables massive data inputs and allows for knowledge discovery. The use of rules and linear terms as base learners makes RuleFit models potentially comprehensible for humans [10]. RuleFit has a better accuracy-complexity trade-off than most of the state-of-the art in ML [11]. This makes it a promising candidate to provide decision support in safety-critical domains with high demands on both accuracy and explainability [21]. The RuleFit algorithm proceeds in two stages: Ensemble Generation and Regularisation.

Stage 1: Ensemble Generation RuleFit models are linear combinations of rules and linear terms, whose predictive relevance is to be defined by respective coefficients. Following Friedman's stochastic gradient boosting strategy of rule learning [8], RuleFit generates an overly large set of candidate rules from boosted tree ensembles. As products of attribute-value tests from the root node to every other node in the tree, rules act as binary classifiers $r(\mathbf{x}) \in \{0, 1\}$, where $\mathbf{x}$ is the covariate vector, indicating whether observations match their conditions. To help illustrate the idea of the rule generation process, Figure 1 depicts an exemplary, simple decision tree generated from the Pima Indian Diabetes (PID) dataset available from the UCI ML Repository [5]. The rules listed in Table 1 correspond to the paths to all nodes of the tree. Note that in RuleFit only the conditions are kept as decision rules, not the predictions in the leaf nodes. The rationale however is that decision rule specify subgroups that are predictive with respect to the target attribute. Rule r_3 in Table 1 specifies patients that are at least 29 years and have a BMI of less than 27. From Fig.1. we can see that the risk for diabetes is lower in this group.

Table 1. Rules corresponding to the decision tree in Fig. 1.

Rules	Conditions
r_1	Age $<$ 29
r_2	Age $\geq$ 29
r_3	Age $\geq$ 29 & BMI $<$ 27
r_4	Age $\geq$ 29 & BMI $\geq$ 27

This process is repeated for each tree from the boosted tree ensemble and the extracted decision rules are concatenated to a large set of rules. The resulting rule set is cleaned according to sufficient support, colinearity and duplicates. To

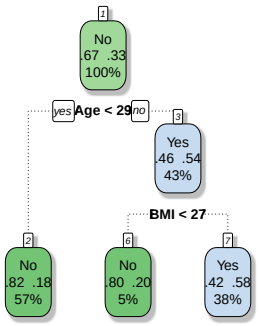


Fig. 1. Decision tree extracted from the PID dataset.

better capture linear effects, numeric attributes $\mathbf{x}_j$ are preprocessed (see [9]) and added as linear terms $l(x_j)$ in the ensemble:

$$F(\mathbf{x}) = \alpha_0 + \sum_{d=1}^{D} \alpha_d r_d(\mathbf{x}) + \sum_{j=1}^{p} \beta_j l(x_j). \tag{1}$$

Stage 2: Regularization To boil the overly large set of candidate rules down to the truly informative ones, Lasso regularized regression is applied to learn the regression coefficients α_d and β_j [19]. The least absolute shrinkage and selection operator "Lasso" is widely considered in ML literature and -practice for sparsification problems. It is easy to implement with a number of efficient solvers available and known for its selective nature when confronted with high dimensional data. Accordingly, the model coefficients $\gamma_{RF} = (\alpha_0, \{\alpha_d\}_1^D, \{\beta_j\}_1^p)$ are learned as:

$$\gamma_{RF} = \arg\min_{\gamma_{RF}} \sum_{i=1}^{N} L\left(y_i, F(\mathbf{x})\right) + \lambda \cdot \left(\sum_{d=1}^{D} |\alpha_d| + \sum_{j=1}^{p} |\beta_j| \right), \tag{2}$$

where L is an appropriately chosen loss function (typically sum of squared for linear regression and negative log-likelihood with sigmoid link-function on $F(\mathbf{x})$ for binary classification) [13]. The result are relatively sparse prediction models, where the majority of coefficient estimates is set to zero [7].

RuleFit's suitability in medical contexts is limited by its dependency on sufficient data and human acceptance. Similar to the majority of ML algorithms, model generalizability is constrained by the quantity and quality of the training set. At the same time, human acceptance of model results is constrained by the number and complexity of learned decision rules as well as their consistency with domain knowledge and expert assessments. Without reference to any general knowledge of the domain, RuleFit rules may often combine conditions

that contradict expert assessments. In this regard, RuleFit offers no possibility to remove some rules and include others. This diminishes RuleFit's chances for regular use and consultation in clinical practice [11].

4 Method

A solution to both the problems of limited generalizability and limited trust lies in an incorporation of expert knowledge into the RuleFit algorithm. Expert knowledge is a natural way to counter the problem of insufficient training data, and in medicine it is widely available. Whereas expert knowledge commonly refers to normal cases, typical symptoms and causal relationships, data-derived patterns reflect real patients with comorbidities, confounding factors and individual differences [11]. Therefore, training data can extend the coverage of expert knowledge through exceptional cases or unknown patterns while expert knowledge can compensate the effects of spurious patterns learned from poor, atypical training examples with medical regularities and consensual knowledge. Furthermore, the inclusion of expert knowledge is likely to increase human trust in model results. After all, physicians formulate patient conditions according to their understanding of the human physiology and task-relevant symptoms and effects, while ML algorithms learn only correlations from the empirical distribution of a patient sample. We therefore present *Expert* RuleFit (ERF) as a classification method, derived in 3 stages: Knowledge Acquisition, Combined Ensemble Generation and Knowledge-Aware Regularization.

Stage 1: Expert Knowledge Acquisition. Prior to the learning process, expert knowledge regarding the learning task may be formulated as rules and linear effects. Similar to the knowledge acquisition strategy used to develop rule-based expert systems, this involves manual knowledge acquisition from domain experts, medical guidelines, study results and textbooks. This information is then translated into rules and linear effects. Rules separate patients into subpopulations with respect to their target values. For example, in diabetes diagnosis the expert rule BMI $>$ 40 & Age $\geq$ 60 & BP $>$ 120 defines a subpopulation of obese, elderly people with increased blood pressure. A physician might specify this subpopulation to have a high incidence rate of diabetes compared to the whole population. Particularly favourable for rule formulation are clinical practice guidelines, whose recommendations on the diagnosis and treatment of patients with specific clinical conditions are either directly formulated as rules or as structured, rule-like statements. To distinguish different degrees to which expert knowledge is validated, ERF allows users to declare some rules and linear terms as *confirmed* and others as *optional* knowledge.

Stage 2: Combined Ensemble Generation. Consequently, at most 4 different sets of expert knowledge enter the ERF model together with the given dataset. These are the sets of confirmed expert rules $r_c, c \in \mathcal{I}_c$ and linear terms

$l_{c_l}, c_l \in \mathcal{I}_{c_l}$ as well as the sets of their optional counterparts $r_o, o \in \mathcal{I}_o$ and $l_{o_l}, o_l \in \mathcal{I}_{o_l}$, where $\mathcal{I}_c$, $\mathcal{I}_{c_l}$, $\mathcal{I}_o$ and $\mathcal{I}_{o_l}$ are disjoint index sets that are also disjoint with $\{1, ..., D\}$. Based on the given dataset and the encoded expert knowledge, data rules r_d are generated using the RuleFit method. This results in one common, enlarged set of base learners to enter the linear predictor

$$F(\mathbf{x}) = \alpha_0 + \sum_{d=1}^{D} \alpha_d r_d(\mathbf{x}) + \sum_{c \in \mathcal{I}_c} \alpha_c r_c(\mathbf{x}) + \sum_{o \in \mathcal{I}_o} \alpha_o r_o(\mathbf{x}) + \sum_{c_l \in I_{cl}} \beta_{c_l} l(x_{c_l}) + \sum_{o_l \in I_{ol}} \beta_{o_l} l(x_{o_l}) \tag{3}$$

of the ERF model. In difference to RuleFit, linear terms are not included by default, but according to expert knowledge on their respective relevance. Since expert knowledge is included before learning the weight coefficients that specify the importance of the base learners, redundant and non-informative expert knowledge can be recognised and assessed as such while expert knowledge incompleteness may be compensated by the rules learned from the training data. ERF models cover the entire spectrum from purely data-driven RuleFit models to models that include only expert knowledge and no data rules.

Stage 3: Knowledge-Aware Regularization. To learn the coefficients, we developed a tailored regularization strategy, where adaptive *penalty factors* serve to guarantee an inclusion of confirmed expert knowledge and allow for a promotion of optional expert knowledge over data-generated predictors in the final model. The term penalty factors refers to multiplicative weight vectors applied to the Lasso penalty term λ, which allow to adjust penalization differently for every coefficient, e.g. to put discount on the inclusion of selected model terms [22]. The optimization problem for estimating the model coefficients $\gamma_{ERF} = (\alpha_0, \{\alpha_d\}_1^D, \{\alpha_c\}_{c \in \mathcal{I}_c}, \{\alpha_o\}_{o \in \mathcal{I}_o}, \{\beta_{c_l}\}_{c_l \in \mathcal{I}_{c_l}}, \{\beta_{o_l}\}_{o_l \in \mathcal{I}_{o_l}})$ extends to

$$\gamma_{ERF} = \arg\min_{\gamma_{ERF}} \sum_{i=1}^{N} L\left(y_i, F(\mathbf{x})\right) + \lambda \left[\sum_{d=1}^{D} |\alpha_d| + \sum_{o \in \mathcal{I}_o} \nu_o |\alpha_o| + \sum_{o_l \in \mathcal{I}_{o_l}} \eta_{o_l} |\beta_{o_l}| \right]. \tag{4}$$

Data rules are fully penalized using **1** as penalty factor. Confirmed expert rules and linear terms are exempted from penalization using **0** as penalty factor: They are certainly included in the final model and therefor do not appear in the penalty term above. For optional expert rules and linear terms, the user may specify customized vectors $\boldsymbol{\nu}$ with $\nu_o \in [0, 1]$ and $\boldsymbol{\eta}$ with $\eta_{o_l} \in [0, 1]$ as penalty factors to prefer each respective base learner to a customized degree over the data rules. The smaller $\boldsymbol{\nu}$ and $\boldsymbol{\eta}$ are chosen, the cheaper it is for the model to include the corresponding covariates. Setting all components of $\boldsymbol{\nu}$ and $\boldsymbol{\eta}$ to 1 leads to an equal treatment of optional terms and data generated terms. This approach loosely resembles the adaptive lasso [22], but with the penalty factors chosen in accordance to medical expertise. Our promotion of expert knowledge

Table 2. An excerpt of expert rules collected from medical guidelines and expert interviews including the degree to which rule accordance is regarded as diabetes indicator.

Expert Rule	Source	Diabetes Prevalence	Type
Age >= 60 & BP >= 81 & BMI > 40	SMC-D	very high	confirmed
Glucose > 110 & BP > 90	NHG-D	mid	optional
Age <= 42 & BP <= 80 & BMI <= 29	Expert 1	low	confirmed
Age >= 45 & BP >= 90 & BMI >= 35 & Glucose >= 130	Expert 2	high	confirmed

through diminished penalties is designed to balance the data rules that precisely fit an empirical distribution and thus help to achieve more robust, generalizable models.

5 Experiments

We evaluate the performance of ERF on the diagnosis of Type 2 diabetes.

Experimental Setting. We use the aforementioned PID dataset, obtained from the UCI repository [5], where the learning task is to diagnose diabetes patients. For 768 adult women, information is recorded regarding the number of pregnancies, age, BMI, triceps skinfold thickness, blood pressure (BP), insulin- and glucose levels, a genetic predisposition to diabetes and the diabetes test result.
As expert knowledge, we manually extracted rules and linear terms from two diabetes guidelines, the *Standards of Medical Care in Diabetes* [3] and the *National healthcare guideline – Diabetes Mellitus Type 2* [15]. Both sources reference attributes in the PID dataset and present knowledge in the form of patient conditions. Guideline information about the extent to which specified conditions indicate the presence of diabetes was used to classify expert knowledge as indicators of *minor*, *moderate*, *strong* and *very strong* diabetes risks. In addition, we conducted two expert interviews with practicing physicians, who specified a set of task-relevant patient subpopulations based on their diagnostic understanding and experience. According to the rationale that indicators of minor or (very) strong diabetes risk are more reliable separators than moderate indicators, we defined 20 confirmed expert rules, 2 confirmed linear terms, 34 optional expert rules and 3 optional linear terms.

Experimental Protocol. We train four different versions of our proposed ERF model, one existing implementation of the conventional RuleFit model and one Random Forest model, whereby the latter two serve as baselines. The four proposed ERF models are as follows: First, a standard ERF model called **ERF** includes data rules together with confirmed and optional expert knowledge with full penalty put on all optional expert knowledge ($\boldsymbol{\nu} = \boldsymbol{\eta} = \mathbf{1}$). Second, the model **ERF prio** is the same, but with optional expert knowledge preferred over data rules using $\boldsymbol{\nu} = \boldsymbol{\eta} = \mathbf{0.5}$. The penalty factors were chosen as equal

because the guidelines and expert assessments did not provide a finer subdivision to justify different penalty values. Third, the model **ERF only** includes only expert knowledge and no data rules. Fourth, the model **RuleFit** – as our implementation of the conventional RuleFit method – contains only data rules but no expert knowledge. In addition, we use the model **PRE** as an existing implementation of the RuleFit method [3] as well as a **Random Forest** model [4] as baselines. We evaluate the models on AUC (area under the ROC curve) and classification accuracy. With regard to explainability, we use the size of the final ensemble as an indicator of model complexity. Furthermore, we consider the proportion of expert knowledge in the final model as an indicator of medically coherent predictors, supporting explainability and trustworthiness of model results. To investigate training data dependence, all models are trained on 4 different sized subsamples of the PID data set. Finally, every individual model was made subject to 10-fold cross validation (CV) to provide balanced accuracy measures [13]. As usual, we derive 10-fold-CV estimates from splitting the original training data into 10 random, equally sized subsets or folds. For each fold k, the model is retrained, using the observations in the other 9 folds and evaluated using the observations in fold k. Eventually, the final performance is calculated as the average performance over the 10 folds [7].

Results. AUC and classification accuracy results (Fig. 2) are similar for all model settings, especially on the full dataset and the sample set of 400 patients. This shows that expert knowledge is task-relevant and often able to replace data rules without sacrificing predictive performance. For the larger data set sizes, the ERF models comprising both data- and expert knowledge are most competitive. For the smaller samples, **ERF** models achieve the same accuracy while including expert-validated patient conditions as predictors. The results of the **ERF only** models, which do not include any predictors learned from the dataset they are evaluated on, suggest that the expert knowledge contains as much task-relevant information as 400 training examples. Looking at the performance of **RuleFit** and **PRE**, we were not able to show a performance gain through the inclusion of expert knowledge. We presume this is partly because our expert knowledge is not complete and partly because the validation set has been randomly subsampled from the same empirical distribution as the training data. Eventually, **ERF** and **ERF prio** are slightly outperformed by the **Random Forest** model on the larger dataset sizes and cleary outperformed on the 200 sample. Our **RuleFit** implementation is rather competitive with **RandomForest** on all dataset sizes and significantly better than **PRE** on the 200 sample.

Final model sizes (Fig. 3) – ranging from 10 to 25 – are similar throughout the competing model settings and dataset sizes, indicating a high interpretability

[3] We use PRE, as the original R-implementation by [9] is no longer available. We adapted our penalization strategy to make results comparable with PRE, by using λ_{1se}, the largest λ within one standard error of the minimal one, to produce a more sparse solution. This was found to produce a better accuracy-complexity tradeoff.

[4] We use the default settings of the R-package randomForest

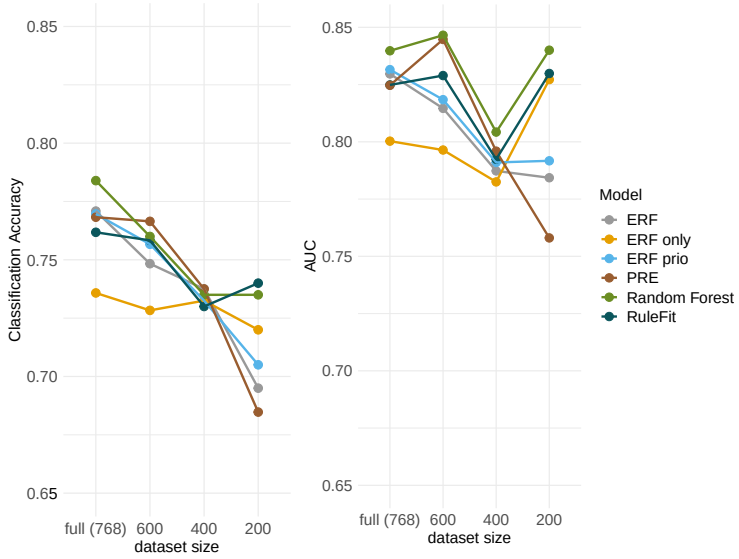


Fig. 2. Cross-validated (10-fold) results on classification accuracy (left) and AUC (right) of ERF- and RuleFit models and a Random Forest model trained on different sized samples of the PID dataset.

of the RuleFit algorithm and its variants, **PRE** and **ERF**, on this dataset. Although we initially entered a total of 59 expert knowledge terms, about half of the confirmed- and about 20 of the optional expert rules were removed due to insufficient support on the dataset or perfect correlation with other expert- or data rules. We see that the inclusion of expert knowledge decreases the ensemble size compared to our implementation of **RuleFit**, but remains slightly above the **PRE** version of RuleFit. Finally, the integration of expert knowledge in the form of additional base learners did not significantly influence the size of the final model.

Results on the proportion of expert knowledge in the final models as well as among their 10 most important base learners (Fig. 4) show high expert knowledge accordance across all ERF models. In particular, 50-75% of all base learners that remain in the final model and 8-10 out of the 10 most important terms (i.e. the terms with highest coefficients) correspond to expert knowledge. Of course, this is partly due to the concept of expert knowledge-aware regularisation, where confirmed expert knowledge is exempted from penalisation. Yet, the results show the value of expert knowledge for adequate predictions. Finally, an exemption from penalisation does not automatically make a model coefficient large and the associated base learner important. Thus, ERF models largely base their results on validated, medically coherent predictors instead of correlations derived from a patient sample.

We conclude that the ERF method yields explainable and more medically coherent models than RuleFit without sacrificing predictive accuracy or adding to model complexity. ERF's potential to yield increased accuracy at decreased model complexity was shown in an associated simulation study in [6]. As such, ERF promises accurate and yet simple models, including a large fraction of

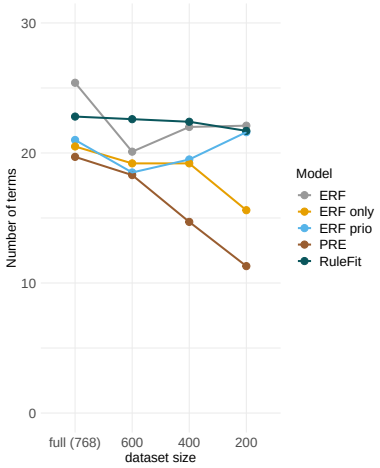


Fig. 3. Cross-validated (10-fold) results on the ensemble size of ERF- and RuleFit models trained on different sized samples of the PID dataset.

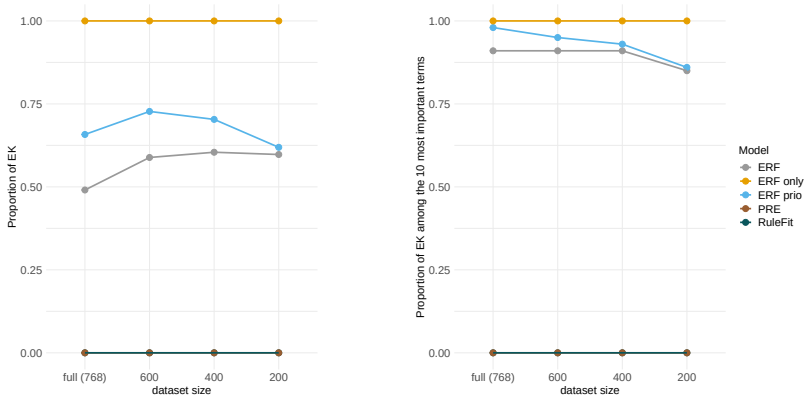


Fig. 4. 10-fold CV results on the proportion of expert knowledge in the final models, among the 10 most important base learners and overall. Importance of a term is defined as the absolute size of its coefficient.

validated, causal knowledge as important predictors and thus lowering the risk of including spurious relationships.

6 Conclusion and Future Work

We presented ERF as an expert-guided ML model for binary classification. Our approach combines the strengths of both ML and rule-based expert systems. While making use of RuleFit's rule learning ability, ERF allows human experts to complement an automatically generated knowledge base with knowledge they themselves work with to make decisions. In addition, ERF allows users to vary between purely expert knowledge-driven and purely data-driven models, depending on which information sources they trust most. This turns machine learning into a tool to enhance human reasoning, instead of overwriting it [12]. Finally, the increased level of human involvement promotes human trust in model results, which in turn raises the chances of adoption in clinical practice.

An inherent limitation is the constraint of expert knowledge to attributes in the dataset. Since ERF learns the weight coefficients of expert knowledge from corresponding data examples, a reference in the data is necessary to empirically evaluate the predictive influence of an expert rule or linear term.

Future work on ERF opens up several research paths. In the first instance, we would like to conduct larger scale experiments with more diverse data sets. Using the PID dataset, all models were evaluated using a validation set that has been randomly subsampled from the empirical distribution. This is risky when the set of patient examples is not representative of the whole population of interest as it is the case with many clinical datasets. However, the test set generally contains similar correlations as the training set [11]. To further evaluate and compare the out-of-sample performance of ERF and RuleFit models, test sets from different health institutions or different survey dates could help to investigate whether the inclusion of general, causal expert knowledge reduces performance decline over time or makes models more robust to changes in underlying variable distributions. If a certain patient group is underrepresented in the dataset used to train the model, expert rules concerning this patient group could help to make the model more generalizable to the entire patient population. To support our assumptions on increased explainability and human acceptance, models should be evaluated and compared by domain experts. On another note, the possibility to evaluate hypotheses or theories on empirical data suggests the use of ERF as an exploratory tool in medical research or even the strongly theory driven social sciences. Finally, the strengths of the ERF method are currently associated with the efforts of manual expert knowledge acquisition and -formulation. Even though ERF facilitates and restricts the latter, good results demand for thought-out development of rules and linear predictors. Especially with regard to scalability, manual knowledge acquisition is suboptimal and may sometimes speak against ERF usage. A promising aspect of future work lies in an integration of ERF with methods for automated knowledge acquisition, such that experts

could point to medical text from which rules and linear effects are extracted automatically.

Acknowledgements

We thank Dr. Waltraud Hoellein and Dr. Peter Hoellein for sharing their diabetes expertise in personal interviews. In addition, we would like to thank the anonymous reviewers for their constructive criticism and detailed comments on a previous version that helped to improve the quality of this paper.

References

1. Błaszczyński, J., Dembczyński, K., Kotłowski, W., Słowiński, R., Szelag, M.: Ensembles of decision rules for solving binary classification problems in the presence of missing values. In: International Conference on Rough Sets and Current Trends in Computing. pp. 224–234. Springer (2006)
2. Buchanan, B.G., Shortliffe, E.H.: Rule Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project. Addison-Wesley (1984)
3. Care, F.: Standards of medical care in diabetes 2019. Diabetes Care **42**(Suppl 1), S124–S138 (2019)
4. Diligenti, M., Roychowdhury, S., Gori, M.: Integrating prior knowledge into deep learning. In: 2017 16th IEEE ICMLA conference. pp. 920–923. IEEE (2017)
5. Dua, D., Graff, C.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
6. Ebner, L.: Expert RuleFit – Complementing Rule Ensembles with Expert Knowledge. Master's thesis, Faculty of Science, Vrije Universiteit Amsterdam (2021), https://www.ub.vu.nl/en/university-library-for-students/vu-thesis-database/index.aspx
7. Fokkema, M.: Fitting prediction rule ensembles with R package pre. Journal of Statistical Software **92**, 1–30 (2020)
8. Friedman, J.H.: Stochastic gradient boosting. Computational Statistics & Data analysis **38**(4), 367–378 (2002)
9. Friedman, J.H., Popescu, B.E.: Predictive learning via rule ensembles. The Annals of Applied Statistics **2**(3), 916–954 (2008)
10. Fürnkranz, J., Gamberger, D., Lavrač, N.: Foundations of rule learning. Springer (2012)
11. Gennatas, E.D., Friedman, J.H., et al.: Expert-augmented machine learning. Proceedings of the National Academy of Sciences **117**(9), 4571–4577 (2020)
12. Giraud-Carrier, C.: Flare: Induction with prior knowledge. Proceedings of Expert Systems 1996 **13**, 173–181 (1996)
13. Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning: data mining, inference, and prediction. Springer (2009)
14. Marino, K., Salakhutdinov, R., Gupta, A.: The more you know: Using knowledge graphs for image classification. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 20–28 (2017)
15. Nationale Versorgungs-Leitlinie: Diabetes Mellitus Typ 2. Nationales Programm für Versorgungs-Leitlinien bei der Bundesärztekammer **7** (2002)

16. Ravuri, M., Kannan, A., Tso, G.J., Amatriain, X.: Learning from the experts: From expert systems to machine-learned diagnosis models. In: Machine Learning for Healthcare Conference. pp. 227–243. PMLR (2018)
17. von Rueden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., Kirsch, B., Pfrommer, J., Pick, A., Ramamurthy, R., et al.: Informed machine learning–a taxonomy and survey of integrating knowledge into learning systems. arXiv:1903.12394 (2019)
18. Sun, R.: Connectionist implementationalism and hybrid systems. Encyclopedia of cognitive science (2006)
19. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological) **58**(1), 267–288 (1996)
20. Xu, J., Zhang, Z., Friedman, T., Liang, Y., Broeck, G.: A semantic loss function for deep learning with symbolic knowledge. In: ICML. pp. 5502–5511. PMLR (2018)
21. Yang, W., Zhang, S., Chen, Y., Chen, Y., Li, W., Lu, H.: Mining diagnostic rules of breast tumor on ultrasound image using cost-sensitive rulefit method. In: 2008 3rd International Conference on Intelligent System and Knowledge Engineering. vol. 1, pp. 354–359. IEEE (2008)
22. Zou, H.: The adaptive Lasso and its oracle properties. Journal of the American Statistical Association **101**(476), 1418–1429 (2006)

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

INSTITUT FÜR STATISTIK

Cornelia Fütterer, Malte Nalenz and Thomas Augustin

Discriminative Power Lasso – Incorporating Discriminative Power of Genes into Regularization-Based Variable Selection

Technical Report Number 239, 2021
Department of Statistics
University of Munich

http://www.statistik.uni-muenchen.de

Discriminative Power Lasso – Incorporating Discriminative Power of Genes into Regularization-Based Variable Selection

Cornelia Fuetterer[+*], Malte Nalenz[+*], and Thomas Augustin[+]

Abstract—In precision medicine, it is known that specific genes are decisive for the development of different cell types. In drug development it is therefore of high relevance to identify biomarkers that allow to distinguish cell-subtypes that are connected to a disease. The main goal is to find a sparse set of genes that can be used for prediction. For standard classification methods the high dimensionality of gene expression data poses a severe challenge. Common approaches address this problem by excluding genes during preprocessing. As an alternative, L1-regularized regression (Lasso) can be used in order to identify the most impactful genes.
We argue to use an adaptive penalization scheme, based on the biological insight that decisive genes are expressed differently among the cell types. The differences in gene expression are measured as their *discriminitive power* (DP), which is based on the univariate compactness within classes and separation between classes. ANOVA based measures, as well as measures coming from clustering theory, are applied to construct the covariate specific DP.
The resulting model, that we call *Discriminative Power Lasso* (DP-Lasso), incorporates the DP as covariate specific penalization into the Lasso. Genes with a higher DP are penalized less heavily and have a higher chance for being part of the final model. With that the model can be guided towards more promising and trustworthy genes, while the coefficients of uninformative genes can be shrunken to zero more reliably.
We test our method on single-cell RNA-sequencing data as well as on simulated data. DP-Lasso leads on average to significantly sparser solutions compared to competing Lasso-based regularization approaches, while being competitive in terms of accuracy.

Keywords—Penalized Regression, Variable Selection, Clustering validation metrics, scRNA-sequencing data.

I. Introduction

In personalized medicine, it is important to identify genes, which can be used to accurately predict the individual outcomes. For the development of biomarkers, a lower number of covariates means less effort in its subsequent clinical testing. As in high-dimensional settings many genes are often noise, the challenge is to select only the covariates that are relevant in terms of prognostic, predictive or biological impact to the drug or the disease [19]. In case of non-small cell lung cancer (NSCLC), the detection of the biomarker EML4-ALK fusion gene [27] led to the development of the drug crizotinib, which is used for patients carrying an ALK-fusion. In contrast to the earlier low response, crizotinib dramatically raised the response rate in NSCLC [19].

[+]Ludwig-Maximilians-University, Munich. Department of Statistics.
[*]These authors contributed equally to this work.

In general, the transition of healthy cells into cancerous cells affects changes in gene expression that can be measured. It is therefore common practice to investigate single-cell RNA sequencing data, introduced by [30], which allows insights into the different cell types of single cells. In the case of a cell cycle, the cell passes from the DNA synthesis (S-phase) to the mitosis (M-phase), including the gap phases (G1 and G2) in between. These different phases can be distinguished by its measured gene expression of a synchronized cell population. For example, a high score at the G2M checkpoint can be an indicator of metastasis tumor [21]. Testing whether genes are differentially expressed among different cell types might therefore lead to valuable insights.

From a biological point of view, it is therefore of relevance to extract a sparse set of genes that can be used to classify and characterize the subpopulations [11]. One common approach is to use penalized regression models, such as the Lasso [31] that find a trade-off between model fit and model complexity. The advantage of the Lasso is that it provides variable selection, by setting coefficients to exactly zero. An extension is the adaptive Lasso [36] which uses covariate specific penalization terms. The penalization terms are inversely proportional to the ordinary least square (OLS) estimates from a multivariate regression model.

In this article, we combine the concepts of regularized regression with the biological background of differentially expressed genes. Genes that differ univariately with respect to the target, should be penalized less heavily.
We therefore introduce the term discriminative power (DP), which allows a covariate specific evaluation of compactness and separation with regard to the outcome. Discriminative power is measured by means of clustering indices [3], as well as by the classic concept of analysis of variance (ANOVA) [12].
The discriminative power is directly incorporated as covariate specific penalization into the adaptive Lasso, resulting in our approach Discriminative Power Lasso (DP-Lasso).
Using the DP as penalization weights in a L1-regularized model can be seen as a soft filtering as we do not exclude any covariates before performing regression, but favour genes with good univariate properties. The idea is to give covariates with low univariate DP a higher penalty, while reducing the penalty on the more promising covariates.

This paper is structured as follows. In Section II we introduce notations give an overview over commonly used regularization based methods. Section III introduces the DP-Lasso model. In Section IV and Section V we test the performance of DP-Lasso on scRNA-sequencing datasets as benchmark datasets, and on simulated data. Section VI concludes and provides an outlook.

II. Methods

In supervised learning, the goal is to estimate the underlying function that maps the p-dimensional covariate space to the outcome. As training data, we are given a matrix X, composed of p covariate vectors each containing the values of the N observations. This leads to the covariate matrix $X = (x_1, \cdots, x_p), j = 1, \cdots, p$, and the vector Y containing the N outcomes. x_{ij} denotes the value of observation i for covariate j, x_j the N values of covariate j, and $x_{i\cdot}$ the p dimensional observation vector for observation i. Given that the outcome is continuous, a common approach is to estimate the linear model

$$\hat{y}_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \quad , \tag{1}$$

where β is the p-dimensional vector of regression coefficients. In the following categorical outcomes $Y \in \{1, \cdots, K\}$ are considered. In this case a generalized linear model (glm) is appropriate, which uses a linear structure as in Equation 1 and connects it to the target through a link function [10]. Thus, for binary outcomes $Y \in \{0, 1\}$ logistic regression is used and for $K > 2$ classes the multinomial-logit model. However, for ease of notation in the following the linear model is used in the description of the methods.

In high dimensional data and especially $p >> N$ glms cannot reliably be estimated, due to the problems of multicollinearity and perfect separation [1, 14]. Also glms can not deal efficiently with irrelevant predictors, as no variable selection is performed. It is therefore common practice to reduce the number of genes before analysis.

For this purpose, the univariate filtering approach selects covariates based on (adjusted) p-values of univariate tests or biological reasoning. The final result highly depends on the researcher's choice, because a threshold or number of genes kept for the analysis has to be specified.

Alternatively, one can use regularized regression models for parameter estimation, that find a trade-off between model fit and model complexity. Regularized regression models also lead to more stable solutions for β coefficients in $p >> N$, as extreme behaviour is penalized [15]. This allows to find a unique solution in situations where glms might fail, such as perfect seperability and multicolinearity.

In regularized regression models, the overall loss function is decomposed in the discrepancy of the observed target and the model prediction and a penalty term that controls the complexity of the model. In case of the classical Lasso, the penalty is equal to the L1-norm of the coefficients β, leading to the overall loss function [31]:

$$L(y, X, \beta, \lambda, w) = \underbrace{\sum_{i=1}^{N} (y_i - x_{i\cdot}\beta)^2}_{\text{SSE}} + \underbrace{\lambda \sum_{j=1}^{p} |\beta_j|}_{\text{Penalty Term}}, \tag{2}$$

for linear regression. The degree of shrinkage and sparsity is controlled by a global shrinkage parameter λ, which is usually chosen via cross-validation.

Lasso regression allows to shrink coefficients to exactly zero, which leads to a covariate selection. Lasso has efficient solvers available, making it a good choice for high dimensional datasets. However, the Lasso has the known deficiency of overshrinkage: To remove a large number of uninformative covariates, a high penalty parameter needs to be chosen. This in return will also shrink the coefficients of informative predictors to some extent. To counteract, the Lasso will take in correlated predictors, to substitute for the overshrinkage [35]. This makes the interpretation of covariates left in the final model somewhat dubious, as it is unclear if the covariate itself is important or just as a substitute for the overshrinkage of another covariate.

If predictive performance is the primary objective, Ridge regression (L2-penalty) is a popular alternative. L2-penalty limits the influence of individual covariates, by penalizing high β's strongly, but shrinks no coefficient to exactly zero [15].

The Elastic Net (Zou and Hastie 2005) uses a mixture of the L1-norm (Lasso) and the L2-norm (Ridge). The loss function of the Elastic Net can be written as

$$L(y, X, \beta, \lambda, w) = \sum_{i=1}^{N} (y_i - x_{i\cdot}\beta)^2 + \alpha \sum_{j=1}^{p} \lambda_j |\beta_j| + (1 - \alpha) \sum_{j=1}^{p} \lambda_j \beta_j^2, \tag{3}$$

where α is a mixing parameter that controls the proportion of L1 and L2-penalty that is put on the coefficients.
Elastic Net often shows better predictive performance than Lasso, while also being able to set coefficients to exactly zero.

To reduce the amount of over-shrinkage and improve variable selection consistency, the adaptive Lasso [36] was proposed. Instead of using the same global shrinkage λ on every coefficient, the adaptive Lasso uses a covariate specific shrinkage parameter λ_j, which allows a separate penalty for each covariate. This leads to the loss function of the adaptive Lasso [36]

$$L(y, X, \beta, \lambda, w) = \sum_{i=1}^{N} (y_i - x_{i\cdot}\beta)^2 + \sum_{j}^{p} \lambda_j |\beta_j|, \tag{4}$$

where $\lambda_j = \lambda w_j$ is the covariate specific penalty and w_j discount factors that increase or decrease the amount of penalization for covariate j. In the original adaptive Lasso, w_j is calculated as the inverse of the parameter estimates of the ordinary least squares (OLS) regression, hence $w_j = 1/\hat{\beta}_j^{(OLS)}$. This approach can be shown to improve the model selection consistency under certain assumptions [36]. More concretely this results in less penalization of important covariates with high $\hat{\beta}_j^{(OLS)}$, which allows the final coefficients to become large, mitigating the over-shrinkage effect. In case of $p >> N$, the covariate specific weighting can be obtained by a ridge regression instead of the OLS estimates.

Several other extensions of the Lasso have been proposed, such as the fused Lasso [32], group Lasso [20], Bayesian Lasso [22] and Bayesian shrinkage priors [2].

Another commonly used approach for gene selection is the usage of tree ensembles, such as random forests [8]. Random forests [4], that combine several decision trees, are a popular choice for genetic classification data, as they posses strong predictive performance and do not require further assumptions. Measures, such as (unbiased) variable importance [29] and SHAP values [17] can be used to assess the importance of individual covariates, to rank covariates and to identify the most impactful genes.

III. Discriminative Power Lasso

In $p >> N$ situations, where the number of covariates exceeds the number of observations, there always exists an infinite amount of solutions for the regression hyperplane defined by the regression coefficients. While regularization helps to promote sparsity and limit extreme behaviour, we argue that additional information can guide the model towards more robust and reliable solutions. In contrast to the original adaptive Lasso, we want to limit the impact of covariates that only work well in a multivariate model, but are not discriminative univariatly. If enough data is available, such interplay between different covariates can be reliably estimated. However, with limited training data, the chance of over-fitting on spurious relationships is high, when learning multivariate models. Therefore, we suggest to instead promote genes that decompose the data in 'natural' groups, measured by the univariate discriminative power based on the conditional distribution $f(X_j|Y), j = 1, ..., p$.
The construction of the DP can be motivated by the concept of analysis of variance that measures the impact of a grouping variable on a numeric outcome by their differences in means. Therefore, for the construction of the DP we use the dependent variable Y as independent variable that we condition on to explain the differences in X. This change in perspective adds new information that is unavailable in a purely supervised regression approach. Secondly, cluster validation measures that have been developed in unsupervised clustering theory can be applied. Instead of using the outputted cluster labels as groups, as it is usually done in unsupervised learning, we directly use the target labels Y as grouping. The discriminative power therefore measures how well a covariate decomposes the underlying groups in terms of compactness and separation.

A. Target Adaptive Regularization

We implement the preference towards covariates with high discriminative power by discounting their penalty, similarly to the adaptive Lasso. The overall loss function of DP-Lasso can be written as

$$L(y, X, \beta, \lambda, w) = \mathcal{E}(\hat{y}, y, \beta) + \sum_{j=1}^{p} \lambda_j |\beta_j|, \tag{5}$$

where $\mathcal{E}$ is an appropriate loss function measuring the deviation from the fitted response vector $\hat{y}$ and the true values y, using a suitable link function. For logistic regression deviance or log-loss are common choices for $\mathcal{E}$. In case of a linear model the model takes the form of Equation 4. We propose to chose the covariate specific penalty as $\lambda_j^{(DP)} := \lambda w_j^{(DP)}$ and $w_j^{(DP)} = 1/DP_j$, where DP_j is the discriminative power of gene j. This gives the model a gentle push towards covariates that appear more natural and reliable, based on their DP. Note that both the calculation of DP and the following regularized regression model are based on all N observations of the training data.
Combining the DP with the supervised approach enriches the regression model with new information. Covariates with high DP are more likely to be selected in the final model, whereas covariates, that only work well in a multivariate model, but have a low individual DP are more likely to be removed. The adaptive shrinkage parameter also counteracts the over-shrinkage. Coefficients of covariates that work well in the multivariate model and also appear as good candidates, based on their DP, will be penalized less heavily and will be allowed to become large. On the other hand, clearly uninformative covariates with a low DP will receive an even higher penalty and can be removed more easily in the regularization step. Lastly, if several solutions to Equation 5 are similarly good, our approach gives a gentle push towards covariates that appear more trustworthy.

B. Characterization of natural groupings

This section motivates the construction of our DP measures. In general, we assume covariates X_j in which the underlying groups Y are homogeneous and well separated from the other groups as more promising . This reflects the idea that relevant genes should express differently among the K classes. Figure 1 shows the distribution of two example genes from the later used single-cell RNA-sequencing dataset EMTAB2805 of [5]. For the gene on the left side, we can see that the two underlying classes show clear differences in their distribution. Also the two groups are relatively compact and their group-means well separated. For the gene on the right side, the two groups show a stronger overlap, and they are less separated. Therefore, the gene on the left side appears to be a more natural candidate for a decisive gene and should have a higher chance of being selected.
The same rationale can be used for $K > 2$. Figure 2 shows

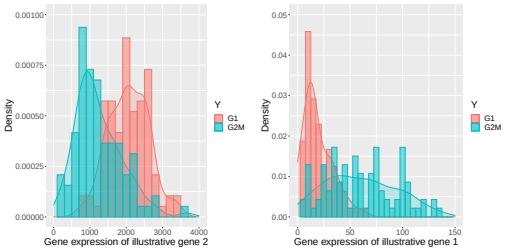


Fig. 1. Univariate distributions of two genes. The colors indicate the two groups. Left side: the two classes show clear differences in their distribution. Right side: the distributions are strongly overlapping with no clear difference.

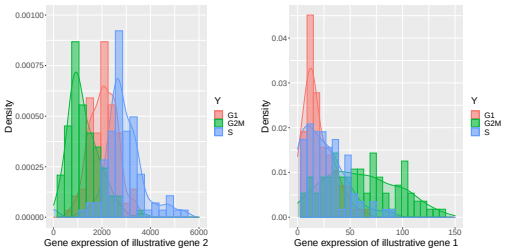


Fig. 2. Univariate distributions of two genes. The colors indicate the three groups. Left side: the three classes show clear differences in their distribution. Right side: the distributions are strongly overlapping with no clear difference.

the univariate distributions for three classes on the same genes, which can be used to assess the compactness and separation. The idea of DP-Lasso is therefore to prefer genes that decompose nicely into the underlying classes with regard to compactness and separation. We call this concept of 'natural grouping' the discriminative power DP. Genes with a high discriminative power will be favored in the regularization step (see Section III-A).

When using for example a logistic regression model, compactness of the groups (as an indication of naturality of the group) is not directly evaluated. The same goes for the distance between groups (or their means): As long as the groups are perfectly separable by a hyperplane, as is the case in $p >> N$, the margin to the discrimination plane is typically not considered in the loss function. Figure 3 shows two simulated covariates with a similar slope from a logistic regression model. While the two classes can be separated similarly good in both covariates, intuitively we would prefer the covariate shown on in right side, due to its distribution. Here the two classes express differently and the two groups are both compact and well separated, whereas the distribution on the left side appears more likely to be random. These descriptive illustrations aim to motivate the inclusion of additional information into the penalization by the discriminative power, which is described in the following.

The natural decomposition can be formalized by the concepts of compactness and separation with respect to the response.

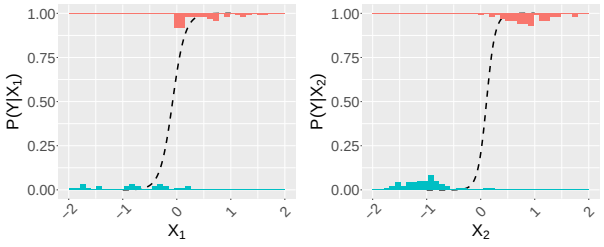


Fig. 3. The graph shows simulated genes, that can similarly well discriminated by a logistic regression. Left side: the clusters appear unnatural. Right side: compact groups with well separated group means.

C. Measures of discriminative power (DP)

In the following we describe three interesting options to measure the discriminative power. The goal is to capture information about the compactness and seperation between classes in each gene. The discriminative power is therefore calculated univariatly over each covariate j using the target variable y as grouping. In the following

$$x_j^{(k)} = \{x_{ij} : y_i = k\}_{i=1}^{N} \tag{6}$$

denotes the set of values of covariate j that belong to observations with the target class k, and $x_{hj}^{(k)}$ denotes the covariate values of the h'th observation in class k.
There exists a large number of quality criteria that are commonly used in unsupervised learning to evaluate clustering solutions. Also the idea of discriminative power can be interpreted as a classical test problem. The following describes three ways to measure DP, based on these principles.

1) ANOVA-approach: One classical way to test for differences in group means is the analysis of variance (ANOVA) [12]. Intuitively, the ANOVA expresses how much of the sample variance can be explained by the grouping. More concretely, the ANOVA tests whether there is a difference in the means of K groups based on its F-statistic.
Let $\bar{x}_j^{(k)} = \frac{1}{n_k}\sum_{h=1}^{n_k} x_{hj}^{(k)}$ denote the class mean of covariate j in target class k, where n_k is number of observations belonging to class k and $\bar{x}_j$ denotes the overall mean over all N observations. The according test statistic F_j measures the ratio of between-group variability and within-group variablity of covariate j via

$$F_j = \frac{(N-K)}{(K-1)} \frac{\sum_{k=1}^{K} n_k (\bar{x}_j^{(k)} - \bar{x}_j)^2}{\sum_{k=1}^{K}\sum_{h=1}^{n_k} (x_{hj}^{(k)} - \bar{x}_j^{(k)})^2}. \tag{7}$$

The value of the F-statistic is large in case that the distances between the groups are considerably higher than the distances within the groups. The higher the F-statistic, the higher the proportion of variance explained by the grouping, indicating significant differences in class means. We thus use the value of the F-statistic as one possibility for the measurement of discriminative power and determine the discount factor w_j^{DP} for the penalization in the subsequent step with $w_j^{(ANOVA)} = 1/F_j$. As $1/F_j$ can become quite large we use

a logarithmic transform to attenuate the differences in DP between the genes and to avoid numerical instabilities.

2) Davies-Bouldin Index: The Davies-Bouldin index *DB* measure was developed for validating the clustering quality based on compactness and separation of the clusters [6]. As mentioned before, instead of evaluating a cluster solution, the K classes are evaluated. The DB index relates the compactness within the groups to the separation between the classes. The compactness of class k is measured root mean squared error of observations from class k to the class mean $\bar{x}_j^{(k)}$ of class k in covariate j, leading to

$$\Delta_j^{DB}(k) = \sqrt{\frac{1}{n_k}\sum_{h=1}^{n_k}(x_{hj}^{(k)} - \bar{x}_j^{(k)})^2},$$

which in the univariate case simplifies to the standard deviation of observations in group k. The separation between the groups k and l groups is measured via the Euclidian distance of their respective class means $\bar{x}_j^{(k)}$ and $\bar{x}_j^{(l)}$, which in the univariate case simplifies to

$$\delta_j^{DB}(k,l) = |\bar{x}_j^{(k)} - \bar{x}_j^{(l)}|.$$

The overall DB Index is then given as

$$DB_j = \frac{1}{K}\sum_{k=1}^{K}\max_{l\neq k}\left\{\frac{\Delta_j^{DB}(k) + \Delta_j^{DB}(l)}{\delta_j^{DB}(k,l)}\right\}, \qquad (8)$$

which compares each class to its closest class, as a more pessimistic measure. The better the groups are separated and compact, the lower the DB index becomes, and as a consequence, the less penalization this covariate should be subjected to. Therefore, the discount factor is taken as $w_j^{(DB)} = DB_j$.

3) Silhouette Index: The silhouette index S_j [24] considers the compactness and separation evaluated on the individual level. For the construction of the 'silhouette width' s_{ij} the closeness of observation i to all observations within its group $k = y_i$ is measured via

$$\Delta_j^{Sil}(i,k) = \frac{1}{(n_k - 1)}\sum_{h:y_h=k, h\neq i}|x_{ij} - x_{hj}^{(k)}|, \qquad (9)$$

which is similar to the compactness measure in the DB index. However, Δ_j^{Sil} takes the closeness to each individual observation into account, instead of measuring the deviation from the mean.
Seperation between the groups is measured via,

$$\delta_j^{Sil}(i,k) = \min_{l\neq k}\left\{\frac{1}{n_l}\sum_{h=1}^{n_l}|x_{ij} - x_{hj}^{(l)}|\right\}, \qquad (10)$$

which takes the minimum average distance to the members of any other class. The silhouette width s_{ij} combines compactness and separation which leads to

$$s_{ij} = \frac{\delta_j^{Sil}(i,k) - \Delta_j^{Sil}(i,k)}{\max\{\Delta_j^{Sil}(i,k), \delta_j^{Sil}(i,k)\}}. \qquad (11)$$

As a last step, the silhouette index S_j is calculated by averaging over the silhouette width s_{ij} of all N individuals,

$$S_j = \frac{1}{N}\sum_{i=1}^{N} s_{ij} \quad \in \quad [-1,1]. \qquad (12)$$

S_j which can be used as a global measure of clustering quality given the covariate j and the target classes.
The absolute silhouette index takes values close to 1, if all observations are compact within their groups and well separated to the other groups. The more the silhouette index S_j approaches 0, the less compact the observations are within their groups and the less separated among covariate j. In this case the groupings are not nicely decomposed, and therefore this covariate is considered as less decisive.
The higher the absolute value of the silhouette index of covariate j, the better the distinction of the two underlying groups. Covariates with a high absolute silhouette index should be penalized less, therefore we set $w_j^{(Sil)} = 1/|S_j|$.

IV. Empirical Comparison

In this section we first present the scRNA-sequencing benchmark data and test the performance of DP-Lasso with different choices for the DP against competing methods. For both the binary classification, described in Section IV-C and the multiclass classification, described in Section IV-D, we perform a $5-$times repeated $10-$fold cross validation. As the supervised model is based on the classes present in the training data, we can only predict the number of underlying classes that are part of the training data set, in contrast to unsupervised clustering models.

A. Single-cell RNA-sequencing data (ScRNA-Seq data)

Based on the paper of [28], we use the same single-cell RNA-sequencing datasets as [16]. As proposed by [28], we only include genes into our analysis with read counts higher than 1 transcript per million mapped read (TPM) in more than 25% of the considered cells. This leads to a differing number of covariates p in case of the binary classification and the multiclass classification task, as shown in Table I. For the choice of cell types, we use the same selection as [16]. In case of the binary response, two selected cell types will be analyzed (left side of Table I). In case of the multiclass classification task (right side of Table I), we analyze K cell populations. The underlying numbers of cells in case of the binary response ($K = 2$) are N_1 and N_2, and for the multiclass response ($K > 2$) the respective cell populations are denoted with $N_1, \cdots, N_K$.
In accordance to the paper of [16], we consider their proposed binary classification tasks. However, instead of their approach of all pairwise combinations, we use a multinomial model for the $K > 2$ cases, which means one model per dataset. In the following, the cell types of the analyzed single-cell RNA-sequencing datasets are described. The EMTAB2805 data of [5] contain the cell cycle stages *G1, S, G2M* of the mouse embryonic stem cell (mESC). For the dataset GSE45719 [7] we include the different states of transition of *mid blastocyst*,

TABLE I
BENCHMARK DATA, SHOWING THE NUMBER OF COVARIATES p, NUMBER OF OBSERVATIONS N, AND THE OBSERVATIONS PER CLASS N_1 VS. N_2 IN THE BINARY CLASSIFICATION TASK AND N_1 VS. N_2 VS. $\cdots$ VS. N_K IN THE MULTICLASS CLASSIFICATION TASK

	Binary Response				Multiclass Response			
	EMTAB2805	GSE45719	GSE48968	GSE74596	EMTAB2805	GSE45719	GSE48968	GSE74596
p	13,110	10,851	7,987	6,748	12,849	11,065	7,831	7,329
Subpopulation 1	*G1*	*mid blastocyst*	BMDC 1h LPS	NKT0	*G1*	*mid blastocyst*	BMDC 1h LPS	NKT0
N_1	96	60	96	45	96	60	96	45
Subpopulation 2	*G2M*	*16-cell stage embryo*	BMDC 4h LPS	NKT17	*G2M*	*16-cell stage embryo*	BMDC 4h LPS	NKT17
N_2	96	50	191	44	96	50	191	44
Subpopulation 3	-	-	-	-	*S*	*8-cell stage embryo*	*BMDC 6h LPS*	*NKT1*
N_3	-	-	-	-	96	37	191	46
Subpopulation 4	-	-	-	-		-	-	NKT2
N_4	-	-	-	-	-	-	-	68

8-cell stage embryo as well *16-cell stage embryo*. In case of the single-cell RNA-sequencing data of GSE48968 bone marrow-derived dendritic cells (BMDCs) were stimulated with three different pathogenic components, analyzing the different responses for the dataset [25]. We will analyze only the component Lipopolysaccharides (LPS) at different timepoints (*1h, 4h, 6h*) after incubation. The data set GSE74596 contains different types of Natural killer T (NKT) cells extracted from the thymus. The cell types *NKT1, NKT2 and NKT17* are subtypes of the helper T cells [9].
The objective is to determine a supervised model that can classify the different cell types, given the expression profiles in these datasets. Also, as a second objective it is important to find a sparse solution to focus on the most important genes.

B. Competing Methods

The L1-regularized regression is carried out with the R package *glmnet* [13]. The λ values are found via the internal 10-fold CV approach and chosen as the value λ leading to the smallest estimated generalization error. For adaptive Lasso, the covariate specific penalty weights are determined with ridge regression $w_j = 1/\hat{\beta}_j^{Ridge}$ due to the $p >> N$ situation.
We also compare our methods to the Elastic Net, as a baseline for good predictive performance. Elastic Net is fit using *glmnet* and $\alpha = 0.5$, leading to a equal mixture of L1 and L2-penalization (cf. Section II).
For DP-Lasso the ANOVA based DP weights are implemented with the R package *stats* [23]. The Silhouette Index is calculated with the R package *cluster* [18] and the Davies-Bouldin Index with the package *clusterSim* [34]. The final DP-Lasso model is again fit using the *glmnet* procedure, using the covariate specific penalty weights derived from the DP.

C. Binary classification

In this section the results for the experiments on binary classification tasks are presented and analysed.

1) Accuracy – Binary: Accuracy is measured in terms of the misclassfication rate, averaged over all folds. The results of the empirical comparison can be found in Table II. Elastic Net shows overall the lowest misclassifcation rate, however the difference to the DP-Lasso models and the normal Lasso is only marginal. The only exception is the adaptive Lasso, which performs clearly worse compared to the other methods. This is likely due to the strong correlation present in the data. The three proposed DP-Lasso model show only minor differences in terms of misclassification rate, with a slight advantage for DP-L$_{ANOVA}$. We conclude, that the accuracy of DP-Lasso is comparable to the competitors irrespective the choice of the discriminative power.

2) Number of Coefficients – Binary: If the primary objective is to identify biomarkers, it is very important to find sparse solutions, as the cost of follow up studies can be high. Next, we therefore analyse the number of covariates selected by each method, which is the number of non-zero coefficients left in the regularized model. Out of all methods, the Elastic Net (Enet) selects the highest number of covariates, which is expected, due to its part of L2-penalty.
All DP-Lasso models select significantly fewer covariates than the competing methods, on all binary classification tasks. Often the difference is quite large. For example on the GSE74596 dataset DP-L$_{Anova}$ selects only 4 covariates, whereas Lasso selects 18. An likely explanation is the over-shrinkage effect in Lasso regression, which takes in irrelevant predictors (cf. Section II). On the other hand, DP-L$_{Anova}$ is able to reduce the penalty on the important covariates and reaches a very sparse solution.
From the class of DP-Lasso models, DP-L$_{ANOVA}$ is the most selective and finds the sparsest solutions. However, DP-L$_{DB}$ and DP-L$_{Sil}$ also produce smaller model sizes compared to the competing methods on all binary classification tasks.

D. Multiclass Classification

DP-Lasso can also be applied for multiclass ($K > 2$) classification. Note, that in case of $K > 2$ and the multinomial-logit model $K - 1$ coefficient vectors β are fit for the different categories, whereas one category is used as reference category. DP is measured as before for each covariate, leading to an equal penalization for each of the outcome categories.
In contrast to the binary case, the adaptive Lasso uses a different penalization weight for each covariate and outcome

TABLE II
THE MISCLASSIFICATION RATE FOR BINARY AND MULTICLASS CLASSIFICATION ON THE FOUR BENCHMARK DATASETS. THE BEST RESULT ON EACH DATASET (LOWEST NUMBER) IS MARKED IN BOLD.

	Binary				Multiclass			
	EMTAB2805	GSE45719	GSE48968	GSE74596	EMTAB2805	GSE45719	GSE48968	GSE74596
Lasso	0.05 (0.006)	**0.01** (0.000)	**0.02** (0.003)	**0.00** (0.000)	**0.06** (0.010)	0.03 (0.009)	0.19 (0.100)	**0.01** (0.003)
Elastic Net	**0.04** (0.006)	**0.01** (0.000)	**0.02** (0.000)	**0.00** (0.000)	**0.06** (0.007)	**0.02** (0.005)	0.18 (0.008)	**0.01** (0.004)
adaptive Lasso	0.11 (0.008)	0.02 (0.000)	0.07 (0.007)	0.15 (0.031)	0.17 (0.006)	0.10 (0.013)	0.26 (0.010)	0.28 (0.015)
DP-L$_{ANOVA}$	0.05 (0.006)	**0.01** (0.000)	**0.02** (0.004)	**0.00** (0.000)	**0.06** (0.009)	0.11 (0.017)	**0.17** (0.009)	0.03 (0.006)
DP-L$_{DB}$	0.05 (0.009)	**0.01** (0.000)	**0.02** (0.004)	0.01 (0.001)	0.08 (0.007)	0.07 (0.016)	0.20 (0.014)	0.03 (0.006)
DP-L$_{Sil}$	**0.04** (0.006)	**0.01** (0.000)	0.04 (0.004)	**0.00** (0.006)	0.18 (0.018)	0.06 (0.008)	0.24 (0.011)	0.06 (0.013)

TABLE III
THE NUMBER OF SELECTED COEFFICIENTS FOR BINARY AND MULTICLASS CLASSIFICATION ON THE FOUR BENCHMARK DATASETS. THE BEST RESULT (LOWEST NUMBER) ON EACH DATASET IS MARKED IN BOLD.

	Binary				Multiclass			
	EMTAB2805	GSE45719	GSE48968	GSE74596	EMTAB2805	GSE45719	GSE48968	GSE74596
Lasso	58 (1.9)	20 (0.4)	55 (0.9)	18 (0.6)	127 (3.5)	67 (1.0)	163 (5.5)	72(1.7)
Elastic Net	142 (1.8)	103 (1.1)	125 (1.2)	66 (0.5)	250 (13.1)	199 (1.5)	276 (10.2)	197 (1.9)
adaptive Lasso	38 (2.1)	13 (0.6)	48 (0.8)	27 (0.7)	65 (1.6)	36 (0.3)	84 (4.8)	52 (3.0)
DP-L$_{ANOVA}$	**17** (0.4)	**5** (0.1)	**19** (0.4)	**4** (0.2)	**45** (0.6)	**23** (1.2)	**70** (1.1)	**17** (0.5)
DP-L$_{DB}$	25 (0.9)	9 (0.1)	30 (0.3)	7 (0.1)	71 (1.3)	39 (0.8)	125 (1.6)	37 (0.3)
DP-L$_{Sil}$	22 (0.5)	9 (0.3)	36 (0.6)	8 (0.4)	181 (2.2)	32 (0.8)	172 (1.8)	90 (3.3)

category again resulting from the ridge estimator.

1) Accuracy – Multiclass: Accuracy is again measured as misclassification rate. The results can be found in Table II. Of all methods the Elastic Net shows the strongest predicitve performance, followed by the Lasso. the adaptive Lasso again performs clearly worse on all datasets in terms of accuracy. From the DP-Lasso models, DP-L$_{DB}$ is competitive on most datasets, and DP-L$_{ANOVA}$ remains competitive on three of the datasets and shows significantly worse performance on the GSE45719 data. DP-L$_{Sil}$ performs worse overall in the multinomial setting, but still notably better than the adaptive Lasso.

2) Number of Coefficients – Multiclass: In terms of model size, DP-L$_{ANOVA}$ again uniformly produces the sparsest solutions on all datasets. Lasso and Elastic Net keep around 3 to 10 times more non-zero coefficients in the model respectively. DP-L$_{DB}$ also produces relatively small models, on par with the adaptive Lasso, whereas DP-L$_{Sil}$ clearly struggles on the EMTAB2805, GSE48968 and GSE74596 datasets.

E. Empirical Results Summary

The empirical comparison on benchmark data indicates that DP-Lasso is able to maintain a high accuracy. At the same time DP-Lasso finds significantly smaller models, often by a factor of 3 to 10 compared to Lasso and Elastic Net. This is due to the incorporation of the DP into the penalization scheme, which helps to remove uninformative genes and instead focus on the relevant ones.

To summarise, DP-Lasso and especially DP-L$_{ANOVA}$ produces significantly smaller model sizes, while being able to maintain accuracy on par with current state-of-the-art regularized regression approaches.

V. SIMULATION STUDY

In this section, we test our method on simulated data. The setup is as follows. $X_1, ..., X_{10}$ are drawn from a normal distribution $\mathcal{N}(-1, \sigma)$, for observations of class 1, and from $\mathcal{N}(1, \sigma)$ for observations of class 2. This reflects the assumption that relevant genes express differently between the target groups. All additional covariates $X_{11}, ..., X_p$ are drawn from $\mathcal{N}(0, \sigma)$ and can therefore be considered as irrelevant. We test the values $p \in \{100, 1000, 5000\}$ and $\sigma^2 \in \{1, 2, 3\}$ and draw $N = 100$ observations in each setting. With increasing σ the groups become more overlapping and we expect learning to become increasingly difficult. Note that the covariates are drawn independently, implying $X \sim \mathcal{N}_p(\mu, \sigma^2 \mathcal{I}_p)$, where $\mathcal{I}$ is the identity matrix, making it an ideal situation for all methods. Each experiment is repeated 10 times and the results averaged. As in this experiment the relevant covariates are known, we measure the methods capabilities to identify the decisive covariates. To this end, we measure the Precision as

$$\text{Precision} = \frac{||\hat{\beta}_{true}||^0}{||\hat{\beta}||^0}, \tag{13}$$

where $||\cdot||^0$ specifies the 0-norm, which counts up the non-zero entries and $\hat{\beta}_{true}$ denotes the first ten entries of the coefficient vector, which by design we know to be the correct effects. $\hat{\beta}$ denotes all coefficients obtained by the regularized model. This measure is useful as the number of potential covariates is

TABLE IV
THE PRECISION AND RECALL ON THE DIFFERENT SIMULATION SETTINGS, AVERAGED OVER 10 RUNS. RESULTS ARE PRESENTED AS PRECISION / RECALL. FOR EACH SETTING THE METHOD WITH THE HIGHEST PRECISION IS MARKED IN BOLD.

	$\sigma^2 = 1$			$\sigma^2 = 2$			$\sigma^2 = 3$		
	$p = 100$	$p = 1000$	$p = 5000$	$p = 100$	$p = 1000$	$p = 5000$	$p = 100$	$p = 1000$	$p = 5000$
Lasso	0.86 / 0.99	0.60 / 0.99	0.53 / 0.98	0.45 / 0.96	0.32 / 0.96	0.23 / 0.93	0.48 / 0.95	0.37 / 0.84	0.28 / 0.84
Elastic Net	0.55 / 1.00	0.27 / 1.00	0.20 / 1.00	0.29 / 1.00	0.15 / 1.00	0.10 / 0.98	0.37 / 0.99	0.20 / 0.96	0.14 / 0.91
adaptive Lasso	0.99 / 0.97	0.97 / 0.98	0.94 / 0.95	0.88 / 0.98	0.58 / 0.96	0.37 / 0.91	0.71 / 0.93	0.35 / 0.82	0.28 / 0.85
DP-L$_{ANOVA}$	**1.00** / 0.87	**1.00** / 0.92	**1.00** / 0.85	**0.99** / 0.95	**0.88** / 0.93	**0.80** / 0.91	**0.82** / 0.87	**0.50** / 0.83	**0.38** / 0.85
DP-L$_{DB}$	**1.00** / 0.95	**1.00** / 0.94	**1.00** / 0.92	0.92 / 0.98	0.77 / 0.94	0.50 / 0.91	0.71 / 0.94	0.35 / 0.85	0.28 / 0.84
DP-L$_{Sil}$	**1.00** / 0.94	**1.00** / 0.94	**1.00** / 0.91	0.96 / 0.98	0.76 / 0.93	0.67 / 0.90	0.63 / 0.88	0.41 / 0.79	0.31 / 0.81

high. However, If the model has a high Precision, the identified genes can be trusted.
Secondly, we measure the Recall

$$\text{Recall} = \frac{||\hat{\beta}_{true}||^0}{10}, \tag{14}$$

as the fraction of the relevant covariates that was discovered by the model.
The results are shown in Table IV. We can see that the DP-Lasso models show significantly higher Precision compared to Lasso and Elastic Net. The adaptive Lasso performs better than the Lasso in this ideal setting, in contrast to the results on the real data from the previous section. Overall DP-L$_{DB}$ and DP-L$_{ANOVA}$ show the highest Precision, even in very difficult data situations. For instance, with $N = 100, p = 5000, \sigma = 1$, DP-L$_{ANOVA}$, DP-L$_{DB}$ and DP-L$_{Sil}$ are able to maintain a 100% Precision and thus are very selective and able to find the correct covariates. DP-L$_{ANOVA}$ has the highest Precision in every setting.
It is also important to compare the Recall, as it reflects the fraction of true effects that are found by a model. Elastic Net shows the highest Recall, which is a result of the large number of coefficients that was kept in the model. On the other hand, all DP-Lasso models show a Recall which is typically slightly lower but still competitive with Lasso and adaptive Lasso. This again is due to very selective nature of DP-Lasso.
Overall, we conclude that the non-zero coefficients found by the DP-Lasso can be trusted more to reflect true mechanisms, compared to its competitors. At the same time DP-Lasso is capable to maintain a competitive Recall.
It is reassuring to note that on average the accuracy measured by the area under the curve AUC of the methods is very similar, with a slight edge for the DP-L$_{DB}$, DP-L$_{ANOVA}$ and the Elastic Net.

VI. CONCLUSION

With DP-Lasso, we propose a novel regularization based approach for covariate selection in the context of gene expression data. Incorporating univariate measures of discriminative power that are based on the principles of separation and compactness enriches the model with additional information. Our approach can also be interpreted as soft filtering: instead of removing genes a-priori, more promising genes are simply promoted, freeing the modeller from ad-hoc choices, such as selecting the correct number of genes to remove. In a boarder context we argue that soft filtering, instead of hard filtering, therefore also enhances reproducibility, as it reduces the 'researchers degrees of freedom' [26] involved in a study.

Empirically, we show that DP-Lasso shows accuracy on par with the popular methods Lasso and Elastic Net, while choosing significantly less genes. With a simulation study we confirm that DP-Lasso is capable of ignoring a large number of irrelevant predictors and instead focusses on the truly relevant ones – due to the double criteria of being relevant both univariatly and in the multivariate model. This selectiveness is very desirable in the context of gene expression data, as both the number of candidate genes is high and follow-up studies are costly. Therefore, a short but confident list of very promising genes, as given by the DP-Lasso model, is preferred in this context.

As currently the discriminative power is calculated univariately, it does not explicitly take the correlation structure ofthe covariates into account. An interesting direction for future work would therefore be to extend the DP-Lasso approach by taking the correlation structure between covariates into account and adjust the penalization accordingly, similar to the approach in [33].

In this article, we focussed on the application for genetic classification data, however DP-Lasso can also be applied in other domains. As long as the classes are expected to show differences in the univariate distribution of covariates, we expect DP-Lasso to deliver a good predictive performance coupled with a low number of selected covariates.

ACKNOWLEDGEMENTS

The first authors are very grateful for the support of the LMU mentoring program, connecting young researchers and providing mentors that give individual advice. In addition, we would like to thank Gerhard Tutz and Christian L. Müller for the very insightful and valuable discussion.

References

[1] Adelin Albert and John A Anderson. On the existence of maximum likelihood estimates in logistic regression models. *Biometrika*, 71(1):1–10, 1984.

[2] Anindya Bhadra, Jyotishka Datta, Nicholas G Polson, and Brandon Willard. Lasso meets horseshoe: A survey. *Statistical Science*, 34(3):405–427, 2019.

[3] Nadia Bolshakova and Francisco Azuaje. Cluster validation techniques for genome expression data. *Signal Processing*, 83(4):825–833, 2003.

[4] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[5] Florian Buettner, Kedar N Natarajan, F Paolo Casale, Valentina Proserpio, Antonio Scialdone, Fabian J Theis, Sarah A Teichmann, John C Marioni, and Oliver Stegle. Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nature Biotechnology*, 33(2):155–160, 2015.

[6] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 224–227, 1979.

[7] Qiaolin Deng, Daniel Ramsköld, Björn Reinius, and Rickard Sandberg. Single-cell RNA-seq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science*, 343(6167):193–196, 2014.

[8] Ramón Díaz-Uriarte and Sara Alvarez De Andres. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7(1):1–13, 2006.

[9] Isaac Engel, Grégory Seumois, Lukas Chavez, Daniela Samaniego-Castruita, Brandie White, Ashu Chawla, Dennis Mock, Pandurangan Vijayanand, and Mitchell Kronenberg. Innate-like functions of natural killer T cell subsets result from highly divergent gene programs. *Nature Immunology*, 17(6):728–739, 2016.

[10] Ludwig Fahrmeir, Thomas Kneib, Stefan Lang, and Brian Marx. *Regression*. Springer, 2007.

[11] Liang Fang and Cheng Su. *Statistical Methods in Biomarker and Early Clinical Development*. Springer, 2019.

[12] Ronald Aylmer Fisher. Statistical methods for research workers. In *Breakthroughs in Statistics*, pages 66–70. Springer, 1992.

[13] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1, 2010.

[14] Joyee Ghosh, Yingbo Li, and Robin Mitra. On the use of cauchy prior distributions for bayesian logistic regression. *Bayesian Analysis*, 13(2):359–383, 2018.

[15] Trevor Hastie, Robert Tibshirani, and Jerome H Friedman. *The Elements of statistical learning: Data mining, inference, and prediction*. Springer, 2017.

[16] Beyrem Khalfaoui and Jean-Philippe Vert. DropLasso: A robust variant of Lasso for single cell RNA-seq data. *hal-01716704v2*, 2019.

[17] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4768–4777, 2017.

[18] Martin Maechler, Peter Rousseeuw, Anja Struyf, Mia Hubert, and Kurt Hornik. *cluster: Cluster Analysis Basics and Extensions*, 2019. R package version 2.1.0.

[19] M Man, TS Nguyen, C Battioui, and G Mi. Predictive subgroup/biomarker identification and machine learning methods. In *Statistical Methods in Biomarker and Early Clinical Development*, pages 1–22. Springer, 2019.

[20] Lukas Meier, Sara Van De Geer, and Peter Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):53–71, 2008.

[21] Masanori Oshi, Hideo Takahashi, Yoshihisa Tokumaru, Li Yan, Omar M Rashid, Ryusei Matsuyama, Itaru Endo, and Kazuaki Takabe. G2m cell cycle pathway score as a prognostic biomarker of metastasis in estrogen receptor (er)-positive breast cancer. *International Journal of Molecular Sciences*, 21(8):2921, 2020.

[22] Trevor Park and George Casella. The Bayesian Lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

[23] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020.

[24] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[25] Alex K Shalek, Rahul Satija, Joe Shuga, John J Trombetta, Dave Gennert, Diana Lu, Peilin Chen, Rona S Gertner, Jellert T Gaublomme, Nir Yosef, et al. Single-cell RNA-seq reveals dynamic paracrine control of cellular variation. *Nature*, 510(7505):363–369, 2014.

[26] Joseph P Simmons, Leif D Nelson, and Uri Simonsohn. False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, 22(11):1359–1366, 2011.

[27] Manabu Soda, Young Lim Choi, Munehiro Enomoto, Shuji Takada, Yoshihiro Yamashita, Shunpei Ishikawa, Shin-ichiro Fujiwara, Hideki Watanabe, Kentaro Kurashina, Hisashi Hatanaka, et al. Identification of the transforming EML4–ALK fusion gene in non-small-cell lung cancer. *Nature*, 448(7153):561–566, 2007.

[28] Charlotte Soneson and Mark D Robinson. Bias, robustness and scalability in single-cell differential expression analysis. *Nature Methods*, 15(4):255–261, 2018.

[29] Carolin Strobl, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin, and Achim Zeileis. Conditional variable importance for random forests. *BMC Bioinformatics*, 9(1):1–11, 2008.

[30] Fuchou Tang, Catalin Barbacioru, Yangzhou Wang, Ellen Nordman, Clarence Lee, Nanlan Xu, Xiaohui Wang, John Bodeau, Brian B Tuch, Asim Siddiqui, et al. mRNA-Seq whole-transcriptome analysis of a single cell. *Nature Methods*, 6(5):377–382, 2009.

[31] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[32] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[33] Gerhard Tutz and Jan Ulbricht. Penalized regression with correlation-based penalty. *Statistics and Computing*, 19(3):239–253, 2009.

[34] Marek Walesiak and Andrzej Dudek. The choice of variable normalization method in cluster analysis. In *Education Excellence and Innovation Management: A 2025 Vision to Sustain Economic Development During Global Challenges*, pages 325–340. International Business Information Management Association (IBIMA), 2020.

[35] Peng Zhao and Bin Yu. On model selection consistency of lasso. *The Journal of Machine Learning Research*, 7:2541–2563, 2006.

[36] Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

Undecided Voters as Set-Valued Information – Machine Learning Approaches under Complex Uncertainty

Dominik Kreiss[1], Malte Nalenz[2], and Thomas Augustin[3]

[1] LMU Munich, Department of Statistics, Ludwigstr. 33, Munich
dominik.kreiss@stat-uni.muenchen.de
[2] LMU Munich, Department of Statistics, Ludwigstr. 33, Munich
malte.nalenz@stat-uni.muenchen.de
[3] LMU Munich, Department of Statistics, Ludwigstr. 33, Munich
thomas.augustin@stat-uni.muenchen.de

Abstract. Undecided voters in pre-election polls, even though an increasing phenomenon and issue in electoral research, have mostly been neglected in conventional analysis so far. We argue to include this inherent form of uncertainty in a set-valued manner, in order to make the most of the valuable information, not improperly reducing voters' response to either an spuriously precise answer or to drop outs. The resulting consideration set consists of all elements the individual is still pondering between and can be interpreted in two ways, depending on the question at hand. First, for the sake of forecasting, it can be seen as a coarse version of the yet unknown element the individual ends up choosing, using the information for so-called epistemic modeling. Second, from an so-called ontic view, it can be seen as entity of its own, representing the individual's current position accurately and thus allowing to examine structural properties within the population. Both views provide good opportunities for machine learning. In this paper we introduce one exemplary approach based on each view, analysing structural properties using spectral clustering and forecasting using random forests, providing initial methodology for this type of complex, non-stochastic uncertainty. The theory is applied with constructed consideration sets to the most recent German federal election of 2017, using data from the *German Longitudinal Election Study*. The results are promising, laying the groundwork for further machine learning approaches concerning this natural type of inherent uncertainty.

Keywords: Epistemic imprecision · Ontic imprecision · Set-Valued Data · Consideration Sets · Random Forests · Spectral Clustering · Election

1 Introduction

Increasing numbers of undecided voters before an election[4] urge us to find new ways to deal with these individuals in statistical analysis and empirical election

[4] see for example [19,4]

research. Conventionally, the undecided voters are either forced by the questionnaire to give a precise answer or neglected in further analysis reliant on possibly unjustified assumptions (e.g. [17,15]). This leaves the undecided with the options to either over-simplify their position conveying incorrect information, or to drop out. Hence, recently in [17,16,15,12,13] the authors argue to include set-valued response options in surveys. Several arguments are put forward, like the reduction of nonresponse, the natural procedure or the more accurate representation of uncertainty. Despite these advantages, set-valued response options are regrettably not yet included in most surveys, also because methodology handling this type of information is in the beginning stages only. Thus, with this paper we contribute to a solution of the resulting "chicken-egg dilemma" [9, p. 7], providing approaches and ideas for such data.

Human choice generally, as argued by [16, p. 256], can be seen as a process in stages, excluding possibilities until arriving at one final element. Thus, at a given point in time before an election, which resembles a choice of N individuals amongst a finite set of alternatives $\{1, \cdots, s\} = S$, not every individual's position can be determined by only one element of the choice set. As several individuals are still pondering between options, the most accurate representation of their position is a set, excluding all options of S they will definitively not choose. This set, consisting in the case of a decided voter of one and a still undecided voter of several elements, determines naturally and accurately their position and will from now on be called *consideration set* following [16].

Indecision amongst voters is hereby a natural and very interesting example with practical relevance for the theoretical groundwork laid by Couso and Dubois (e.g. [8,7]). Following them, the resulting set-valued information can be interpreted in two ways, dependent on the question at hand. First, considering the election outcome, it can be seen as a coarse version of one true but at the time unknown element contained in the set, providing incomplete information. This is the so-called *epistemic* or *disjunctive* view. Second, focussing on the time point of the survey, the set represents the positions as a non-reducible entity of its own. This so-called *ontic* or *conjunctive* view regards a decided or undecided alike as a viable position with its own characteristics. Both views, even though very different, are justified, dealing with complementary issues.

In this paper we develop initial methodology for either view, providing first approaches and opportunities for machine learning to incorporate this set-valued information. With the ontic approach, regarding the undecided between specific parties as positions of their own, new structural properties concerning the political landscape can be examined. We generate socioeconomic clusters (using *spectral clustering*) and assess structural properties within the undecided and decided before the German federal election of 2017. For the epistemic view, we develop a forecasting approach incorporating the otherwise wasted information of the undecided. We hereby estimate *transition probabilities* of the undecided with *random forests* based on the decided individuals and provide an overall forecasting approach, reliant on simulation and assumptions, that is able to take the information of the consideration set into account. Both approaches are ap-

plied to data of the most recent German federal election of 2017, provided by the *German Longitudinal Election Study* [10] with constructed consideration sets.

This paper is structured as follows: First, in Section 2 we consolidate the ontic and epistemic methodology and introduce possible approaches for either view. We later apply the approaches to the most recent German federal election in Section 3. The concluding remarks in Section 4 reflect on the possibilities and challenges of this new way of incorporating undecided voters.

2 Methods

2.1 The Ontic and Epistemic Views

Dependent on the question at hand, a set consisting of the same elements can be interpreted in two different ways. To take a meanwhile classical example (e.g. [8]), if we are interested in the languages an individual is capable to speak, the set {English, French, German} is a precise representation of the truth, while if we are interested in the language he or she feels the most comfortable with, the same set contains only incomplete information. Equally, in the case of an undecided voter before an election, we can either focus on the indecision itself, which is accurately represented by the set as a whole, or focus on the choice outcome, in which case only incomplete information is provided. Thus, set-valued information obtained by a pre-election survey can be used in two different ways. Reflecting uncertainty in electoral analysis in a set-valued manner is a natural and especially interesting application for the theoretical groundwork laid by Couso and Dubois, presented for example in [8,7,3]. The state space of the consideration sets consist of all possible combinations of the original options, which can naturally be represented by the power set $P(S)$ of the set of the original options. Hence, in the case of an undecided, we are provided with a set ℓ that can be described as the realization of a measurable mapping $\mathcal{Y} : \Omega \to P(S)$ from some underlying space Ω into the set of all combinations. This set-valued representation can now be interpreted under ontic or epistemic imprecision.

Starting with the set as entity of its own, also called ontic or conjunctive interpretation, we consider undecided voters between specific parties as a further position. In this case, the consideration set is a precise representation of something naturally imprecise. Hence, it cannot be reduced or improved in any way. As the original choice set consists of finite elements measured on a nominal scale, the power set does as well, satisfying the same basic mathematical properties. Hence, methodology based on conventional approaches can broadly be transferred. Quite naturally, but most importantly, this protruding trait of ontic approaches opens up a wide range of options to apply state of the art machine learning approaches to data with this type of complex non-stochastic uncertainty. By this, the ontic view of undecided voters prior to the election enables new ways to examine structural properties within the political landscape.

The epistemic view, in contrast, focuses on the election outcome. Hereby, the set at the time point of the poll, accurately representing the position of an

undecided individual, is a coarse version of the one true element the individual ends up choosing. In other words, the set-valued information is an imprecise version of something precise. Thus, only incomplete information about the phenomena of interest (the eventual choice) is provided within the consideration set. To obtain statements about the precise value of interest, next to incorporating further information, one can make rather rigorous assumptions or reflect the uncertainty within interval-valued results. After all, we are only provided with incomplete information in the sense that $\forall\omega \in \Omega$ only $Y(\omega) \in \ell = \mathcal{Y}(\omega)$ is observable, with $\mathcal{Y}$ again as a mapping $\Omega \to P(S)$ now representing the set of mappings $\{Y : \Omega \to S, \forall\omega, Y(\omega) \in \mathcal{Y}(\omega)\}$, where we assume one of each is the true underlying mapping (e.g. [7, p. 1504]). As a consequence, reducing the set or assigning probabilities to each of its elements is usually strived for, in order to retrieve as precise information as possible about the variable of interest.

The following two sections reflect on possible applications of ontic as well as epistemic imprecision conducted with data from pre-election polls.

2.2 More on the Ontic Approaches

While in conventional pre-election voter analysis the undecided are neglected, we try to show in this section how including those individuals in a set-valued manner can open up new perspectives and findings about structural properties. The common procedure to monitor each month and regular before elections political orientations and developments in the political landscape of a country[5] could be enriched by these approaches, including further positions of interest. As the consideration sets are, as described in Section 2.1, the most accurate representation of the undecided, ontic approaches not only enable new findings, but also represent the current structural properties of the political landscape in the most accurate way. Several approaches are possible, examining different aspects of the political landscape concerning the undecided. Recently, as one example, we [12] extended discrete choice models with the undecided's consideration sets, providing new findings about the undecided in Germany.

For the ontic approach, we focus on the connection between socioeconomic clusters within the population and the undecided. Hereby, trends of indecisiveness could be located and assigned towards specific clusters. Thus, we cluster our data according to socioeconomic variables and examine structural differences of decided and undecided within the resulting socioeconomic groups. Conclusions from the composition of the clusters can then be interpreted from a political science perspective. We use spectral clustering (e.g. [18]) as a common machine learning approach for dividing our population in characteristics based on similarity in their covariate values. Hereby, we make use of the spectrum of a similarity matrix in order to perform dimensionality reduction and natural scaling on the data before clustering in fewer dimensions. The eventual clustering on this new data is usually performed by a simple algorithm like k-means.

[5] like for example in Germany the *Politbarometer* `https://www.forschungsgruppe.de/Aktuelles/Politbarometer/` last visited: 28.07.2020

The approach introduced in this paper is only meant to exemplify the opportunities of machine learning to describe this new type of data under ontic imprecision. It goes without saying, that there are numerous possibilities for straightforward applications of machine learning approaches, examining structural properties concerning the undecided, while already this rather simple one can initiate new ways to think about the political landscape.

2.3 More on the Epistemic Approaches

The epistemic approach, like sketched in Section 2.1, concerns itself with the yet unknown element in the consideration set the individual ends up voting for. Hence, in contrast to the ontic approaches addressing diverse questions, the epistemic ones try to improve forecasting, using the potentially valuable information of the undecided. As there is no information about the final choice of the undecided provided, either rather strong assumptions have to be made, or the uncertainty is manifested in the results using interval-valued identification. Thus, several approaches are possible, weighting the justifiability of assumptions with the precision of the results.[6] In a recent paper [13], we discuss this question, considering different approaches to incorporate the set-valued information into election forecasting, resulting in three different suggestions. Here, we pick up on the second one, achieving point-valued estimation by assuming that, given the covariates, the undecided choose identical to the decided with the consideration set as restriction of the possible outcomes.

Each individual holds a consideration set $\ell \in P(S)$ and covariates $X = x$ in some space $\mathcal{X}$. The consideration set is written as an event $\{\mathcal{Y} = \ell\}$ with $\ell \in P(S)$ and his or her possibly unknown choice on election day as $\{Y = l\}$ with $l \in S$. In order to estimate transition probabilities, the approach uses the distribution of the decided $P(Y = l|X = x, I_d = 1)$, which can be estimated from the data, with I_d as the indicator function for being decided. In order to incorporate the information of the consideration sets, all options not in ℓ are excluded. Therefore, scaling the estimates from the decided to comply with the multinomial distribution results in:

$$\underbrace{\hat{P}(Y = l|\mathcal{Y} = \ell, X = x)}_{Transition\ Probabilities} = \frac{\hat{P}(Y = l|X = x, I_d = 1)}{\sum_{a \in \ell} \hat{P}(Y = a|X = x, I_d = 1)} \tag{1}$$

leading to point-valued estimation of every parameter. Hence, to ensure point valued estimation, some implicit assumption of independent coarsening in the sense that undecided behave identical to the undecided is made. This resembles a random coarsening process, but satisfies mathematical properties different from the common CAR assumption of [11].

We utilize random forests [5] to estimate the conditional distributions for each undecided individual in Equation (1). Random forests grow a sequence of independent decision trees on bootstrap samples of the original data. At each

[6] also see Manski's Law of Decreasing Probability [14, p. 1]

node, only a subset of the covariates is used for splitting, efficiently reducing the correlation between the individual trees. These decorrelated, individually weak, trees are subsequently combined into an ensemble, typically through voting or by averaging the probability estimates. The resulting ensemble classifier was generally shown to significantly improve generalization performance and stability. As random forests are based on a set of decision trees, they posses several properties that are desirable in epistemic forecasting:

- They can naturally capture interaction effects between variables, without the need of prespecification.
- Non-linear effects can be appoximated. While single decision trees struggle to capture linear relationships, random forests can approximate them reasonably well.
- Both numeric and categorical covariates are natively supported without the need of any preprocessing.

Another reason to chose random forests over other popular ensemble methods, such as gradient boosting, is their stability towards a large grid of reasonable parameter choices [1].

As for the decided voters both the outcome Y and the covariate values X are known, random forests are applied directly, using the decided as training data. This implicitly presupposes, in accordance with above, that the conditional distributions of Y given the covariates are equal for decided and undecided voters, hence $P(Y = l|X = x, I_d = 1) = P(Y = l|X = x, I_d = 0)$. For easier reference in the discussion, we call this *structural similarity assumption*. Thus, for the undecided voters we can estimate the conditional multinomial distribution over all possible parties for each individual, using the structural similarity assumption. Note, however, that the random forest output is only a first level prediction, that is subsequently refined by taking into account the information given by the consideration sets, using Equation (1). This combines the predicitve power of random forests with the additional information given by the consideration sets. [7]

Provided with the estimated transition probabilities resulting from Equation (1), hence the probability an undecided chooses a particular party from their consideration set, we want to estimate the overall distribution together with the decided individuals. To this end, we use a Monte Carlo simulation approach: For the undecided we simulate precise decisions, drawing from the restricted multinomial distribution of each individual. Thus, the decided and the simulated data from the undecided can be used together for straightforward estimation of the overall distribution. In order to minimize the variance of the results, we repeat the process, averaging over the different estimates. The resulting point-valued estimates can be directly used for forecasting. Nevertheless, one should explicitly mention that the underlying assumptions are disputable.

[7] We do not use the undecided in the first level of estimation with some kind of simulation, in order to avoid strong assumptions about the final outcome in the consideration sets.

Thus, this approach can be seen as only a first example of how to integrate state of the art machine learning reliant on set-valued information of the undecided.

3 Application

3.1 The Data from The GLES

The ideas developed in Section 2.2 and 2.3 are applied for the most recent German federal election of 2017, using the state of the art pre-election poll conducted by the *GLES*[8]. Set-valued answer options are regrettably not included in this survey, but the assessment of the parties by the individuals and their statement about the certainty of their choice are, enabling construction of a consideration set as already conducted by [17, p. 261].

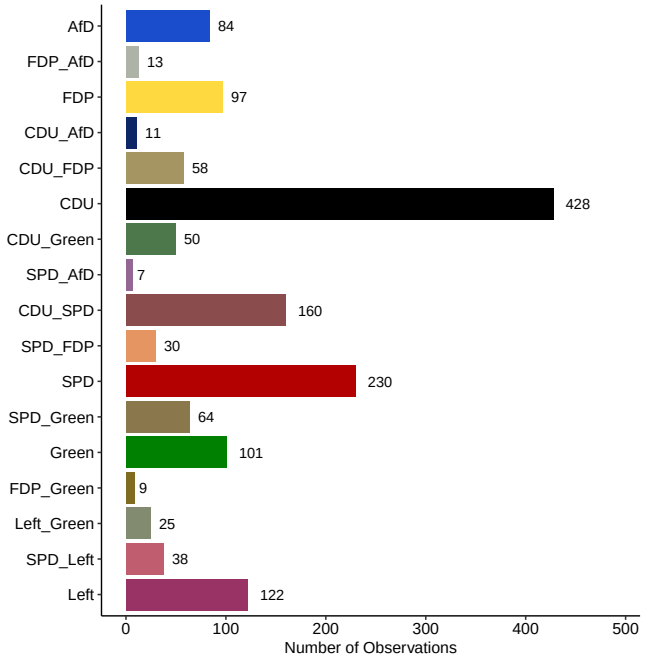


Fig. 1. The plot illustrates the distribution of the positions in our dataset, including decided and undecided individuals between exactly two parties. On the x-axis the numbers of observations and on the y-axis the corresponding position are shown.

[8] German Longitudinal Election Study: Pre- and post- election cross-section available under `https://www.gesis.org/wahlen/gles/daten`; last visited: 27.07.20

For our analysis, we use the so-called second vote[9] for the six main parties[10] anticipated to reach at least one seat in the parliament, in addition not including non-voters. As always in our illustrative example, structures of nonresponse in the dataset are not explicitly adjusted for. Moreover, we only focus on the most common case of indifference between exactly two parties.

The distribution of the positions in our data is illustrated in Figure 1. As one can see, the decided make up the major positions within this dataset, but 546 of the overall 1558 individuals are undecided, constituting one third of the population. A big proportion of the undecided is pondering between the two biggest and currently governing parties CDU and SPD with 160 observations, while there are few voters undecided between (combinations with) smaller parties in our dataset. These first descriptive results already hint towards a structural difference between the decided and undecided.

3.2 Clustering to Examine Ontic Structures

The approach sketched in Section 2.2 can be divided into two parts. First, we use spectral clustering with the three variables *age*, *household size* and *household income* to identify three separate socioeconomic groups within our population. The results are shown in Figure 2. While the first cluster mostly represents rather

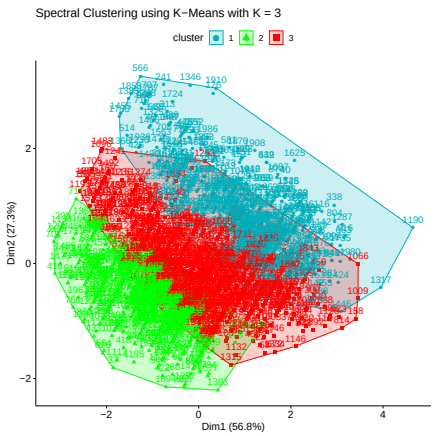


Fig. 2. This figure visualises the resulting three clusters using spectral clustering with the three variables *age*, *household size* and *household income* and k-means.

[9] The second vote basically determines the distribution of the seats among the parties, and thus is usually used for forecasting. For more information see: `https://www.bundeswahlleiter.de/en/bundestagswahlen/2021/informationen-waehler/wahlsystem.html`, last visited 27.07.20

[10] The parties are: AfD, FDP, CDU (including CSU), SPD, Green, Left

young and well earning individuals, living in a household with in average almost three individuals and the second one consist predominantly of pensioners, the third one is more intermixed. Considering we used three variables, the separation visualised in Figure 2 is proficient for our purposes.

Second, we examine the distribution of the consideration sets amongst the clusters as viable positions of their own. Thus, Figure 3 visualises the distribution of the positions, on the left side for the decided only and on the right side for the consideration sets, separate for the three clusters. As we can see, the

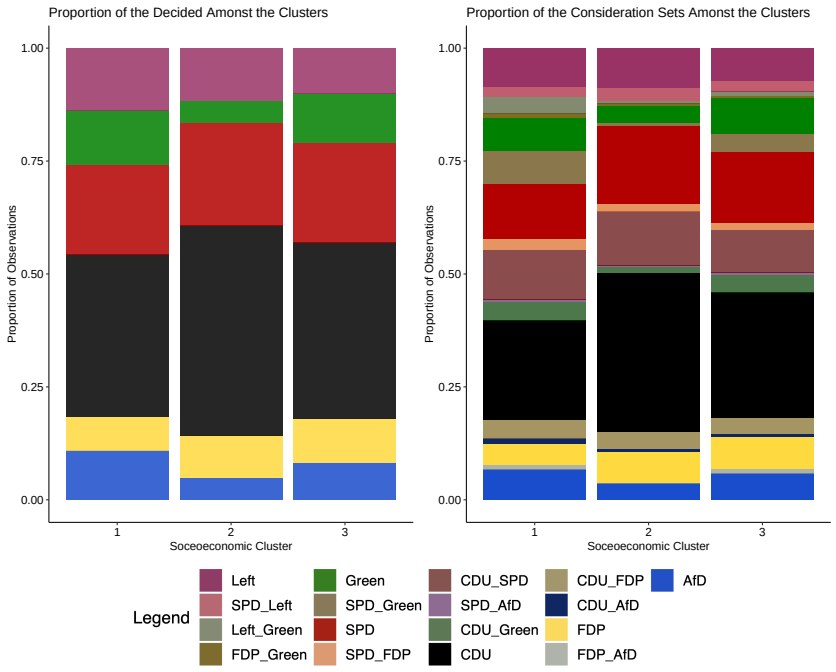


Fig. 3. This figure illustrates the composition of the three socioeconomic clusters, on the left for the decided only and on the right for the consideration sets.

positions are very unevenly distributed amongst the clusters. Notable, for example, is the high proportion of undecided between the Green and other parties within the first cluster, as mentioned above mostly consisting of young voters with comparable high income. The proportion of overall undecided is the highest within this first cluster in our data as well. Next to the insights into the political landscape, Figure 3 also shows structural differences between the decided and undecided. This underlines the importance of including undecided voters in electoral forecasting in order to avoid bias. The results of this first analysis are

therefore twofold. First, we examined structural properties, analysing predominate affiliation of specific undecided voters towards specific clusters. Second, we established structural differences between the decided and undecided.

3.3 Epistemic Forecasting

As described in Section 2.3, a random forest was applied using all available covariates, consisting of sociodemographic variables and several batteries of opinion questions. For training only the decided voters were used, as argued above. Using 10-fold cross validation on the decided voters led us to an estimated error rate of 25.4 %. This suggests that some of the covariates are clearly predictive. Furthermore, restricting the outcome space via the consideration sets adds important information. The Monte Carlo simulation to obtain overall estimates as explained in Section 2.2 is repeated 1000 times, leading to results illustrated in Figure 4 next to the ones only based on the decided.

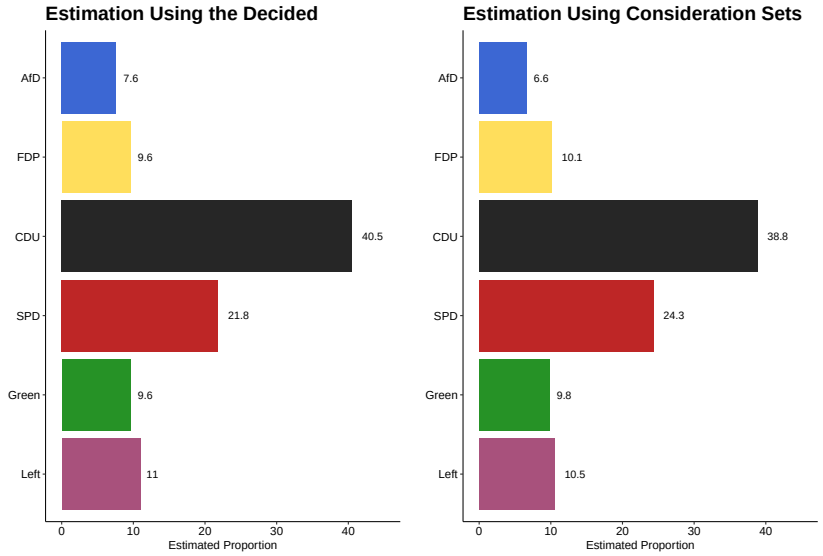


Fig. 4. The plot illustrates the forecasts of the overall distribution for the six main parties. On the left side based only on the decided and on the right incorporating undecided voters using random forest and simulation. The y-axis shows the six main parties while the x-axis shows the corresponding estimated proportion.

There are notable differences, stressing the impact of including the undecided. The biggest party CDU is less strongly represented including the undecided, while the SPD has a higher proportion. While the Green Party and FDP have

slightly higher estimates including the undecided, the wing parties AfD and Left Party have lower ones.

When drawing conclusion on political issues, one has to be cautious not to overinterpret our results, as the nonresponse structures are not adjusted for and the consideration sets had to be constructed. Nevertheless, including the undecided using random forests with the structural similarity assumption is straightforward applicable, providing first sound methodology which could be improved by further research.

4 Concluding Remarks

In this paper we proposed new ways to include the otherwise wasted information of undecided voters by making use of their consideration sets. For the ontic view, common methodology can broadly be transferred as the power set satisfies the same basic mathematical properties of the original data, while for the epistemic view, rather strong and untestable assumptions are necessary in order to obtain more concise forecasting. Thus, numerous approaches are possible, integrating machine learning into this natural type of uncertainty. While the ontic view focuses on new findings in structural properties, the epistemic one may improve election forecasting by including this valuable information.

We introduced one approach each, analysing structural properties with spectral clustering and extending forecasting reliant on the structural similarity assumption and random forests. Both approaches, even though not yet perfected, yield promising results. Thus, we provided initial methodology which must be further developed and improved. Concerning forecasting, new sources of information could be incorporated like decisions in previous elections or expert knowledge in a (generalised) Bayesian way. Furthermore, set-valued approaches are promising. This includes cautious data completion explicitly [2] (see also, e.g. for classifiers, [6]) as well as working in the spirit of partial identification following [14], permitting to weaken assumptions resulting in more credible results. For ontic approaches, discrete choice models are of particular interest, examining connections between attributes and indecision between specific parties. Hereby, highlighting attributes of individuals determined to vote for the right-wing party AfD compared to those only considering it, might provide essential insights into the trend towards nationalistic parties.

With this paper, we open up this complex uncertainty structure towards exciting applications for a broad spectrum of machine learning methodology.

Acknowledgement. We sincerely thank the anonymous reviewers for their helpful remarks. Further we thank the LMU mentoring, supporting young researchers, and the GLES for providing the dataset.

References

1. Aggarwal, C.C.: Outlier analysis. In: Data mining. pp. 237–263. Springer (2015)
2. Augustin, T., Walter, G., Coolen, F.: Statistical inference. In: Augustin, T., Coolen, F., de Cooman, G., Troffaes, M. (eds.) Introduction to Imprecise Probabilities, pp. 135–189. Wiley (2014)
3. Augustin, T., Coolen, F., De Cooman, G., Troffaes, M., (Eds.): Introduction to imprecise probabilities. Wiley (2014)
4. BBC: Why has the UK become a nation of political swingers? BBC News (2017), `https://www.bbc.com/news/uk-politics-39103972`, (Last visited 28.07.2020)
5. Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)
6. Corani, G., Zaffalon, M.: Learning reliable classifiers from small or incomplete data sets: the naive credal classifier 2. Journal of Machine Learning Research **9**, 581–621 (2008)
7. Couso, I., Dubois, D.: Statistical reasoning with set-valued information: Ontic vs. epistemic views. International Journal of Approximate Reasoning **55**(7), 1502–1518 (2014)
8. Couso, I., Dubois, D., Sánchez, L.: Random sets and random fuzzy sets as lll-perceived random variables. Springer (2014)
9. Fink, P.: Contributions to reasoning on imprecise data. Ph.D. thesis, LMU Munich, Faculty of Mathematics, Computer Science and Statistics (2018), `https://edoc.ub.uni-muenchen.de/22547/`
10. GLES: German longitudinal election study (2019), `https://www.gesis.org/wahlen/gles/`, (Last visited 28.07.2020)
11. Heitjan, D., Rubin, D.: Ignorability and coarse data. The Annals of Statistics pp. 2244–2253 (1991)
12. Kreiss, D.: Examining Undecided Voters in Multiparty Systems. Master's thesis, LMU Munich, Department of Statistics (2019), `https://epub.ub.uni-muenchen.de/70668/`
13. Kreiss, D., Augustin, T.: Undecided voters as set-valued information, towards forecasts under epistemic imprecision. In: Davis, J., Tabia, K. (eds.) SUM 2020. Springer (2020)
14. Manski, C.: Partial identification of probability distributions. Springer (2003)
15. Oscarsson, H., Oskarson, M.: Sequential vote choice: Applying a consideration set model of heterogeneous decision processes. Electoral Studies **57**, 275–283 (2019)
16. Oscarsson, H., Rosema, M.: Consideration set models of electoral choice: Theory, method, and application. Electoral Studies **57**, 256–262 (2019)
17. Plass, J., Fink, P., Schöning, N., Augustin, T.: Statistical modelling in surveys without neglecting 'The undecided'. In: Augustin, T., Doria, S., Miranda, E., Quaeghebeur, E. (eds.) ISIPTA 15, pp. 257–266. SIPTA (2015)
18. Von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing **17**(4), 395–416 (2007)
19. Zeit: Die Hälfte der Wähler hat sich noch nicht entschieden. Die Zeit: Online Newspaper (2017), `https://www.zeit.de/politik/deutschland/2017-08/bundestagswahl-umfrage-waehler-unentschlossen`, (Last visited 28.07.2020)

PLOS ONE

RESEARCH ARTICLE

A computational reproducibility study of PLOS ONE articles featuring longitudinal data analyses

Heidi Seibold[1,2,3,4]*, **Severin Czerny**[1], **Siona Decke**[1], **Roman Dieterle**[1], **Thomas Eder**[1], **Steffen Fohr**[1], **Nico Hahn**[1], **Rabea Hartmann**[1], **Christoph Heindl**[1], **Philipp Kopper**[1], **Dario Lepke**[1], **Verena Loidl**[1], **Maximilian Mandl**[1], **Sarah Musiol**[1], **Jessica Peter**[1], **Alexander Piehler**[1], **Elio Rojas**[1], **Stefanie Schmid**[1], **Hannah Schmidt**[1], **Melissa Schmoll**[1], **Lennart Schneider**[1], **Xiao-Yin To**[1], **Viet Tran**[1], **Antje Völker**[1], **Moritz Wagner**[1], **Joshua Wagner**[1], **Maria Waize**[1], **Hannah Wecker**[1], **Rui Yang**[1], **Simone Zellner**[1], **Malte Nalenz**[1]

1 Department of Statistics, LMU Munich, Munich, Germany, **2** Data Science Group, University of Bielefeld, Bielefeld, Germany, **3** Helmholtz AI, Helmholtz Zentrum München, Munich, Germany, **4** LMU Open Science Center, LMU Munich, Munich, Germany

* heidi@seibold.co

OPEN ACCESS

Citation: Seibold H, Czerny S, Decke S, Dieterle R, Eder T, Fohr S, et al. (2021) A computational reproducibility study of PLOS ONE articles featuring longitudinal data analyses. PLoS ONE 16(6): e0251194. https://doi.org/10.1371/journal.pone.0251194

Editor: Jelte M. Wicherts, Tilburg University, NETHERLANDS

Received: August 19, 2020

Accepted: April 13, 2021

Published: June 21, 2021

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: https://doi.org/10.1371/journal.pone.0251194

Data Availability Statement: All results including detailed reports and code for each of the 11 papers are available in the GitLab repository https://gitlab.com/HeidiSeibold/reproducibility-study-plos-one.

Abstract

Computational reproducibility is a corner stone for sound and credible research. Especially in complex statistical analyses—such as the analysis of longitudinal data—reproducing results is far from simple, especially if no source code is available. In this work we aimed to reproduce analyses of longitudinal data of 11 articles published in PLOS ONE. Inclusion criteria were the availability of data and author consent. We investigated the types of methods and software used and whether we were able to reproduce the data analysis using open source software. Most articles provided overview tables and simple visualisations. Generalised Estimating Equations (GEEs) were the most popular statistical models among the selected articles. Only one article used open source software and only one published part of the analysis code. Replication was difficult in most cases and required reverse engineering of results or contacting the authors. For three articles we were not able to reproduce the results, for another two only parts of them. For all but two articles we had to contact the authors to be able to reproduce the results. Our main learning is that reproducing papers is difficult if no code is supplied and leads to a high burden for those conducting the reproductions. Open data policies in journals are good, but to truly boost reproducibility we suggest adding open code policies.

Introduction

Reproducibility is—or should be—an integral part of science. While computational reproducibility is only one part of the story, it is an important one. Studies on computational reproducibility (e.g. [1–6]) have found reproducing findings in papers is far from simple. Obstacles

PLOS ONE

A computational reproducibility study of PLOS ONE articles featuring longitudinal data analyses

All files can also be accessed through the Open Science Framework (https://osf.io/xqknz).

Funding: This research has been supported by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A (Munich Center of Machine Learning) to HS.

Competing interests: The authors have declared that no competing interests exist.

include lack of methods descriptions and no availability of source code or even data. Researchers can choose from a multitude of analysis strategies and if they are not sufficiently described, the likelihood of being able to reproduce the results are low [7, 8]. Even in cases where results can be reproduced, it is often *tedious and time-consuming* to do so [6].

We conducted a reproducibility study based on articles published in the journal PLOS ONE to learn about reporting practices in longitudinal data analyses. All PLOS ONE papers which fulfilled our selection criteria (see Fig 1) in April 2019 were chosen ([9–19]).

Longitudinal data is data containing repeated observations or measurements of the objects of study over time. For example, consider a study investigating the effect of alcohol and marijuana use of college students on their academic performance [10]. Students perform a monthly survey on their alcohol and marijuana use and consent to obtain their grade point averages (GPAs) each semester during the study period. In this study not only the outcome of interest (GPAs during several semesters) is longitudinal, but also the covariates (alcohol and marijuana use) change over time. This does not always have to be the case in longitudinal data analysis. Covariates may also be constant over time (e.g. sex) or baseline values (e.g. alcohol consumption during the month before enrollment).

Due to the clustered nature of longitudinal data with several observations per subject, special statistical methods are required. Common statistical models for longitudinal data are mixed effect models or generalized estimating equations. These models can have complex structures and rigorous reporting is required for reproducing model outputs. A study on reporting in generalized linear mixed effect models (GLMMs) on papers from 2000 to 2012 found that there is room for improvement on reporting of these models [20]. Alongside the models, visualization of the data often plays an important role in analyzing longitudinal data. An example is the spaghetti plot, a line graph with the outcome on the y-axis and time on the x-axis. Research on computational reproducibility when methods are complex—such as in this case—is still in its infancy. With this study we aim to add to this field and to provide some

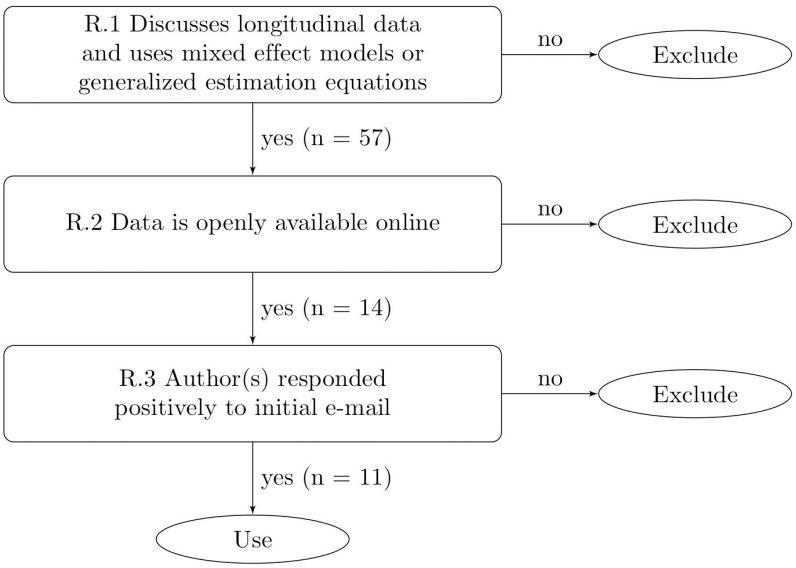


Fig 1. Data selection. Data selection procedure according to our requirements and number of papers fulfilling the respective requirements.

https://doi.org/10.1371/journal.pone.0251194.g001

insights on challenges of reproducibility in the 11 papers investigated. Furthermore we would like to note that each reproduced paper, is another paper that we can put more trust in. As such reproducing a single paper is already a relevant addition to science.

Computational (or analytic [21]) reproducibility studies—as we define them for this work—take existing papers and corresponding data sets and aim to obtain the same results from the statistical analyses. One prerequisite for such a study is the access to the data set which was used for the original analyses. Also, a clear description of the methods used is essential. An easily reproducible paper provides openly licensed data alongside an openly licensed source code in a programming language commonly used for statistical analyses and also available under a free open source software license (e.g. R [22] or python [23]). If the source code is accompanied with a detailed description of the computing environment (e.g. operating system and versions of R packages) or the computing environment itself (e.g. a Docker container [24]) we believe the chances of obtaining the exact same results to be highest. It is difficult to determine whether a scientific project is *reproducible*: Is it possible to obtain exactly the same values? Is the (relative) deviation lower than a certain value? Is the difference in p-value lower than a certain value? These and more are questions that can be asked and if answered "yes" the results can be marked as reproducible. Yet all of these come with downsides including being too strict, incomparable, uncomputable, or downright not interesting. Here, we use the definition of leading to the same interpretation, without a rigorous formal definition. The reason is, that the papers analysed here use very different models, so it is hard to compare them on a single scale (such as absolute relative deviation, see e.g. [6]). We argue, that in combination with a qualitative description of challenges and difficulties that we faced in each reproduction process, this definition fits our small scale, heterogeneous, setting better.

In this work we investigated longitudinal data analyses published in PLOS ONE. The multidisciplinarity of PLOS ONE is a benefit for our study as longitudinal data play a role in various fields. Additionally the requirement for a data availability statement in PLOS ONE (see https://journals.plos.org/plosone/s/data-availability) facilitates the endeavour of a reproducibility study. Note that we only selected papers which provided data openly online and where authors agreed with being included in this study. We assume that this leads to a positive bias in the sense that other papers would be more difficult to reproduce.

In the following we discuss the questions we asked in this reproducibility study, the setup of the study within the context of a university course, the procedure of paper selection, and describe the process of reproducing the results.

Materials and methods

Study questions

The aim of this study is to investigate reproducibility in a sample of 11 PLOS ONE papers dealing with longitudinal data. We also collect information on usage of methods, how they are made available and computing environments used. We expect that this study will help future authors in making their work reproducible, even in complex settings such as when working with longitudinal data. Note that based on the selection of 11 papers we cannot make inferences on papers in general or in the journal. We can, however, learn from the obstacles we encountered in the given papers. Also, even reproducing a single paper creates scientific value. It provides a scientific check of the work and increases (or in case of failure decreases) trust in the results.

With the reproducibility study we want to answer the following questions:

1. Which methods are used?

(a) What types of tables are shown?

(b) What types of figures are shown?

(c) What types of statistical models are used?

2. Which software is used?

(a) Is the software free and open source?

(b) Is the source code available?

(c) Is the computing environment described (or delivered)?

3. Are we able to reproduce the data analysis?

(a) Are the methods used clearly documented in the paper or supplementary material (e.g. analysis code)?

(b) Do we have to contact the authors in order to reproduce the analysis? If so, are authors responsive and helpful? How many e-mails are needed to reproduce the results?

(c) Do we receive the same (or very similar) numbers in tables, figures and models?

4. What are characteristics of papers which make reproducibility easy/possible or difficult/impossible?

5. What are learnings from this study? What recommendations can we give future authors for describing their methods and reporting their results?

Project circumstances

This project was conducted as part of the master level course *Analysis of Longitudinal Data* running during the summer term 2019 (23.01.19—27.07.19) at the Ludwig-Maximilians-Universität München. The course is a 6 ECTS (credit points according to the European Credit Transfer and Accumulation System) course aimed at statistics master students (compulsory in biostatistics master, elective in other statistics masters) with 4 hours of class each week: 3 hours with a professor (Heidi Seibold), 1 with a teaching assistant (Malte Nalenz). The course teaches how to work with longitudinal data and discusses appropriate models, such as mixed effect models and generalized estimating equations, and how to apply them in different scenarios. As part of this course, student groups (2-3 students) were assigned a paper for which they aimed to reproduce the analysis of longitudinal data. In practical sessions the students received help with programming related problems and understanding the general theory of longitudinal data analysis. To limit the likelihood of bias due to differing skills of students, all groups received support from the teachers. Students were advised to contact the authors directly in case of unclear specifications of methods. Internal peer reviews, where one group of students checked the setup of all other groups, ensured that all groups had the same solid technical and organizational setup. Finally all projects were carefully evaluated by the teachers and updated in case of problems. Replications and a student paper were the output of the course for each student group and handed in in August 2019. We believe that the setup of this reproducibility study benefits from the large time commitment the students put into reproducing the papers. Also having several students and two researchers work on each paper, ensures a high quality of the study.

This project involved secondary analyses of existing data sets. We had not worked with the data sets in question before.

Selection of papers

For a paper to be eligible for the reproducibility study it has to fulfill the following requirements:

R.1 The paper deals with longitudinal data and uses mixed effect models or generalized estimating equations for analysis.

R.2 The paper is accompanied by data. This data is freely available online without registration.

R.3 At least one author is responsive to e-mails.

Requirement **R.1** allows us to select only papers relevant to the topic of this project. Requirement **R.2** is necessary to allow for reproducing results without burdens (e.g. application for data access). Although PLOS ONE does have an open data policy (https://journals.plos.org/plosone/s/data-availability), we found many articles which had statements such as "Data cannot be made publicly available due to ethical and legal restrictions". Issues with data policies in journals have been studied in [25]. Requirement **R.3** is important to be able to contact the authors later on in case of questions. Fig 1 shows the selection procedure. All papers which did not fulfill the criteria were excluded. The PLOS website search function was utilized to scan through PLOS ONE published works. Key words used were "mixed model", "generalized estimating equations", "longitudinal study" and "cohort study". This key word search—performed for us by a contact at PLOS ONE—resulted in 57 papers. From these 14 papers fulfilled all criteria and were selected. Two authors prohibited to use of their work within our study. We note that authors do not have the right to prohibit the reuse of their work as all papers are published under CC-BY license. However the negative response lead us to drop the papers, as we expected to have the need to contact authors with questions. For one paper we did not receive any response. Discussions on the selection criteria of all proposed papers are documented in https://osf.io/dx5mn/?branch=public.

Table 1 shows a summary of all papers selected so far.

Replication

In the reproducibility study we adhered to open science best practices. (1) We contacted all corresponding authors of papers we aimed to reproduce via e-mail; (2) all of our source code and data used is available; (3) any potential errors in the original publications were reported immediately to the corresponding author.

In our study we conducted all analyses as close to the original analyses as possible. If many analyses were performed in the original paper, we focused on the analyses of longitudinal data. We conducted all analyses using R [22] regardless of the software used in the original paper to mimic a situation where no access to licensed software is available (R was the only open source software used in the 11 papers).

Each analyis consisted of the following steps:

1. Read the data into R.
2. Prepare data for analysis.
3. Produce overview figure(s) with outcome(s) on the y-axis and time on the x-axis.
4. Reproduce analysis results (e.g. model coefficients, tables, figures).

The description about all these steps was generally vague (see classification of reported results in [6]) meaning that there were multiple ways of preparing or analysing the data that were in line with the descriptions in the original paper. This study, thus, exposed a large

Table 1. Selected papers.

	Citation	Title
[9]	Wagner et al (2017)	Airway Microbial Community Turnover Differs by BPD Severity in Ventilated Preterm Infants
[10]	Meda et al (2017)	Longitudinal Influence of Alcohol and Marijuana Use on Academic Performance in College Students
[11]	Visaya et al (2015)	Analysis of Binary Multivariate Longitudinal Data via 2-Dimensional Orbits: An Application to the Agincourt Health and Socio-Demographic Surveillance System in South Africa
[12]	Vo et al (2018)	Optimizing Community Screening for Tuberculosis: Spatial Analysis of Localized Case Finding from Door-to-Door Screening for TB in an Urban District of Ho Chi Minh City, Viet Nam
[13]	Aerenhouts et al (2015)	Estimating Body Composition in Adolescent Sprint Athletes: Comparison of Different Methods in a 3 Years Longitudinal Design
[14]	Tabatabai et al (2016)	Racial and Gender Disparities in Incidence of Lung and Bronchus Cancer in the United States: A Longitudinal Analysis
[15]	Rawson et al (2015)	Association of Functional Polymorphisms from Brain-Derived Neurotrophic Factor and Serotonin-Related Genes with Depressive Symptoms after a Medical Stressor in Older Adults
[16]	Kawaguchi, Desrochers (2018)	A Time-Lagged Effect of Conspecific Density on Habitat Selection by Snowshoe Hare
[17]	Lemley et al (2016)	Morphometry Predicts Early GFR Change in Primary Proteinuric Glomerulopathies: A Longitudinal Cohort Study Using Generalized Estimating Equations
[18]	Carmody et al (2018)	Fluctuations in Airway Bacterial Communities Associated with Clinical States and Disease Stages in Cystic Fibrosis
[19]	Villalonga-Olives et al (2017)	Longitudinal Changes in Health Related Quality of Life in Children with Migrant Backgrounds

https://doi.org/10.1371/journal.pone.0251194.t001

amount of "researcher degrees of freedom" [26] coupled with a lack in transparency about in the original studies. We aimed to take steps that align as closely as possible with the original paper and the results therein. That means, if the methods description in paper or supplementary material were clear, we used those; If not, we tried different possible strategies that we assumed could be correct; If this was not possible or did not lead to the expected results, we contacted the authors to ask for help. All code used by us is publicly available including software versions and in a format easily readable by humans (literate programming, for further information see section on technical details).

Results

The results of our study are summarized in Tables 2–4. As each paper has its own story and reasons why it was or wasn't reproducible and what the barriers were, we provide a short description of each individual paper reproduction.

Which methods are used? For an overview on the following questions we refer to Table 2.

What types of tables are shown? Most of the papers show tables on characteristics of the observation units at baseline or other summary tables (similar to the so called "Table 1" commonly used in biomedical research) which give a good overview of the data.

What types of figures are shown? Few papers include classical visualizations taught in courses on longitudinal data, such as spaghetti plots. They mostly present other visualizations (for details, see Table 2).

Table 2. Which statistical methods were used by the papers?.

	Overview Tables	Visualisations	Models Used
[9]	Baseline demographics	Several, e.g. spaghetti plot	Beta Binomial Mixed Model
[10]	Baseline demographics, model output	Several, e.g. scatter plots (alkohol vs. marijuana use) of different time points	LMM
[11]	Overview of household types	Several, e.g. lasagna plot	GEE
[12]	Baseline demographics	none	GEE
[13]	Correlation	none	LMM (cross-classified)
[14]	Many especially smoking and lung cancer incidence rates for different year, genders, races and regions	Mean curves	LMM
[15]	Baseline demographics	Mean curves	GEE
[16]	Data overview	Mean curves	GEE
[17]	Correlation matrix	Mean curves	GEE
[18]	Sample characteristics	Several, e.g. FEV1 over time	GEE
[19]	Baseline demographics	DAG	GEE

https://doi.org/10.1371/journal.pone.0251194.t002

What types of statistical models are used? Although in most cases (G)LMMs are superior to GEEs (see [27] for an in-depth discussion and further references)—, 7 out of the 11 papers used GEEs for their analyses [11, 12, 15–19]. There is, in fact, only one complex mixed model among the methods used (Beta Binomial Mixed Model, [9]). The other articles [10, 13, 14] use LMMs which are equivalent to GEEs for normally distributed response variables. It should be noted that the selection of papers may not be representative of the general use of GEEs and (G)LMMs. Nevertheless it seems that the reluctance of using GEEs has not spilled over from the statistics community to some other fields, which we speculate to have historical reasons, as GLMMs used to be difficult to compute.

Which software is used? The results of this section are summarized in Table 3.

Is the software free and open source? All except one paper (paper [16]) used closed source software. As our goal was to evaluate how hard reproducing results is when licenses for software products are not available we worked with the open source software R. Implementations in different software products for complex methods such as GEEs and (G)LMMs may show slightly different results even when given the same inputs and with this we expected difficulties in reproducing exactly the same numbers for all papers using software other than R.

Table 3. Which software was used by the papers?.

	Software	Open Source	Source Code	Computing Environment
[9]	SAS	no	partly	SAS version
[10]	SPSS	no	no	SPSS version
[11]	no information (email contact states Stata)	no	no	no information
[12]	no information (email contact states Stata)	no	no	no information
[13]	SAS	no	no	SAS version
[14]	SAS	no	no	SAS version
[15]	SAS	no	no	SAS version
[16]	R	yes	upon request	Package version
[17]	SAS	no	no	SAS version
[18]	SPSS	no	no	SPSS version
[19]	MPlus	no	no	MPlus version

https://doi.org/10.1371/journal.pone.0251194.t003

Table 4. Were the results reproducible?.

	Method documentation	Contact Attempts	Author Responses	Models Computable	Same Interpretation	Classification of Failure
[9]	Missing Details	2	1	partly	no	Software differences
[10]	Missing Details	0	0	yes	yes	
[11]	yes	1	1	partly	yes	Software differences
[12]	Missing Details	1	1	yes	yes	
[13]	Missing Details	3	2	partly	no	Software differences
[14]	yes	1	0	no	no	Software differences, Model Description
[15]	Correlation Structure missing	1	1	yes	yes	
[16]	Correlation Structure missing	1	1	yes	yes	
[17]	Correlation Structure missing	3	1	yes	yes	
[18]		4	1	no		Data and Model description
[19]	yes	0	0	yes	yes	

https://doi.org/10.1371/journal.pone.0251194.t004

Is the source code available? Only one paper (paper [9]) provided source code. The source code provided was only a small part of the entire code needed to reproduce the results. Nevertheless it was a major help in obtaining the specifications of the models. For one paper we received the code through our email conversations [16]. For all other papers we had to rely on the methods and results sections of the papers. Often we resorted to reverse engineering the results as the methods sections were not sufficiently detailed.

Is the computing environment described (or delivered)? In most cases the authors provided information on the software used and the software version (9 out of 11). None of the papers described the operating system or provided a computing environment (e.g. Docker container).

Are we able to reproduce the data analysis? The results of this section are summarized in Table 4.

Are the methods used clearly documented in paper or supplementary material (e.g. analysis code)? Although all papers in question had methods sections, for most papers we were not able to extract all needed information to reproduce the results by ourselves. The most common issue was that papers did not provide enough detail about the methods used (e.g. model type was mentioned but no detailed model specifications, for details see Table 4). Since, in addition, no source code was provided (except for paper [9]), reproducing results was generally only possible by reverse engineering and/or contacting the authors. As most authors used licensed software which was not available to us, we could not determine if we would have reached the same results using default settings in the respective software. A clear documentation therefore requires enough detail to explicitly specify all necessary parameters for the model, even when using a different software.

Do we have to contact the authors in order to reproduce the analysis? How many e-mails are needed to reproduce the results? In all but two cases (papers [10, 19]) we contacted the authors to ask questions on how the results were generated (for four of them several emails were exchanged). All but one of the authors responded, which was to be expected as we had previously contacted them asking whether they would agree with us doing this project and only papers were chosen where authors responded positively. In most cases responses by authors were helpful.

Do we receive the same (or very similar) numbers in tables, figures and models? As the articles use different models and present their main results in terms of different statistics (model coefficients, F-statistics, correlation), the purely numerical deviation between our

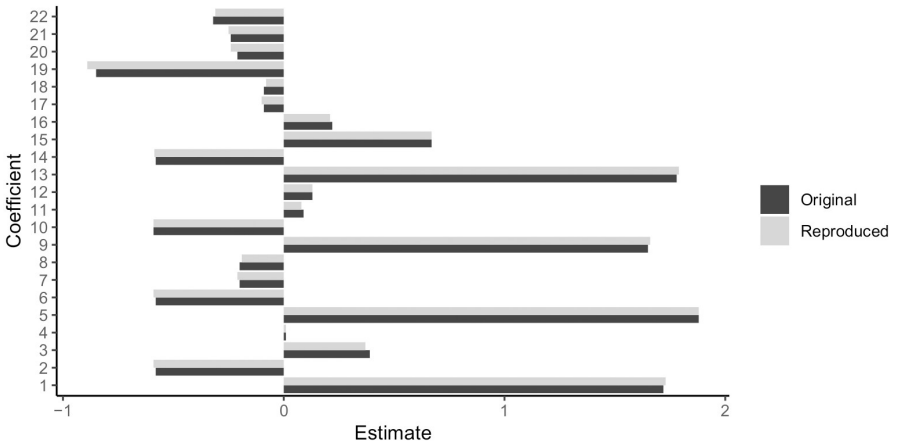


Fig 2. Original and reproduced model parameter estimates for the ewbGEE model of article [15]. In this article the differences in parameters do not lead to a different interpretation.

https://doi.org/10.1371/journal.pone.0251194.g002

results and the original results is not informative in isolation. Also, as we used different software implementations, some deviation was to be expected. Therefore, we define similar results as having the same implied interpretations, regarding sign and magnitude of effects. If the signs of the coefficients was the same and the ordering and magnitude of coefficients roughly the same, we regarded the results as successfully reproduced. We were able to fully reproduce 6 out of 11 articles (see also Table 4). Here differences were marginal and did not lead to a change of interpretations. An example (original and reproduced coefficients of article [15]) can be seen in Fig 2. For another two articles at least parts of the analysis could be reproduced (e.g. one out of two models used by the authors). For the 8 articles, that we found to be fully or partly reproducible, we were able to follow the data preprocessing and identify the most likely model specifications. Only three out of the 11 papers could not be reproduced at all, one because of implementation differences [13] and one due to problems preparing the data set used by the authors [18]. In [14] it was unclear how the data was originally analysed and without responses from the authors to our contact attempts via email we were not able to determine whether the different conclusions reached by our analysis are due to incorrect analysis on side of the authors or missing information.

Note that for some of the results, a considerable amount of time and effort needed to be invested to reverse engineer model settings. In the following we summarize the reproduction process for each paper individually, in order to give more insights about the specific problems and challenges that we encountered. (see also Table 4).

In [9] problems arose with the provided data set. The data description was found to be insufficient. Variable names in the data set differed from the ones in the code provided by the authors. We were able to resolve this problem based on feedback from the authors. When running the analysis using R and the R package PROreg [28], results differed from the original results due to details in the implementation and a different optimization procedure. The reproduced coefficients had the same sign as in the original study. However, differences in magnitude were large for some of the coefficients, likely due to differences in the optimization procedure. Given our definition, we were unable to reproduce the results. A second model fitted by the authors was not reproduced, due to convergence problems (model could not be fitted at all).

We were able to reproduce the results in [10] without contacting the authors. Some difficulty arose from the very sparse model description in the publication, such as, which variables were included as fixed or random effects. Also no source code was available. However within reasonable trial of different model specifications we obtained very similar results as in the original publication.

In [11] the number of observations differed between the publication and the provided data set. Upon request one of the authors provided a data set, that was almost identical to the one used in the study. The performed descriptive analysis and correlation analysis yielded the same results. A second difficulty arose, as the authors did not specify the correlation structure used in their model, but instead relied on the Stata routine to determine the best fitting correlation structure using the Quasi-Likelihood information criterion. If the correlation structure yielding the coefficients closest to the ones in [11] is used, the coefficients are almost identical. However, we also performed the aforementioned model search procedure in R but ended up with a different correlation structure as the best fitting. Using the correlation structure found best by our R implementation, would lead to a change in interpretation of the coefficients.

In [12] difficulties arose from different implementations in the software used. Also the model description was incomplete, which required us to try all possible combinations of variables to include. However, the correlation structure was well described and with feedback from the authors we were able to obtain the same results deviating only on the third decimal.

[13] used a cross-classified LMM, via the SAS "PROC mixed procedure". Reproduction in R was difficult, as no R package offered the exact same functionality. After trying several R implementations, we settled on the nlme R package [29]. The random effects were not specified in the publication. Also SAS code to shed light on this question was not available. Other questions regarding preprocessing and model specifications could be resolved through the feedback of the authors, but we did not receive the needed information on the random effects. As such we could not reproduce the results.

In [14] the data set used for modeling was not given as a file. Instead the authors provided links to the website where the data had been initially obtained from. We were not able to obtain the same data set given the sources and the description. This might be due to changes in the online sources. Still, differences in summary statistics were not substantial. We were unable to reproduce the same model due to unclear model specification. Our attempts led to some vastly different estimates. Possible reasons for failure are an insufficient model description or even incorrect analysis.

We were able to reproduce the results in [15] with only minor differences in the estimated coefficients. Feedback from the authors was required to find the correct correlation structure used in their GEE model, which was not explicitly stated in the paper.

The results in [16] were computationally reproducible. Despite minor differences in the coefficients we arrived at the same interpretations and differences were most likely due to different optimization procedures in the softwares used. The correlation structure was not stated in the article, but we were able to find the correct one using reverse engineering (grid search).

For the reproduction of [17] we had problems with data preprocessing. This was partly due to the unclear handling of missing values and due to details of the dimensionality reduction procedure used in preprocessing. The authors provided the final data set when we contacted them. The model specifications of the GEE used by the authors were not stated, but we were able to reproduce the exact same results as the authors by reverse engineering the correlation structure and link function. During this we found that using different model specifications or slightly different versions of the data set leads to substantially different results. Given the above definition this article was reproducible.

The results in [18] could not be reproduced. The (DNA) data was given in raw format as a collection of hundreds of individual files, without any provided code or step by step guide for preprocessing, making reproduction of the data set to be used in the statistical analysis impossible for us. Figures and Tables of the clinic data were reproducible.

The results in [19] were reproducible. All necessary model specifications for their GEE model and reasoning behind it were explicitly stated in the paper. The original analysis was carried out in M-plus, but reproduction in R gave almost identical results.

What are characteristics of papers which make reproducibility easy/possible or difficult/impossible? Based on the discussion of the individual papers we identified determinants of successes and failures. We found that the simpler the methods used in the paper the easier it was to reproduce the paper. Papers dealing with classical LMMs (papers [10, 14]) were reasonably easy to reproduce.

The data provided by the authors played a major role as well. If the clean data was provided, reproducing was much easier than for papers providing raw data (papers [14, 17, 18]), where preprocessing was still necessary. For one paper [18] getting and preparing the data was so complex that we gave up. Even after the authors provided us with an online tutorial on working with this type of data, we were far from understanding what needed to be done. If specialists (e.g. bioinformaticians) on working with this type of data had been involved, we might have had better chances.

We believe that with code provided—even if it is written using software we do not have access to—computational reproducibility is easier to obtain. It is hard to make this conclusion based on the 11 papers we worked with, because only one provided partial code and 1 provided code on request, but they also did not contradict our prior beliefs.

What are learnings from this study? What recommendations can we give future authors for describing their methods and reporting their results? Trying to reproduce 11 papers gave us a glimpse at how hard computational reproducibility is. We used papers published in an open access journal, which provided data and the authors were supportive of the project. We think it is fair to assume that these papers are among the most open projects available in academic literature at the moment. Nevertheless we were only able to reproduce the results without contacting the authors for two papers.

We not only recommend authors to provide data **and** code with their paper, but we suggest that this should be made a requirement from journals.

Further points

One paper published raw names of study participants, which we saw as unnecessary information and with that as an unreasonable breach of the participants. We informed the authors who updated the data on the journal website.

Discussion

In this study we aimed at reproducing the results from 11 PLOS ONE papers dealing with statistical methods for longitudinal data. We found that most authors use tables and figures as tools for presenting research results. Although all papers in question had data available for download, only one paper came with accompanied source code. From our point of view the lack of source code is the main barrier in reproducing results of the papers. For some papers we were still able to reproduce results by using a strategy of reverse engineering the results and by asking the authors. In an ideal situation, however, the information needed should not be hidden within the computers and minds of original authors, but should be shared as part of

the article (optimally in the form of a research compendium with paper, data, code, and metadata).

One of the authors initially contacted asked us to refrain from reproducing their paper on the grounds that students would not have the capabilities to do such complex analyses. We did not include the article in our study, but strongly disagree with this statement, especially since the students in question all have a strong statistics background and benefited from the guidance of researchers. Furthermore the students checked each other's works in an internal peer review. We would even go so far as to claim that a lot of other statistical work is less understood by the researcher and less thoroughly checked by peers before it is combined into a publication. Working as a big team gave us the option to conduct time intensive reverse engineering attempts of results, which small research teams or single researchers would potentially not have had.

We did not choose the papers randomly, but based on the set of potential papers given to us by PLOS ONE and then selected all papers meeting our criteria (see Fig 1). We can and should not draw conclusions from our findings on the 11 selected papers on the broader scientific landscape. Our work does, however, give us some insights on what researchers, reviewers, editors and publishers could focus on improving in the future: Publish code next to the data. To PLOS ONE we propose to include code in their open data policy.

Reproducing a scientific article is an important contribution to science and knowledge discovery. It increases trust in the research which is computationally reproducible and raises doubt in the research which is not.

Technical details

All results including detailed reports and code for each of the 11 papers are available in the GitLab repository https://gitlab.com/HeidiSeibold/reproducibility-study-plos-one. All files can also be accessed through the Open Science Framework (https://osf.io/xqknz). For all computations all relevant computational information (R and package versions, operating system) are given below the respective computations. The relevant information for this article itself is shown below.

- R version 4.0.3 (2020-10-10), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=de_DE.UTF-8`, `LC_COLLATE=en_US.UTF-8`, `LC_MONETARY=de_DE.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=de_DE.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=de_DE.UTF-8`, `LC_IDENTIFICATION=C`
- Running under: `Ubuntu 20.04.2 LTS`
- Matrix products: default
- BLAS: `/usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0`
- LAPACK: `/usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0`
- Base packages: base, datasets, graphics, grDevices, methods, stats, tools, utils
- Other packages: data.table 1.13.0, dplyr 1.0.2, ggplot2 3.3.3, googlesheets 0.3.0, kableExtra 1.3.1, knitr 1.32, plyr 1.8.6, rcrossref 1.1.0
- Loaded via a namespace (and not attached): cellranger 1.1.0, cli 2.4.0, codetools 0.2-18, colorspace 2.0-0, compiler 4.0.3, crayon 1.4.1, crul 1.1.0, curl 4.3, digest 0.6.27, DT 0.18, ellipsis 0.3.1, evaluate 0.14, fansi 0.4.2, farver 2.1.0, fastmap 1.1.0, generics 0.0.2, glue 1.4.2, grid

4.0.3, gtable 0.3.0, hms 0.5.3, htmltools 0.5.1.1, htmlwidgets 1.5.3, httpcode 0.3.0, httpuv 1.5.5, httr 1.4.2, jsonlite 1.7.2, labeling 0.4.2, later 1.1.0.1, lifecycle 1.0.0, magrittr 2.0.1, mime 0.10, miniUI 0.1.1.1, munsell 0.5.0, pillar 1.6.0, pkgconfig 2.0.3, promises 1.2.0.1, ps 1.6.0, purrr 0.3.4, R6 2.5.0, Rcpp 1.0.6, readr 1.4.0, reshape2 1.4.4, rlang 0.4.10, rmarkdown 2.7, rstudioapi 0.13, rvest 0.3.6, scales 1.1.1, shiny 1.6.0, stringi 1.5.3, stringr 1.4.0, tibble 3.1.1, tidyselect 1.1.0, utf8 1.2.1, vctrs 0.3.7, viridisLite 0.4.0, webshot 0.5.2, withr 2.4.2, xfun 0.22, xml2 1.3.2, xtables 1.8-4

Author Contributions

Conceptualization: Heidi Seibold.

Formal analysis: Severin Czerny, Siona Decke, Roman Dieterle, Thomas Eder, Steffen Fohr, Nico Hahn, Rabea Hartmann, Christoph Heindl, Philipp Kopper, Dario Lepke, Verena Loidl, Maximilian Mandl, Sarah Musiol, Jessica Peter, Alexander Piehler, Elio Rojas, Stefanie Schmid, Hannah Schmidt, Melissa Schmoll, Lennart Schneider, Xiao-Yin To, Viet Tran, Antje Völker, Moritz Wagner, Joshua Wagner, Maria Waize, Hannah Wecker, Rui Yang, Simone Zellner.

Investigation: Heidi Seibold.

Methodology: Heidi Seibold.

Project administration: Heidi Seibold, Malte Nalenz.

Software: Heidi Seibold.

Supervision: Heidi Seibold, Malte Nalenz.

Visualization: Heidi Seibold, Malte Nalenz.

Writing – original draft: Heidi Seibold, Malte Nalenz.

Writing – review & editing: Heidi Seibold, Malte Nalenz.

References

1. Stodden V, Seiler J, Ma Z. An Empirical Analysis of Journal Policy Effectiveness for Computational Reproducibility. Proceedings of the National Academy of Sciences. 2018; 115(11):2584–2589. https://doi.org/10.1073/pnas.1708290115 PMID: 29531050
2. Kirouac DC, Cicali B, Schmidt S. Reproducibility of Quantitative Systems Pharmacology Models: Current Challenges and Future Opportunities. CPT: Pharmacometrics & Systems Pharmacology. 2019. https://doi.org/10.1002/psp4.12390 PMID: 30697975
3. Hardwicke TE, Bohn M, MacDonald K, Hembacher E, Nuijten MB, Peloquin BN, et al. Analytic reproducibility in articles receiving open data badges at the journal Psychological Science: an observational study.
4. Obels P, Lakens D, Coles NA, Gottfried J, Green SA. Analysis of open data and computational reproducibility in registered reports in psychology. Advances in Methods and Practices in Psychological Science. 2020; 3(2):229–237. https://doi.org/10.1177/2515245920918872
5. Maassen E, van Assen MA, Nuijten MB, Olsson-Collentine A, Wicherts JM. Reproducibility of individual effect sizes in meta-analyses in psychology. PloS one. 2020; 15(5):e0233107. https://doi.org/10.1371/journal.pone.0233107 PMID: 32459806
6. Artner R, Verliefde T, Steegen S, Gomes S, Traets F, Tuerlinckx F, et al. The reproducibility of statistical results in psychological research: An investigation using unpublished raw data. Psychological Methods. 2020;. PMID: 33180514
7. Hoffmann S, Schönbrodt FD, Elsas R, Wilson R, Strasser U, Boulesteix AL. The multiplicity of analysis strategies jeopardizes replicability: lessons learned across disciplines; 2020. Available from: osf.io/preprints/metaarxiv/afb9p.

8. Baumgaertner B, Devezer B, Buzbas EO, Nardin LG. Openness and Reproducibility: Insights from a Model-Centric Approach; 2019.
9. Wagner BD, Sontag MK, Harris JK, Miller JI, Morrow L, Robertson CE, et al. Airway Microbial Community Turnover Differs by BPD Severity in Ventilated Preterm Infants. PLOS ONE. 2017; 12(1): e0170120. https://doi.org/10.1371/journal.pone.0170120 PMID: 28129336
10. Meda SA, Gueorguieva RV, Pittman B, Rosen RR, Aslanzadeh F, Tennen H, et al. Longitudinal Influence of Alcohol and Marijuana Use on Academic Performance in College Students. PLOS ONE. 2017; 12(3):e0172213. https://doi.org/10.1371/journal.pone.0172213 PMID: 28273162
11. Visaya MV, Sherwell D, Sartorius B, Cromieres F. Analysis of Binary Multivariate Longitudinal Data via 2-Dimensional Orbits: An Application to the Agincourt Health and Socio-Demographic Surveillance System in South Africa. PLOS ONE. 2015; 10(4):e0123812. https://doi.org/10.1371/journal.pone.0123812 PMID: 25919116
12. Vo LNQ, Vu TN, Nguyen HT, Truong TT, Khuu CM, Pham PQ, et al. Optimizing Community Screening for Tuberculosis: Spatial Analysis of Localized Case Finding from Door-to-Door Screening for TB in an Urban District of Ho Chi Minh City, Viet Nam. PLOS ONE. 2018; 13(12):e0209290. https://doi.org/10.1371/journal.pone.0209290 PMID: 30562401
13. Aerenhouts D, Clarys P, Taeymans J, Cauwenberg JV. Estimating Body Composition in Adolescent Sprint Athletes: Comparison of Different Methods in a 3 Years Longitudinal Design. PLOS ONE. 2015; 10(8):e0136788. https://doi.org/10.1371/journal.pone.0136788 PMID: 26317426
14. Tabatabai MA, Kengwoung-Keumo JJ, Oates GR, Guemmegne JT, Akinlawon A, Ekadi G, et al. Racial and Gender Disparities in Incidence of Lung and Bronchus Cancer in the United States: A Longitudinal Analysis. PLOS ONE. 2016; 11(9):e0162949. https://doi.org/10.1371/journal.pone.0162949 PMID: 27685944
15. Rawson KS, Dixon D, Nowotny P, Ricci WM, Binder EF, Rodebaugh TL, et al. Association of Functional Polymorphisms from Brain-Derived Neurotrophic Factor and Serotonin-Related Genes with Depressive Symptoms after a Medical Stressor in Older Adults. PLOS ONE. 2015; 10(3):e0120685. https://doi.org/10.1371/journal.pone.0120685 PMID: 25781924
16. Kawaguchi T, Desrochers A. A time-lagged effect of conspecific density on habitat selection by snowshoe hare. PLOS ONE. 2018; 13(1):e0190643. https://doi.org/10.1371/journal.pone.0190643 PMID: 29320564
17. Lemley KV, Bagnasco SM, Nast CC, Barisoni L, Conway CM, Hewitt SM, et al. Morphometry Predicts Early GFR Change in Primary Proteinuric Glomerulopathies: A Longitudinal Cohort Study Using Generalized Estimating Equations. PLOS ONE. 2016; 11(6):e0157148. https://doi.org/10.1371/journal.pone.0157148 PMID: 27285824
18. Carmody LA, Caverly LJ, Foster BK, Rogers MAM, Kalikin LM, Simon RH, et al. Fluctuations in Airway Bacterial Communities Associated with Clinical States and Disease Stages in Cystic Fibrosis. PLOS ONE. 2018; 13(3):e0194060. https://doi.org/10.1371/journal.pone.0194060 PMID: 29522532
19. Villalonga-Olives E, Kawachi I, Almansa J, von Steinbüchel N. Longitudinal Changes in Health Related Quality of Life in Children with Migrant Backgrounds. PLOS ONE. 2017; 12(2):e0170891. https://doi.org/10.1371/journal.pone.0170891 PMID: 28151986
20. Casals M, Girabent-Farrés M, Carrasco JL. Methodological Quality and Reporting of Generalized Linear Mixed Models in Clinical Medicine (2000–2012): A Systematic Review. PLoS ONE. 2014; 9(11): e112653. https://doi.org/10.1371/journal.pone.0112653 PMID: 25405342
21. LeBel EP, McCarthy RJ, Earp BD, Elson M, Vanpaemel W. A Unified Framework to Quantify the Credibility of Scientific Findings. Advances in Methods and Practices in Psychological Science. 2018; 1 (3):389–402. https://doi.org/10.1177/2515245918787489
22. R Core Team. R: A Language and Environment for Statistical Computing; 2020. Available from: https://www.R-project.org/.
23. Python Software Foundation. Python Software; 2020. Available from: http://www.python.org.
24. Boettiger C. An Introduction to Docker for Reproducible Research. SIGOPS Oper Syst Rev. 2015; 49 (1):71–79. https://doi.org/10.1145/2723872.2723882
25. Couture JL, Blake RE, McDonald G, Ward CL. A Funder-Imposed Data Publication Requirement Seldom Inspired Data Sharing. PLOS ONE. 2018; 13(7):e0199789. https://doi.org/10.1371/journal.pone.0199789 PMID: 29979709
26. Simmons JP, Nelson LD, Simonsohn U. False-Positive Psychology. Psychological Science. 2011; 22 (11):1359–1366. https://doi.org/10.1177/0956797611417632 PMID: 22006061
27. Muff S, Held L, Keller LF. Marginal or conditional regression models for correlated non-normal data? Methods in Ecology and Evolution. 2016; 7(12):1514–1524. https://doi.org/10.1111/2041-210X.12623

28. Najera J, Lee DJ, Arostegui I. PROreg: Patient Reported Outcomes Regression Analysis; 2017. Available from: https://CRAN.R-project.org/package=PROreg.

29. Pinheiro J, Bates D, DebRoy S, Sarkar D, R Core Team. nlme: Linear and Nonlinear Mixed Effects Models; 2020. Available from: https://CRAN.R-project.org/package=nlme.

Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12.07.11, §8, Abs. 2 Pkt. .5.)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

München, 28.07.2022	Malte Nalenz
Ort, Datum	Unterschrift Doktorand