Distributional Trees and Forests

Lisa Schlosser

Dissertation

in Mathematics

submitted to the Faculty of Mathematics, Computer Science and Physics
of the Leopold-Franzens-University Innsbruck

in partial fulfillment of the requirements for the degree of
Doctor of Philosophy (PhD)

Supervisors:
Univ.-Prof. Dr. Achim Zeileis
Univ.-Prof. Dr. Torsten Hothorn
Univ.-Prof. Dr. Dr. h.c. Michael Oberguggenberger

Innsbruck, January 2020
Abstract

Obtaining valuable information from given data requires the use of appropriate methods of analysis. For example, if a certain variable of interest is assumed to depend on a (set of) covariate(s), this relationship can be described by regression models based on a set of observations. Once a well fitting model is found not only analysis of dependencies but also predictions of the variable of interest can be performed.

Classical data models assume to know the underlying data generating mechanism up to a few unknown parameters which are estimated based on the given data. In contrast, regression trees and random forests follow an algorithm to fit a model in a purely data-driven way without requiring any model assumptions, i.e., the data generating process remains unknown. Over the last decade important progress has been made in the field of distributional modeling with flexible additive data models. However, until now these developments have not yet been fully integrated into the idea of regression trees and random forests. Therefore, the main goal of this dissertation is to provide the missing synthesis by introducing distributional trees and forests. On the one hand, this novel methodology allows for a full specification of a probability distribution in each node of the tree by modeling all distribution parameters, also accounting for possible censoring, truncation, tail behavior etc. This offers a wide range of information as not only expected values but for example also exceeding probabilities for a certain threshold or confidence intervals can be predicted. On the other hand, the tree-based models select covariates and their possible interactions automatically such that no model specifications are required in advance. Moreover, smooth effects of the explanatory variables can be approximated by an ensemble of trees combined to a more stable and regularized forest model, while non-linear and non-additive effects are detected by the splitting algorithm used to build a tree.

The algorithms employed for the distributional trees and forests are so-called unbiased recursive partitioning techniques. This refers to the property that the key step of selecting a split variable is based on a statistical test. In order to compare three commonly applied testing strategies a unifying testing framework is provided which allows for combining the building blocks of the considered strategies as required such that their effects can be investigated separately. Among other results this analysis shows that considering all model parameters for splitting as employed in distributional trees and forests is advantageous over including only the mean.

In practice, distributional trees and forests provide a flexible and easy-to-use alternative to already existing modeling approaches, especially if no prior information on possible effects is available and/or if the underlying distribution is not expected to be normal. This advantage is demonstrated by applying distributional trees and forests to precipitation data which is left censored at zero and to wind direction data which follows a circular distribution. The corresponding implementation of the proposed methodology has been developed in the statistics system \textsc{R} and is provided in an open source software package.
I would like to express my sincere gratitude to all those who supported me throughout my studies as this work would not have been possible without them.

First of all, I want to thank my supervisor Achim Zeileis for his guidance and expertise all through the way – from introducing me to the research field in an inspiring and motivating manner to providing invaluable help in each step to the completion of this thesis – and for taking the time to answer all my questions.

In addition, I would like to thank my co-supervisors, Torsten Hothorn for his always helpful advice and feedback and Michael Oberguggenberger for supporting this dissertation project.

I am also very grateful to my colleagues Heidi Seibold, Moritz Lang, and Reto Stauffer for the truly enjoyable and productive collaboration.

Furthermore, I would like to thank the whole team of the Department of Statistics for all their help, for providing such a comfortable and friendly working environment, and for making me feel welcome from the first day on. A special thank you also goes to Daniel Gstrein for being a great office mate adding many cheerful moments to everyday working life.

Moreover, I gratefully appreciate a PhD scholarship granted from the Vizerektorat für Forschung of the University of Innsbruck.

Finally, I would like to thank my family and friends for their continuous support, especially my parents and my brother, who have always been there for me.

Innsbruck, January 2020

Lisa Schlosser
Abstract

Acknowledgements

Chapter 1. Introduction

1.1 Introduction to the research field

1.1.1 Parametric vs. tree-based modeling

1.1.2 Tree algorithms incorporating parametric models

1.2 Tree-based distributional modeling

1.2.1 Distributional trees and forests

1.2.2 Adaptive local likelihood estimation

1.3 Application: Probabilistic weather forecasting

1.3.1 Precipitation

1.3.2 Wind direction

1.4 Software

1.5 Dissertation outline

1.6 Summary and outlook

Chapter 2. Paper I: Distributional Regression Forests

2.1 Introduction

2.2 Methodology

2.2.1 Distributional fit

2.2.2 Distributional tree

2.2.3 Distributional forest

2.3 Probabilistic precipitation forecasting in complex terrain

2.3.1 Data

2.3.2 Models and evaluation

2.3.3 Application for one station

2.3.4 Application for all stations

2.4 Discussion

2.5 Appendix: Tree algorithm
### Chapter 3. Paper II: The Power of Unbiased Recursive Partitioning

3.1 Introduction .......................................................... 32
3.2 Unbiased recursive partitioning .................................... 33
   3.2.1 Generic algorithm ............................................. 33
   3.2.2 CTree ......................................................... 34
   3.2.3 MOB .......................................................... 35
   3.2.4 GUIDE ....................................................... 35
   3.2.5 Pruning ....................................................... 36
3.3 Unifying framework .................................................. 36
   3.3.1 Building blocks of testing strategies ....................... 37
   3.3.2 Linear model tree .......................................... 38
3.4 Simulation setting and evaluation ................................. 39
   3.4.1 Data generating process ................................... 39
   3.4.2 Evaluation ................................................... 41
3.5 Results ............................................................... 42
   3.5.1 “Stump” scenario: Residuals vs. full model scores .......... 42
   3.5.2 “Stump” scenario: Full factorial analysis of building blocks ......................... 43
   3.5.3 “Tree” scenario: Pre-pruning vs. post-pruning .................. 45
3.6 Discussion ............................................................. 46
3.7 Appendix ............................................................. 48
   3.7.1 Test statistics ................................................ 48
   3.7.2 Combinations of building blocks applied .................... 49
   3.7.3 Significance level ............................................ 49
   3.7.4 Continuous “stump” scenario ................................ 49
   3.7.5 Results for increasing effect size ............................ 51

### Chapter 4. Paper IIIa: Distributional Trees for Circular Data

4.1 Motivation .............................................................. 55
4.2 Methodology .......................................................... 56
4.3 Application ............................................................ 56
4.4 Discussion and outlook .............................................. 58
   4.4.1 Ensembles and random forests ............................... 58
   4.4.2 Splits in circular covariates .................................. 58
CHAPTER 1
INTRODUCTION

This thesis introduces a novel methodology in the field of statistical regression modeling by embedding distributional parametric models into the framework of regression trees and random forests. An overview of the resulting distributional trees and forests is presented in Section 1.2 after an introduction to the research field in Section 1.1. Two application studies and the therein employed software are discussed in Sections 1.3 and 1.4, followed by an outline of the dissertation in Section 1.5 and a summary with an outlook on possible future work in Section 1.6.

1.1 Introduction to the research field

The main goal of statistical modeling is to find an approximation of reality based on a set of observations. It is assumed that the current state of a system can be described by some measurements $X$. After a usually unknown process has taken place in this system, its state or a certain quantity of interest might have been affected. To investigate this, measurements $Y$ are collected yielding the basic initial situation in statistical regression modeling as illustrated in Figure 1.1 where an input $X$ is handed over to a usually unknown process which returns an output $Y$.

Building a model that describes the unknown process is mainly motivated by two possible applications: (1) investigating the dependency of $Y$ on $X$ with possible interactions and (2) making predictions for $Y$ based on a new given $X$. For example, $X$ can consist of meteorological quantities describing today’s weather such as the amount of precipitation, air pressure, temperature or wind speed, and $Y$ can be the daily precipitation sum of the following day. The here considered atmospheric process is not completely unknown, hence, available information on changes in the atmosphere can also be included in $X$, e.g., in the form of numerical weather predictions obtained from physical models. In this application, a well-fitting statistical model can not only be used to predict the amount of precipitation that is to be expected for the next day but also to provide information on how the daily precipitation sum is affected by the state of weather of the foregoing day.

As explained by Breiman (2001b) there are two main approaches of statistical modeling: data modeling and algorithmic modeling. In data modeling a fixed type of model is selected in

\[ X \rightarrow \text{process} \rightarrow Y \]

**Figure 1.1.** Basic initial situation in statistical regression modeling: An input $X$ is handed over to a usually unknown process which returns an output $Y$. 
advance and the given data is used to estimate its parameters. Thus, mathematical equations are applied and model assumptions have to be made in order to approximate the true data generating process up to a few unknown influences. While this clear structure allows for straightforward analysis, the risk of incorrect model assumptions has to be taken into consideration in every application such that expert knowledge is often required to set up a model. On the contrary, in algorithmic modeling no assumptions have to be made in advance. Therefore, this data-driven approach provides a more flexible way of fitting a model, however, the unknown process remains a ’black box’ as the goal is not to rebuild it (as in data modeling) but to imitate it. Thus, each of these two concepts has advantages and disadvantages and both of them have been investigated and developed further, however, mostly separated from each other. While tree-based regression models have gained more and more attention in the field of algorithmic modeling, parametric data models provide a commonly used toolbox that is suitable for a wide range of situations.

1.1.1 Parametric vs. tree-based modeling

In a parametric regression model \( M_\theta(Y; X) \) with response \( Y \) and regressor(s) \( X \) the (vector of) model parameter(s) \( \theta \) is estimated based on observations of \( Y \) and \( X \) such that a goodness-of-fit measurement of the model is optimized. For example, a common method to estimate \( \theta \) is the ordinary least squares method (OLS) which minimizes the sum of squared errors made in the prediction of the response variable. Many different parametric regression models have been developed, starting from a well known basic example, the linear model (LM), which is built on the assumption that there is a linear relationship between the conditional mean of the response \( Y \) and the regressor(s) \( X \). Extending this approach, in generalized linear models (GLMs, Nelder and Wedderburn, 1972) the idea is to provide more flexibility by encompassing an additional link function for the conditional mean. Generalized additive models (GAMs, Hastie and Tibshirani, 1986) also allow for smooth nonlinear effects in the regressors. In order to fully specify the underlying distribution of the response variable \( Y \), generalized additive models for location, scale, and shape (GAMLSS, Rigby and Stasinopoulos, 2005) adopt a probabilistic modeling approach. In GAMLSS, each parameter of a statistical distribution can be modeled explicitly by an additive function of the covariates comprising linear and/or smooth nonlinear terms. In that way, a fully specified distribution of the response variable enables a much broader range of statistical inference as not only the expected value but also confidence intervals or certain thresholds can be estimated. However, the additive structure of the model remains and non-additive effects or interactions of regressor variables have to be prespecified and are harder to detect. This challenge is one of the core ideas of algorithmic trees and forests: to detect nonlinear and interaction effects “automatically”.

In a regression tree (such as built by the CART algorithm; Breiman, Friedman, Olshen, and Stone, 1984) the data set is split recursively into more homogeneous subgroups based on information provided by the regressor variables. The resulting tree structure can capture shifts and approximate nonlinear functions. Furthermore, regressors and their possible interactions are selected automatically. By building an ensemble of trees and combining them, a forest model can be obtained. Random forests (Breiman, 2001a) average the predictions of the trees that are each built on a slightly different data set which can either be a subsample or a bootstrap of the original data. Moreover, in each node only a random subset of the covariates is considered as possible split variables in order to reduce the correlation among the trees. The resulting forest model stabilizes the recursive partitions from individual trees and enables a better approximation of smooth functions.

Hence, parametric modeling and algorithmic tree-based modeling both provide a well working methodology that is applied in many different subject areas. This has risen the interest in combining them in a unifying framework in order to take advantage of their respective strengths.
1.2 Tree-based distributional modeling

1.1.2 Tree algorithms incorporating parametric models

Fitting a global parametric model to a (large) data set can often be very challenging, particularly in case of strongly fluctuating data and/or a high number of regressor variables. These possible difficulties can be avoided by separating the data into more homogeneous subgroups, e.g., by building a tree, and fitting a local parametric model in each resulting subgroup. In the past this procedure that combines parametric data models and algorithmic tree models has gained more and more interest since it allows for a more flexible and clear structure.

Algorithms such as GUIDE (Loh, 2002), CRUISE (Kim and Loh, 2001) and LOTUS (Chan and Loh, 2004) were among the first to provide trees with parametric models in each terminal node. The MOB algorithm (Zeileis, Hothorn, and Hornik, 2008), an algorithm for model-based recursive partitioning, extended this idea by fitting parametric models to each node and using the gained information of the fitted model to create further splits in inner nodes. Using this algorithm as a framework to build trees, LMs and GLMs can be fitted in the nodes of the trees. However, up to now trees which only model the mean of the response variable in its nodes are still more common and widely used and until now the above listed development in the field of distributional modeling has not yet been fully integrated in the idea of algorithmic tree and forest models. Therefore, the main goal of this thesis is to provide the missing synthesis by introducing distributional trees and forests.

1.2 Tree-based distributional modeling

With the aim of establishing a probabilistic approach to tree-based modeling, the flexible framework provided by GAMLSS is incorporated in the idea of regression trees and random forests. In particular, a complete probability distribution is specified in each node of a tree in order to capture location, scale, and shape as well as tail behavior, censoring, truncation, etc. The employed tree structure detects nonlinear and non-additive effects and selects covariates and possible interactions automatically such that no model assumptions are required in advance. Combining an ensemble of trees, the resulting forest can stabilize the model and enable a better approximation of smooth effects. Based on these ideas, distributional trees and forests are introduced in Schlosser, Hothorn, Stauffer, and Zeileis (2019b) as reprinted in Chapter 2. An overview of the presented novel methodology is given in the following section.

1.2.1 Distributional trees and forests

Distributional trees are built by employing an unbiased recursive partitioning algorithm, namely the CTree (Hothorn, Hornik, and Zeileis, 2006b) or the MOB (Zeileis et al., 2008) algorithm, and fitting a distributional model in each node of the tree. The latter requires the selection of a distribution family $D$ in advance such that the task of fitting a full probabilistic model $D(Y; \theta)$ to a response variable $Y$ turns into the task of estimating the corresponding distribution parameter vector $\theta \in \Theta$ with $\theta = (\theta_1, \ldots, \theta_k)$, $k \in \mathbb{N}$. Starting at the root node of the tree, this parameter estimation can be done by maximizing the log-likelihood function $l$ over the complete set of observations. Once a model is obtained, its goodness of fit can be assessed by calculating model scores, i.e., by evaluating the derivative of the log-likelihood function $\frac{\partial l}{\partial \theta}$ for the estimated parameter vector $\hat{\theta}$ and the observations $\{y_i\}_{i=1}^n, n \in \mathbb{N}$. In that way, for each pair of observation and parameter an indicator of discrepancy is obtained which is then used in the tree-building algorithm for the crucial decision on how and where to split the data. More precisely, statistical tests are employed to investigate associations between the model scores and each possible split variable $Z_m \in \{Z_1, \ldots, Z_m\}, m \in \mathbb{N}$. After selecting the split variable showing the highest association in a first step, the breakpoint within the selected variable that
leads to the highest improvement in the model fit is chosen in a second step. While searching for the best breakpoint over all available split variables in one single step would tend to be biased towards variables with a high number of possible split points, performing these two steps separately allows for an unbiased variable selection. After splitting the data, the same procedure is repeated in each of the resulting subgroups until a prespecified stopping criterion is met. This can for example be a significance level for the statistical test applied for the split variable selection or a minimal number of observations in the subgroups.

Summed up, the steps for building a distributional tree are:

1. Fit a global distributional model $D(Y; \theta)$ to the whole data set by estimating $\hat{\theta}$ via maximum likelihood: $\hat{\theta} = \arg\max_{\theta \in \Theta} \sum_{i=1}^{n} \ell(\theta; y_i)$
2. Test for associations/instabilities of the scores $\frac{\partial \ell}{\partial \theta}(\hat{\theta}; y_i)$ and each covariate $Z_l$.
3. Split along the covariate $Z$ with the strongest association or instability and at breakpoint $p$ leading to the highest improvement in log-likelihood.
4. Repeat steps 1–3 recursively until some stopping criterion is met, yielding $B$ subgroups $\{B_b\}_{b=1,...,B}$ with $B \in \mathbb{N}$.

Depending on the applied tree-building algorithm different testing strategies can be employed for the selection of a split variable in step 2. The MOB algorithm was explicitly designed for a model-based approach, which is also provided by an adaption of the CTree algorithm. In particular, in distributional trees model scores are included in the parameter instability tests (MOB) or the conditional independence tests (CTree) allowing for the detection of splits in all available model parameters. On the contrary, the GUIDE algorithm (Loh, 2002) only considers residuals in the therein employed classical categorical association tests. These three popular testing strategies are compared in Schlosser, Hothorn, and Zeileis (2019d) – see Chapter 3 of this thesis – in order to assess their respective advantages and disadvantages. By embedding them into a common inference framework encompassing parametric model trees, the different building blocks of the considered strategies can be combined as required which allows for assessing their effects separately. In particular, the focus is on investigating how binning of possible split variables, considering only residuals or full model scores for splitting, and dichotomizing the considered discrepancy measure affects the power of the algorithms to select the appropriate covariates for splitting. The results show that specifically the goodness-of-fit measure is crucial for the power of the procedures, with model scores without dichotomization performing best in many scenarios.

The advantage of considering all model parameters for splitting by including full model scores in the testing strategy is also demonstrated in Figure 1.2. The right panel illustrates a distributional tree fitted to the learning data depicted in the two-dimensional plot in the left panel. This data set consists of one covariate $Z$ and a response variable $Y$ which follows a normal distribution where the mean $\mu$ and the variance $\sigma^2$ both depend on $Z$, i.e., $(Y|Z = z) \sim \mathcal{N}(\mu(z), \sigma^2(z))$. The inner nodes of the tree indicate the selected split variable (here $Z$ is the only possible split variable) together with the corresponding $p$-value. The respective split point is displayed along the lines leading to the resulting daughter nodes. Looking at the obtained tree it can be observed that the first split is performed in the scale parameter $\sigma$ followed by a split in the location parameter $\mu$ in node 3. Hence, a tree algorithm including only the mean in the splitting strategy might miss the first split yielding a different model, probably without any distinction between nodes 2 and 5 as they show the same location parameter $\mu$ but differ in the scale parameter $\sigma$.

Moreover, the fitted distributional tree does not only provide predictions for the expected mean but specifies a full probabilistic model in each node, and hence also for each possibly
new observation, as illustrated by the corresponding estimated density function in the terminal nodes. This allows for modeling also non-normal distributions and for much broader inference, e.g., in addition to predicting a mean value one can also obtain confidence intervals or estimate the probability that a certain threshold is exceeded. For example in probabilistic precipitation forecasting, not only the expected amount of precipitation can be estimated but also the probability that it will rain at all or the amount of precipitation that will not be exceeded by a probability of 90 percent.

In this example the underlying data set provides only one covariate which simplifies the task of finding the best split to finding solely the best split point, if a split is performed at all. However, in a more general case with more than one possible split variable, the tree structure automatically selects covariates and their possible interactions such that no model specifications are required in advance.

As also shown in Figure 1.2, non-additive effects of explanatory variables can be detected by the splitting algorithm used to build the tree. In order to also enable a proper approximation of smooth effects, the idea of distributional trees is extended to the concept of random forests yielding distributional forests which combine an ensemble of distributional trees. In particular, for predictions from a distributional forest a set of weights, which serve as a measure of similarity between a (possibly new) observation and each learning observation, is obtained from the ensemble of trees. By including these weights in the parameter estimation process for a (possibly new) observation the log-likelihood contribution of each learning observation is determined. A more precise description of the employed weights and the prediction procedure is provided by Schlosser et al. (2019b), as reprinted in Chapter 2, and summarized in the following section.
1.2.2 Adaptive local likelihood estimation

Alternatively, distributional trees and forests can also be regarded as adaptive local likelihood models. This representation allows for a simple and general description of how predictions are performed by both tree-based models.

For a global likelihood model with learning data \( \{y_i\}_{i=1,...,n} \), \( n \in \mathbb{N} \), the parameter estimator is defined by

\[
\hat{\theta} = \arg\max_{\theta \in \Theta} \sum_{i=1}^{n} \ell(\theta; y_i).
\] (1.1)

In that way, one fixed estimated parameter vector \( \hat{\theta} \) is obtained for the full data set. However, only the response is employed in the fitting process. One way to also incorporate covariates \( \{z_i = (z_{i1},...,z_{im})\}_{i=1,...,n} \), \( n \in \mathbb{N} \), \( m \in \mathbb{N} \), is to include weights which depend on the covariates. In particular, for a learning data set \( \{(y_i, z_i)\}_{i=1,...,n} \), observation-specific weights can determine the influence of the corresponding observation in the estimation process by defining

\[
\hat{\theta}(z) = \arg\max_{\theta \in \Theta} \sum_{i=1}^{n} w_i(z) \cdot \ell(\theta; y_i)
\] (1.2)

which turns the model into an adaptive local likelihood model yielding an individual parameter estimate \( \hat{\theta}(z) \) for each (possible new) observation \( z = (z_1,\ldots,z_m) \).

For a basic global model all weights are set to 1, i.e., \( w_{i}^{\text{base}}(z) = 1 \). In a tree model with \( B \) terminal nodes \( \{B_1,\ldots,B_B\} \) the weight for the \( i \)-th learning observation is 1 if the (possibly new) observation \( z \) and the \( i \)-th learning observation are assigned to the same terminal node, otherwise 0:

\[
w_{i}^{\text{tree}}(z) = \sum_{b=1}^{B} 1((z_i \in B_b) \land (z \in B_b))\] (1.3)

Therefore, only learning observations of the same terminal node are considered for the estimation of the distribution parameters in Equation 1.2. This idea is extended to forest models by incorporating averaged “nearest neighbor weights” (Lin and Jeon, 2006; Hothorn and Zeileis, 2017). These are based on the number of trees in which the (possibly new) observation is assigned to the same terminal node:

\[
w_{i}^{\text{forest}}(z) = \frac{1}{T} \sum_{t=1}^{T} \sum_{b=1}^{B_t} 1((z_i \in B_t^b) \land (z \in B_t^b)) \frac{1}{|B_t^b|}\] (1.4)

for an ensemble of \( T \) trees with \( B_t \) terminal nodes in the \( t \)-th tree and \( |B_t^b| \) denoting the number of observations in the \( b \)-th subgroup of the \( t \)-th tree.

Hence, a fully specified distribution can be predicted for a possibly new observation \( z \) by obtaining the observation-specific weights \( w(z) \) from the distributional tree or forest fitted to the learning data and plugging them in the corresponding parameter estimator to calculate the estimated distribution parameter vector \( \hat{\theta}(z) \).

1.3 Application: Probabilistic weather forecasting

1.3.1 Precipitation

To illustrate the strengths of distributional trees and particularly of distributional forests in practice, they are applied to probabilistic precipitation forecasts based on a large number of
1.3 Application: Probabilistic weather forecasting

Numerical weather prediction quantities at various sites in the mountainous region of Tyrol. This case study is presented in Schlosser et al. (2019b) as reprinted in Chapter 2.

In particular, distributional forests are evaluated at 95 weather stations spread all over Tyrol together with reference models that are commonly applied in the field of weather forecasting: an ensemble model output statistics (EMOS, Gneiting, Raftery, Westveld III, and Goldman, 2005) approach and GAMLSS models specified based on meteorological knowledge/experience or on a computationally more demanding boosting approach. As precipitation sums cannot be negative, a Gaussian distribution left-censored at zero is employed such that the zero-censored point mass describes the probability of observing no precipitation on a given day. Additional evaluations based on two other distributional assumptions (a logistic distribution, left-censored at zero, and a two-part hurdle model) are provided in the supplementary material of the paper.

In this setting, for the majority of stations distributional forests perform at least similar or even better than the reference models. Thus, it can be concluded from this case study that distributional forests provide a good alternative approach to already well working modeling strategies in the field of probabilistic precipitation forecasting.

1.3.2 Wind direction

Wind directions are a popular example of circular data and one of the most important weather variables in meteorology. Hence, predictions of wind directions are of high interest, e.g., for airports, as they play an important role in many decision making processes for coordinating flights. But even though circular data can also be found in many other subject areas such as hourly crime rate in socio-economics or animal movement direction or gene-structure in biology, the methodology for distributional modeling of a circular response variables is rather limited.

Most of the already existing approaches to modeling circular data are built on additive regression models, such as introduced by Gould (1969) and further extended by Fisher and Lee (1992), Jammalamadaka and Sengupta (2001), and Mulder and Klugkist (2017) among others. However, apart from potential difficulties in the optimization procedure incorporated in these models, the interpretation of the underlying additive effects is often challenging as well because the link function is highly nonlinear and the representation of smooth transitions on the unit circle is not straightforward. Therefore, a distributional tree-based alternative employing the von Mises distribution is introduced in Lang, Schlosser, Hothorn, Mayr, Stauffer, and Zeileis (2020) as reprinted in Chapter 5. The resulting circular regression trees and forests are applied on shortterm probabilistic wind direction forecasting at the airports of Innsbruck and Vienna. In this case study various meteorological quantities measured at the airports of Vienna and Innsbruck and surrounding weather stations are used as covariates to predict hourly wind directions at the airports.

While first results for the airport of Innsbruck (as discussed in Schlosser, Lang, Hothorn, Mayr, Stauffer, and Zeileis, 2019e, which can be found in Chapter 4) show that circular regression trees already provide valuable forecasts, employing circular regression forests can improve them by regularizing and stabilizing the model. This is confirmed by the results obtained from the more extensive evaluations in this case study (as presented in Lang et al., 2020, which can be found in Chapter 5) where circular regression trees and forests are benchmarked against three alternative approaches to probabilistic forecasting of wind directions. Overall, the forest model performs best in all four considered settings: 1-hour and 3-hour forecasts for the airports of Innsbruck and Vienna. Therefore, this application shows that circular regression trees and forests provide reliable forecasts and offer a very flexible and easy-to-use methodology for probabilistic modeling of a circular response variable.
1.4 Software

As part of this dissertation the above described methodology of distributional trees and forests has been implemented in the statistics software R and is provided in the new open source R package disttree which contains the following three basic functions:

- **distfit** fits a distributional model for a specified distribution family via maximum likelihood and returns information on the goodness of fit in the form of model scores.

- **disttree** builds a distributional tree by employing tree-building functions provided by the partykit package (Hothorn and Zeileis, 2015) and using distfit in each node. Next to specifying a distribution family the user can also choose different testing strategies for the selection of split variables and split points providing a high level of flexibility. For example, the approaches applied in the MOB algorithm or the CTree algorithm can be chosen.

- **distforest** builds a distributional forest by combining an ensemble of distributional trees which are constructed by employing the function disttree.

For the specific application of distributional trees and forests to circular data an additional R package circtree has been developed. There, the basic functions of the package disttree are applied accounting for the circularity of data by employing the von Mises distribution and providing additional user options such as switching between different circular intervals.

Both packages are already freely available on R-Forge at https://R-Forge.R-project.org/projects/partykit/, together with the R package partykit which they are based on.

1.5 Dissertation outline

The main part of the thesis comprises the three papers reprinted in the Chapters 2, 3, 4, and 5:

- The first paper (Schlosser et al., 2019b) in Chapter 2 introduces distributional trees and forests together with an application to probabilistic precipitation forecasts at various weather stations in Tyrol. The paper has been accepted for publication at one of the leading applied statistics journals, The Annals of Applied Statistics, and various stages of the paper have been presented at national and international conferences (useR! 2017, CMStatistics 2017, JSM 2018, Statistische Woche 2018).

- The second paper (Schlosser et al., 2019d) in Chapter 3 focuses on different testing strategies employed in tree-building algorithms for the selection of split variables. By providing a unifying testing framework the building blocks of the considered strategies can be investigated separately, showing their respective advantages and disadvantages. The paper is currently under review in Advances in Data Analysis and Classification (and also available on arXiv) and has been presented at the international DAGStat 2019 conference.

- The third paper develops distributional trees and random forests for circular responses, using probabilistic wind direction forecasts as the application case. It has lead to preliminary results being published in a conference proceedings contribution (Schlosser et al., 2019e) for the International Workshop on Statistical Modelling (IWSM 2019), reprinted in Chapter 4. The extension to circular random forests is introduced in the full paper (Lang et al., 2020) in Chapter 5, submitted to the Journal of the Royal Statistical Society C and available on arXiv. In addition to IWSM 2019 the work has been presented at CMStatistics 2019.

In the first and second paper Lisa Schlosser was the leading author. In the third paper the two leading authors Moritz N. Lang and Lisa Schlosser contributed in equal parts.
1.6 Summary and outlook

The novel methodology of distributional trees and forests incorporates the development of distributional parametric models into the framework of regression trees and random forests. Combining these two modeling approaches allows to benefit from their respective strengths: While the tree structure detects nonlinear effects and selects covariates and their possible interactions automatically, the fitted distributional models offer a wide range of information as a fully specified distribution is provided in each node. Additionally, this probabilistic approach enables the detection of splits in all distribution parameters as full model scores are included in the testing strategies employed in the tree-building algorithm. Moreover, a distributional forest stabilizes the model such that also smooth effects can be modeled.

The presented application studies show that distributional trees and forests provide reliable alternatives to already existing commonly used modeling approaches, especially if it can not be assumed that the response variable follows a normal distribution such as in precipitation data (left-censored at zero) and wind direction data (circular). Furthermore, for complex data sets with a high number of covariates and no given information about them, the tree-based methods facilitate setting up a model as no specifications or model assumptions are required.

A possible extension of the introduced methodology could be the incorporation of different models in the nodes of a tree, such as for example in an extension of (generalised) linear model trees presented by Seibold, Hothorn, and Zeileis (2019). In particular, the covariates could be considered not only as split variables but also as regressors in the models fitted in each node of a tree. Staying in the framework of GAMLSS, each distribution parameter could be regressed on available covariates in parameter-specific additive models. While this would allow for capturing smooth effects also within nodes, it has to be considered that this would require a specification of the additive models. For the case of probabilistic weather forecasting as in the presented applications, a standard model output statistics (MOS) approach could be considered to postprocess output from numerical weather prediction (NWP) models also within the nodes of a tree.
Distributional Regression Forests for Probabilistic Precipitation Forecasting in Complex Terrain


Lisa Schlosser*, Torsten Hothorn**, Reto Stauffer*, Achim Zeileis*

*University of Innsbruck,
Department of Statistics, Universitätsstr. 15, A-6020 Innsbruck, Austria
**University of Zurich,
Epidemiology, Biostatistics and Prevention Institute, Hirschengraben 84, CH-8001 Zürich, Switzerland

Abstract To obtain a probabilistic model for a dependent variable based on some set of explanatory variables, a distributional approach is often adopted where the parameters of the distribution are linked to regressors. In many classical models this only captures the location of the distribution but over the last decade there has been increasing interest in distributional regression approaches modeling all parameters including location, scale, and shape. Notably, so-called non-homogeneous Gaussian regression (NGR) models both mean and variance of a Gaussian response and is particularly popular in weather forecasting. Moreover, generalized additive models for location, scale, and shape (GAMLSS) provide a framework where each distribution parameter is modeled separately capturing smooth linear or nonlinear effects. However, when variable selection is required and/or there are non-smooth dependencies or interactions (especially unknown or of high-order), it is challenging to establish a good GAMLSS. A natural alternative in these situations would be the application of regression trees or random forests but, so far, no general distributional framework is available for these. Therefore, a framework for distributional regression trees and forests is proposed that blends regression trees and random forests with classical distributions from the GAMLSS framework as well as their censored or truncated counterparts. To illustrate these novel approaches in practice, they are employed to obtain probabilistic precipitation forecasts at numerous sites in a mountainous region (Tyrol, Austria) based on a large number of numerical weather prediction quantities. It is shown that the novel distributional regression forests automatically select variables and interactions, performing on par or often even better than GAMLSS specified either through prior meteorological knowledge or a computationally more demanding boosting approach.

Keywords: parametric models, regression trees, random forests, recursive partitioning, probabilistic forecasting, GAMLSS.
2.1 Introduction

In regression analysis a wide range of models has been developed to describe the relationship between a response variable and a set of covariates. The classical model is the linear model (LM) where the conditional mean of the response is modeled through a linear function of the covariates (see the left panel of Figure 2.1 for a schematic illustration). Over the last decades this has been extended in various directions including:

- **Generalized linear models** (GLMs, Nelder and Wedderburn, 1972) encompassing an additional nonlinear link function for the conditional mean.

- **Generalized additive models** (GAMs, Hastie and Tibshirani, 1986) allowing for smooth nonlinear effects in the covariates (Figure 2.1, middle).

- **Generalized additive models for location, scale, and shape** (GAMLSS, Rigby and Stasinopoulos, 2005) adopting a probabilistic modeling approach. In GAMLSS, each parameter of a statistical distribution can depend on an additive predictor of the covariates comprising linear and/or smooth nonlinear terms (Figure 2.1, right).

Thus, the above-mentioned models provide a broad toolbox for capturing different aspects of the response (mean only vs. full distribution) and different types of dependencies on the covariates (linear vs. nonlinear additive terms).

While in many applications conditional mean regression models have been receiving the most attention, there has been a paradigm shift over the last decade towards distributional regression models. An important reason for this is that in many fields forecasts of the mean are not the only (or not even the main) concern but instead there is an increasing interest in probabilistic forecasts. Quantities of interest typically include exceedance probabilities for certain thresholds of the response or quantiles of the response distribution. Specifically, consider weather forecasting where there is less interest in the mean amount of precipitation on the next day. Instead, the probability of rain vs. no rain is typically more relevant or, in some situations, a prediction interval of expected precipitation (say from the expected 10% to 90% quantiles). Similar considerations apply for other meteorological quantities and hence attention in the weather forecasting literature has been shifting from classical linear deterministic models (Glahn and Lowry, 1972) towards probabilistic models such as the non-homogeneous Gaussian regression (NGR) of Gneiting et al. (2005). The NGR typically describes the mean of some meteorological response variable through the average of the corresponding quantity from an ensemble of physically-based numerical weather predictions (NWPs). Similarly, the variance of the response is captured through the variance of the ensemble of NWPs. Thus, the NGR considers both the mean as well as the uncertainty of the ensemble predictions to obtain probabilistic forecasts calibrated to a particular site.

In summary, the models discussed so far provide a broad and powerful toolset for parametric distributional fits depending on a specified set of additive linear or smooth nonlinear terms. A rather different approach to capturing the dependence on covariates are tree-based models.

- **Regression trees** (Breiman et al., 1984) recursively split the data into more homogeneous subgroups and can thus capture abrupt shifts (Figure 2.2, top left) and approximate nonlinear functions. Furthermore, trees automatically carry out a forward selection of covariates and their interactions.

- **Random forests** (Breiman, 2001a) average the predictions of an ensemble of trees fitted to resampled versions of the learning data. This stabilizes the recursive partitions from individual trees and hence better approximates smooth functions (Figure 2.2, top middle).
2.1 Introduction

Figure 2.1. Parametric modeling developments. (Generalized) linear models (left), generalized additive models (middle), generalized additive models for location, scale, and shape (right).

While classical regression trees and random forests only model the mean of the response we propose to follow the ideas from GAMLSS modeling – as outlined in Figure 2.1 – and combine tree-based methods with parametric distributional models, yielding two novel techniques:

- **Distributional regression trees** (for short: *distributional trees*) split the data into more homogeneous groups with respect to a parametric distribution, thus capturing changes in any distribution parameter like location, scale, or shape (Figure 2.2, bottom left).

- **Distributional regression forests** (for short: *distributional forests*) utilize an ensemble of distributional trees for obtaining stabilized and smoothed parametric predictions (Figure 2.2, top right).
In the following, particular focus is given to distributional forests as a method for obtaining probabilistic forecasts by leveraging the strengths of random forests: the ability to capture both smooth and abruptly changing functions along with simultaneous selection of variables and possibly complex interactions. Thus, these properties make the method particularly appealing in case of many covariates with unknown effects and interactions where it would be challenging to specify a distributional regression model like GAMLSS. However, distributional forests should not be considered as a replacement of GAMLSS but rather as a complementing technique for flexible distributional regression – much like GAMs and random forests are complements for conditional mean regression.

In weather forecasting, the flexibility of distributional forests is especially appealing in mountainous regions and complex terrain where a wide range of local-scale effects are not yet resolved by the NWP models. Thus, effects with abrupt changes and possibly nonlinear interactions might be required to account for site-specific unresolved features. To illustrate this in practice, precipitation forecasts are obtained with distributional forests at 95 meteorological stations in a mountainous region in the Alps, covering mainly Tyrol, Austria, and adjacent areas (see the map in Figure 2.8). More specifically, a Gaussian distribution left-censored at zero, is employed to model 24-hour total precipitation so that the zero-censored point mass describes the probability of observing no precipitation on a given day (see Figure 2.3). Forecasts for July are established based on data from the same month over the years 1985–2012 including 80 covariates derived from a wide range of different NWP quantities. As Figure 2.3 shows, the station-wise forests yield a full distributional forecast for each day – here for one specific day (July 24) at one station (Axams) over four years (2009–2012) – based on the previous 24 years as learning data. The corresponding observations conform reasonably well with the predictions. In Section 2.3 we investigate the performance of distributional forests in this forecasting task in more detail. Compared to three alternative zero-censored Gaussian models distributional forests perform at least on par and sometimes clearly better while requiring no meteorological knowledge about the atmospheric processes which drive formation of precipitation for the model specification. The three alternatives are: a standard ensemble model output statistics approach (EMOS, Gneiting et al., 2005)
2.2 Methodology

To embed the distributional approach from GAMLSS into regression trees and random forests, we proceed in three steps. (1) To fix notation, we briefly review fitting distributions using standard maximum likelihood in Section 2.2.1. (2) A recursive partitioning strategy based on the corresponding scores (or gradients) is introduced in Section 2.2.2, leading to distributional trees. (3) Ensembles of distributional trees fitted to randomized subsamples are employed to establish distributional forests in Section 2.2.3.

The general distributional notation is exemplified in all three steps using the Gaussian distribution left-censored at zero (for short: zero-censored Gaussian). The latter is employed in the empirical case study in Section 2.3 to model power-transformed daily precipitation amounts.

2.2.1 Distributional fit

A distributional model $D(Y, \theta)$ is considered for the response variable $Y \in \mathcal{Y}$ using the distributional family $D$ with $k$-dimensional parameter vector $\theta \in \Theta$ and corresponding log-likelihood function $\ell(\theta; Y)$. The GAMLSS framework (Rigby and Stasinopoulos, 2005) provides a wide range of such distributional families with parameterizations corresponding to location, scale, and shape. Furthermore, censoring and/or truncation of these distributions can be incorporated in the usual straightforward way (see e.g., Long, 1997, Chapter 7.2).

To capture both location and scale of the probabilistic precipitation forecasts while accounting for a point mass at zero (i.e., dry days without rain), a zero-censored Gaussian distribution with location parameter $\mu$ and scale parameter $\sigma$ is employed. Therefore, the corresponding log-likelihood function with parameter vector $\theta = (\mu, \sigma)$ is

$$
\ell(\mu, \sigma; Y) = \begin{cases} 
\log \left\{ \frac{1}{\sigma} \cdot \phi \left( \frac{Y - \mu}{\sigma} \right) \right\}, & \text{if } Y > 0 \\
\log \left\{ \Phi \left( \frac{-\mu}{\sigma} \right) \right\}, & \text{if } Y = 0
\end{cases}
$$

(2.1)

where $\phi$ and $\Phi$ are the probability density function and the cumulative distribution function of the standard normal distribution $\mathcal{N}(0, 1)$. Other distributions $D$ and corresponding log-likelihoods $\ell(\mu, \sigma; Y)$ could be set up in the same way, e.g., for censored shifted gamma distributions (Scheuerer and Hamill, 2015) or zero-censored logistic distributions (Gebetsberger, Messner, Mayr, and Zeileis, 2017).

With the specification of the distribution family and its log-likelihood function the task of fitting a distributional model turns into the task of estimating the distribution parameter $\theta$. This is commonly done by maximum likelihood (ML) based on the learning sample with observations $\{y_i\}_{i=1,\ldots,n}$ of the response variable $Y$. The maximum likelihood estimator (MLE) $\hat{\theta}$ is given by

$$
\hat{\theta} = \arg\max_{\theta \in \Theta} \sum_{i=1}^{n} \ell(\theta; y_i).
$$

(2.2)

Equivalently, this can be defined based on the corresponding first-order conditions

$$
\sum_{i=1}^{n} s(\hat{\theta}, y_i) = 0,
$$

(2.3)
where $s(\theta; y_i)$ is the associated score function

$$s(\theta; y_i) = \frac{\partial \ell}{\partial \theta}(\theta; y_i). \quad (2.4)$$

The latter is subsequently employed as a general goodness-of-fit measure to assess how well the distribution with parameters $\theta$ fits one individual observation $y_i$.

### 2.2.2 Distributional tree

Typically, a single global model $D(Y; \theta)$ is not sufficient for reasonably representing the response distribution. Therefore, covariates $Z = Z_1, \ldots, Z_m \in \mathcal{Z}$ are employed to capture differences in the distribution parameters $\theta$. In weather forecasting, these covariates typically include the output from numerical weather prediction systems and/or lagged meteorological observations.

To incorporate the covariates into the distributional model, they are considered as regressors in additive predictors $g_j(\theta_j) = f_{j,1}(Z) + f_{j,2}(Z) + \ldots$ in GAMLSS. Link functions $g_j(\cdot)$ are used for every parameter $\theta_j$ ($j = 1, \ldots, k$) based on smooth terms $f_{j,k}$ such as nonlinear effects, spatial effects, random coefficients, or interaction surfaces (Klein, Kneib, Lang, and Sohn, 2015). However, this requires specifying the additive terms and their functional forms in advance which can be challenging in practice and potentially require expert knowledge in the application domain, especially if the number of covariates $m$ is large.

Regression trees generally take a different approach for automatically including covariates in a data-driven way and allowing for abrupt changes, nonlinear and non-additive effects, and interactions. In the context of distributional models the goal is to partition the covariate space $\mathcal{Z}$ recursively into disjoint segments so that a homogeneous distributional model for the response $Y$ can be found for each segment with segment-specific parameters. More specifically, the $B$ disjoint segments $B_b$ ($b = 1, \ldots, B$) partition the covariate space

$$\mathcal{Z} = \bigcup_{b=1,\ldots,B} B_b, \quad (2.5)$$

and a local distributional model $D(Y; \theta^{(b)})$ (i.e., with segment-specific parameters $\theta^{(b)}$) is fitted to the response $Y$ in each segment.

To find the segments $B_b$ that are (approximately) homogeneous with respect to the distributional model with given parameters, the idea is to use a gradient-based recursive-partitioning approach. In a given subsample of the learning data this fits the model by ML (see Equation 2.2) and then assesses the goodness of fit by assessing the corresponding scores $s(\hat{\theta}; y_i)$ (see Equation 2.4).

To sum up, distributional trees are fitted recursively via:

1. Estimate $\hat{\theta}$ via maximum likelihood for the observations in the current subsample.
2. Test for associations (or instabilities) of the scores $s(\hat{\theta}, y_i)$ and $Z_{l,i}$ for each partitioning variable $Z_l$ ($l = 1, \ldots, m$).
3. Split the sample along the partitioning variable $Z_l^*$ with the strongest association or instability. Choose the breakpoint with the highest improvement in the log-likelihood or the highest discrepancy.
4. Repeat steps 1–3 recursively in the subsamples until these become too small or there is no significant association/instability (or some other stopping criterion is reached).
2.2 Methodology

Different inference techniques can be used for assessing the association between scores and covariates in step 2. In the following we use the general class of permutation tests introduced by Hothorn, Hornik, Van de Wiel, and Zeileis (2006a) which is also the basis of conditional inference trees (CTree, Hothorn et al., 2006b). Alternatively, one could use asymptotic M-fluctuation tests for parameter instability (Zeileis and Hornik, 2007) as in model-based recursive partitioning (MOB, Zeileis et al., 2008). More details are provided in the Appendix 2.5.

For obtaining probabilistic predictions from the tree for a (possibly new) set of covariates $z = (z_1, \ldots, z_m)$, the observation simply has to be “sent down” the tree and the corresponding segment-specific MLE has to be obtained. Thus, in practice $\hat{\theta}(z)$ does not have to be recalculated for each new $z$ but one can simply extract the parameter estimates for the corresponding segment which have been computed already while learning the tree. However, to understand this estimator conceptually it is useful to denote it as a weighted MLE where the weights select those observations from the learning sample that fall into the same segment:

$$u_i^{\text{tree}}(z) = \sum_{b=1}^{B} 1((z_i \in B_b) \land (z \in B_b)),$$

where $1(\cdot)$ is the indicator function. The predicted distribution for a given $z$ is then fully specified by the estimated parameter $\hat{\theta}(z)$ where

$$\hat{\theta}(z) = \arg\max_{\theta \in \Theta} \sum_{i=1}^{n} u_i^{\text{tree}}(z) \cdot \ell(\theta; y_i).$$

2.2.3 Distributional forest

While the simple recursive structure of a tree model is easy to visualize and interpret, the abrupt changes are often too rough, instable, and impose steps on the model even if the true underlying effect is smooth. Hence, ensemble methods such as bagging or random forests (Breiman, 2001a) are typically applied to smooth the effects, stabilize the model, and improve predictive performance.

The idea of random forests is to learn an ensemble of trees, each on a different learning data obtained through resampling (bootstrap or subsampling). In each node only a random subset of the covariates $Z$ is considered for splitting to reduce the correlation among the trees and to stabilize the variance of the model. For a simple regression random forest the mean of predictions over all trees is considered. In that way changes in the location of the response across the covariates are detected (e.g., in Breiman and Cutler’s random forests, Breiman, 2001a). This idea is now taken one step further by embedding it in a distributional framework based on maximum-likelihood estimation. Distributional forests employ an ensemble of $T$ distributional trees which pick up changes in the “direction” of any distribution parameter by considering the full likelihood and corresponding score function for choosing splitting variables and split points.

To obtain probabilistic predictions from a distributional forest, it still needs to be specified how to compute the parameter estimates $\hat{\theta}(z)$ for a (potentially new) set of covariates $z$. Following Hothorn and Zeileis (2017) we interpret random forests as adaptive local likelihood estimators using the averaged “nearest neighbor weights” (Lin and Jeon, 2006) from the $T$ trees in the forest

$$w_i^{\text{forest}}(z) = \frac{1}{T} \sum_{t=1}^{T} \sum_{b=1}^{B_t} 1((z_i \in B_t^b) \land (z \in B_t^b)) \cdot |B_t^b|,$$

where $|B_t^b|$ denotes the number of observations in the $b$-th segment of the $t$-th tree. Thus, these $u_i^{\text{forest}}(z) \in [0, 1]$ whereas $u_i^{\text{tree}}(z) \in \{0, 1\}$. Hence, weights cannot only be 0 or 1 but change
more smoothly, giving high weight to those observations \( i \) from the learning sample that co-
occur in the same segment \( B_i \) as the new observation \( z \) for many of the trees \( t = 1, \ldots, T \). Consequently, the parameter estimates may, in principle, change for every observation and can be obtained by

\[
\hat{\theta}(z) = \arg\max_{\theta \in \Theta} \sum_{i=1}^{n} w_i^{\text{forest}}(z) \cdot \ell(\theta; y_i).
\] (2.9)

In summary, this yields a parametric distributional regression model (through the score-based
approach) that can capture both abrupt effects and high-order interactions (through the trees)
and smooth effects (through the forest).

Distributional forests share some concepts and algorithmic aspects with other generalizations
of Breiman and Cutler’s random forests. Nearest neighbor weights are employed for aggregation
in survival forests (Hothorn, Lausen, Benner, and Radespiel-Tröger, 2004), quantile regression
forests (Meinshausen, 2006), transformation forests (Hothorn and Zeileis, 2017), and generalized
random forests for causal inferences (Athey, Tibshirani, and Wager, 2019). These procedures
aggregate over trees fitted to specific score functions (e.g., log rank scores in survival trees,
model residuals in transformation or generalized forests). Distributional forests, in contrast to
nonparametric approaches, provide a compromise between model flexibility and interpretability:
The parameters of a problem-specific distribution (zero-censored Gaussian for precipitation)
have a clear meaning but may depend on external variables in a quite general way.

2.3 Probabilistic precipitation forecasting in complex terrain

Many statistical weather forecasting models leverage the strengths of modern numerical ensemble
prediction systems (EPSs; see Bauer, Thorpe, and Brunet, 2015). EPSs not only predict the
most likely future state of the atmosphere but provide information about the uncertainty for a
specific quantity and weather situation. This is done by running the NWP model several times
using slightly perturbed initial conditions and model specifications to account for uncertainties
in both, the initial atmospheric state and the NWP model (and its parametrizations). One
frequently-used method based on distributional regression models is the ensemble model output
statistics (EMOS) approach first proposed by Gneiting et al. (2005) to produce high-quality
forecasts for specific quantities and sites. In case of precipitation forecasting, EMOS typically
uses the ensemble mean of “total precipitation” (tp) forecasts as the predictor for the location
parameter \( \mu \) and the corresponding ensemble standard deviation for the scale parameter \( \sigma \), e.g.,
assuming the observations to follow a zero-censored Gaussian distribution. This distributional
approach of modeling both parameters allows to correct for possible errors of the NWP ensemble
in both, the expectation but also the uncertainty of a specific forecast. Thus, a basic EMOS
specification typically models the two distribution parameters by two linear predictors, e.g.,

\[
\mu = \beta_0 + \beta_1 \cdot \text{mean}(tp) \quad \text{and} \quad \log(\sigma) = \gamma_0 + \gamma_1 \cdot \log(\text{sd}(tp))
\]

with regression coefficients \( \beta_0, \beta_1, \gamma_0, \) and \( \gamma_1 \) (where the log link assures positivity of the scale parameter, following Gebetsberger
et al., 2017).

While this approach alone is already highly effective in the plains, it typically does not
perform as well in complex terrain due to unresolved effects in the NWP system (Bauer et al.,
2015). For example, in the Tyrolean Alps – considered in the following case study – the NWP grid
cells of \( 50 \times 50 \) km\(^2\) are too coarse to capture single mountains, narrow valleys, etc. Therefore,
it is often possible to substantially improve the predictive performance of a basic EMOS by
including additional predictor variables, either from local meteorological observations or an
NWP model. Unfortunately, it is typically unknown which variables are relevant for improving
the predictions. Simply including all available variables may be computationally burdensome
and can lead to overfitting but, on the other hand, excluding too many variables may result in a loss of valuable information. Therefore, selecting the relevant variables and interactions among all possible covariates is crucial for improving the statistical forecasting model.

In the following, it is illustrated how distributional forests can solve this problem without requiring prior expert knowledge about the meteorological covariates. For fitting the forest only the response distribution and the list of potential predictor variables need to be specified (along with a few algorithmic details) and then the relevant variables, interactions, and potentially nonlinear effects are determined automatically in a data-driven way. Here, we employ a zero-censored Gaussian distribution and 80 predictor variables computed from ensemble means and spreads of various NWP outputs. The predictive performance of the forest is compared to three

Table 2.1. Basic covariates together with the number (#) and the type of variations. Time periods indicate aggregation time periods in hours after NWP model initialization (e.g., 6–30 corresponds to +6 h to +30 h ahead forecasts, 0600 UTC to 0600 UTC of the next day). Note: *Minimum values of \(d_{swrf}\) over 24 h are always zero and thus neglected.

<table>
<thead>
<tr>
<th>Basic covariates</th>
<th># Variations</th>
</tr>
</thead>
<tbody>
<tr>
<td>(tp): total precipitation,</td>
<td>12</td>
</tr>
<tr>
<td>power transformed (by (1.6^{-1}))</td>
<td>ensemble mean of sums over 24h,</td>
</tr>
<tr>
<td></td>
<td>ensemble std. deviation of sums over 24h,</td>
</tr>
<tr>
<td>(cape): convective available</td>
<td>ensemble minimum of sums over 24h,</td>
</tr>
<tr>
<td>potential energy,</td>
<td>ensemble maximum of sums over 24h</td>
</tr>
<tr>
<td>power transformed (by (1.6^{-1}))</td>
<td>all for 6–30</td>
</tr>
<tr>
<td></td>
<td>ensemble mean of sums over 6h</td>
</tr>
<tr>
<td></td>
<td>for 6–12, 12–18, 18–24, 24–30</td>
</tr>
<tr>
<td></td>
<td>ensemble std. deviation of sums over 6h</td>
</tr>
<tr>
<td></td>
<td>for 6–12, 12–18, 18–24, 24–30</td>
</tr>
<tr>
<td>(d_{swrf}): downwards short wave</td>
<td>6</td>
</tr>
<tr>
<td>radiation flux (“sunshine”)</td>
<td>ensemble mean of mean values,</td>
</tr>
<tr>
<td></td>
<td>ensemble mean of minimum values*,</td>
</tr>
<tr>
<td></td>
<td>ensemble mean of maximal values,</td>
</tr>
<tr>
<td></td>
<td>ensemble std. deviation of mean values,</td>
</tr>
<tr>
<td></td>
<td>ensemble std. deviation of minimum values*,</td>
</tr>
<tr>
<td></td>
<td>ensemble std. deviation of maximal values,</td>
</tr>
<tr>
<td></td>
<td>all over 6–30</td>
</tr>
<tr>
<td>(msl): mean sea level pressure</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>(pwat): precipitable water</td>
<td></td>
</tr>
<tr>
<td>(t_{max}): 2m maximum temperature</td>
<td></td>
</tr>
<tr>
<td>(t_{colc}): total column-integrated</td>
<td></td>
</tr>
<tr>
<td>condensate</td>
<td></td>
</tr>
<tr>
<td>(t_{500}): temperature on 500 hPa</td>
<td></td>
</tr>
<tr>
<td>(t_{700}): temperature on 700 hPa</td>
<td></td>
</tr>
<tr>
<td>(t_{850}): temperature on 850 hPa</td>
<td></td>
</tr>
<tr>
<td>(t_{diff500850}): temperature</td>
<td>3</td>
</tr>
<tr>
<td>difference 500 to 850 hPa</td>
<td>ensemble mean of difference in mean,</td>
</tr>
<tr>
<td></td>
<td>ensemble minimum of difference in mean,</td>
</tr>
<tr>
<td></td>
<td>ensemble maximum of difference in mean</td>
</tr>
<tr>
<td></td>
<td>all over 6–30</td>
</tr>
<tr>
<td>(t_{diff500700}): temperature</td>
<td></td>
</tr>
<tr>
<td>difference 500 to 700 hPa</td>
<td></td>
</tr>
<tr>
<td>(t_{diff700850}): temperature</td>
<td></td>
</tr>
<tr>
<td>difference 700 to 850 hPa</td>
<td></td>
</tr>
<tr>
<td>(msl\textunderscore diff): mean sea level pressure</td>
<td>1</td>
</tr>
<tr>
<td>difference</td>
<td>(msl\textunderscore mean\textunderscore max (-) (msl\textunderscore mean\textunderscore min) over 6–30</td>
</tr>
</tbody>
</table>
other zero-censored Gaussian models: (a) a basic EMOS, (b) a GAMLSS with prespecified effects and interactions based on meteorological knowledge/experience, and (c) a boosted GAMLSS with automatic selection of smooth additive terms based on all 80 predictor variables.

2.3.1 Data

Learning and validation data consist of observed daily precipitation sums provided by the National Hydrographical Service (BMLFUW, 2016) and numerical weather forecasts from the U.S. National Oceanic and Atmospheric Administration (NOAA). Both, observations and forecasts are available for 1985–2012 and the analysis is exemplified using July, the month with the most precipitation in Tyrol.

Observations are obtained for 95 stations all over Tyrol and surroundings, providing 24-hour precipitation sums measured at 0600 UTC and rigorously quality-checked by the National Hydrographical Service. NWP outputs are obtained from the second-generation reforecast data set of the global ensemble forecast system (GEFS, Hamill, Bates, Whitaker, Murray, Fiorino, Galarneau Jr., Zhu, and Lapenta, 2013). This data set consists of an 11-member ensemble based on a fixed version of the numerical model and a horizontal grid spacing of about $50 \times 50$ km$^2$ initialized daily at 0000 UTC from December 1984 to present providing forecasts on a 6-hourly temporal resolution. Each of the 11 ensemble members uses slightly different perturbed initial conditions to predict the situation-specific uncertainty of the atmospheric state.

From the GEFS, 14 basic forecast variables are considered with up to 12 variations each such as mean/maximum/minimum over different aggregation time periods. A detailed overview is provided in Table 2.1, yielding 80 predictor variables in total.

To remove large parts of the skewness of precipitation data, a power transformation (Box and Cox, 1964) is often applied, e.g., using cubic (Stidd, 1973) or square root (Hutchinson, 1998) transformations. However, the power parameter may vary for different climatic zones or temporal aggregation periods and hence we follow Stauffer, Mayr, Messner, Umlauf, and Zeileis (2017a) in their choice of $1.6^{-1}$ as a suitable power parameter for precipitation in the region of Tyrol. The same power transformation is applied to both the observed precipitation sums and the NWP outputs “total precipitation” ($tp$) and “convective available potential energy” ($cape$).

2.3.2 Models and evaluation

The following zero-censored Gaussian regression models are employed in the empirical case study, see Table 2.2 for further details:

- **Distributional forest:** All 80 predictor variables are considered for learning a forest of 100 trees. Subsampling is employed for each tree using a third of the predictors in each split of the tree (argument `mtry` in our implementation `distforest`, with more “computational details” provided at the end of the manuscript). Parameters are estimated by adaptive local likelihood based on the forest weights as described in Section 2.2. The stopping criteria are the minimal number of observations to perform a split ($\text{minsplit} = 50$), the minimal number of observations in a segment ($\text{minbucket} = 20$), and the significance level for variable selection ($\alpha = 1$). The latter means that no early stopping (or “pre-pruning”) is applied based on results of the statistical tests.

- **EMOS:** The basic ensemble model output statistics models use the ensemble mean of total precipitation as regressor in the location submodel and the corresponding ensemble standard deviation in the scale submodel. The parameters are estimated by maximum likelihood, using an identity link for the location part and a log link for the scale part (following the advice of Gebetsberger et al., 2017).
2.3 Probabilistic precipitation forecasting in complex terrain

- **Prespecified GAMLSS**: Smooth additive splines are selected for the most relevant predictors based on meteorological expert knowledge following Stauffer et al. (2017b). More specifically, based on the 80 available variables, eight terms are included in the location submodel and five in the scale submodel. Both involve an interaction of \textit{tp} and \textit{cape} in the afternoon (between 1200 UTC and 1800 UTC) to capture the potential for thunderstorms that frequently occur in summer afternoons in the Alps. The model is estimated by maximum penalized likelihood using a backfitting algorithm (Stasinopoulos and Rigby, 2007).

- **Boosted GAMLSS**: Smooth additive splines are selected automatically from all 80 available variables, using non-cyclic boosting for parameter estimation (Hofner et al., 2016; Messner et al., 2017). This updates the predictor terms for the location or scale submodels iteratively by maximizing the log-likelihood only for the variable yielding the biggest improvement. The iteration stops early – before fully maximizing the in-sample likelihood – based on a (computationally intensive) out-of-bag bootstrap estimate of the log-likelihood. The grid considered for the number of boosting iterations (\textit{mstop}) is: 50, 75, \ldots, 975, 1000.

The predictive performance in terms of full probabilistic forecasts is assessed using the continuous ranked probability score (CRPS, Hersbach, 2000). For each of the models this assesses the discrepancy of the predicted distribution function \( F \) from the observation \( y \) by

\[
CRPS(y, F) = \int_{-\infty}^{\infty} (F(z) - \mathbb{1}(y \leq z))^2 dz
\]

where \( \mathbb{1}(\cdot) \) is the indicator function. In the subsequent applications, the mean CRPS is always evaluated out of sample, either using cross-validation or a hold-out data set (2009–2012) that was not used for learning (1985–2008). CRPS is a proper scoring rule (Gneiting and Raftery, 2007) often used within the meteorological community. Lower values indicate better performance.

To assess differences in the improvement of the forests and GAMLSS models over the basic EMOS, a CRPS-based skill score with EMOS as the reference method is computed:

\[
CRPSS_{\text{method}} = 1 - \frac{\text{CRPS}_{\text{method}}}{\text{CRPS}_{\text{EMOS}}}
\]

**Table 2.2.** Overview of models with type of covariate dependency and included covariates for each distribution parameter. \( A \ast B \) indicates an interaction between covariate \( A \) and \( B \).
2.3.3 Application for one station

In a first step, we show a detailed comparison of the competing models for one observation site, Axams in Tyrol (in the center of the study area, see Figure 2.8). As for all other stations, daily precipitation observations and numerical weather predictions are available for the month of July from 1985 through 2012. In Figure 2.3 in the introduction the probabilistic forecasts from the distributional forest, trained on 1985–2008, for July 24 in 2009–2012 have already been shown as a motivational example. In particular, the figure depicts the forecasted point mass at zero (i.e., the probability of a dry day) along with the forecasted probability density function for the total amount of precipitation. Based on this illustration it can be observed that the four sample forecasts differ considerably in location $\mu$, scale $\sigma$, and the amount of censoring while conforming quite well with the actual observations from these days. While this is a nice illustrative example we are interested in the overall predictive performance and calibration of the distributional fits. More details of this assessment as well as an application to 14 further meteorological stations is provided in Supplement B (Schlosser, Hothorn, Stauffer, and Zeileis, 2019c).

To assess calibration, Figure 2.4 shows residual QQ plots for out-of-sample predictions (2009–2012) from the different models trained on 1985–2008. Due to the point masses at zero 100 draws from the randomized quantile residuals (Dunn and Smyth, 1996) are plotted in semi-transparent gray. Overall, the randomized quantile residuals conform quite well with the theoretical standard

Figure 2.4. Out-of-sample residual QQ plots (2009–2012) for station Axams based on models learned on data from 1985–2008.
normal quantile (i.e., form a straight line close to the diagonal), indicating that all four models are sufficiently well calibrated. This is also supported by the corresponding probability integral transform (PIT, Gneiting, Balabdaoui, and Raftery, 2007) histograms for station Axams in Supplement B (Schlosser et al., 2019c) which contains a more detailed explanation of residual QQ plots and PIT histograms.

To assess the predictive performance, a full cross-validation is carried out rather than relying on just the one fixed test set for the years 2009–2012. To do so, a 10 times 7-fold cross-validation is carried out where each repetition splits the available 28 years into 7 subsets of 4 randomly-selected (and thus not necessarily consecutive) years. The models are learned on 6 folds (= 24 years) and evaluated on the 7-th fold (= 4 years) using the average CRPS across all observations. The resulting 10 CRPS skill scores are displayed by boxplots in Figure 2.5 using EMOS as the reference model (horizontal line at a CRPSS of 0). Both GAMLSS models and the distributional forest perform distinctly better than the EMOS model. While the two GAMLSS lead to an improvement of around 4 percent, the distributional forest has a slightly higher improvement of around 5.5 percent in median.

Finally, it is of interest how this improvement in predictive performance by the distributional forest is accomplished, i.e., which of the 80 covariates are selected in the trees of the forest. As the 100 trees of the forest do not allow to simply assess the variables’ role graphically, a common solution for random forests in general is to consider variable importance measures. Here, this is defined as the amount of change in CRPS when the association between one covariate and the response variable is artificially broken through permutation (and thus also breaking the association to the remaining covariates).

Figure 2.6 shows the 10 covariates with the highest permutation importance (i.e., change in CRPS) for station Axams. As expected the NWP outputs for total precipitation ($tp$) are particularly important along with total column-integrated condensate ($tcolc$). Also, both variables occur in various transformations such as means (either of the full day or certain parts of the afternoon), spreads, or minima/maxima. Thus, while the covariates themselves are not surprising, selecting a GAMLSS with a particular combination of all the transformations would be much more challenging.
Variable importance: mean increase in CRPS

Figure 2.6. CRPS-based variable importance for the top 10 covariates in the distributional forest. Based on data for station Axams, learning period 1985–2008 and assessed in 2009–2012.

2.3.4 Application for all stations

After considering only one observational site up to now, a second step evaluates and compares the competing methods on all 95 available stations. As in the previous section, all models are learned on the first 24 years and evaluated by the average CRPS on the last 4 years. More specifically, the CRPS skill score against the EMOS model is computed for the out-of-sample predictions at each station and visualized by parallel coordinates plots with boxplots superimposed in Figure 2.7. Overall, distributional forests have a slightly higher improvement in CRPSS compared to the two GAMLSS which is best seen by looking at the boxplots and the green line representing the results for station Axams. The underlying parallel coordinates additionally bring out that the prespecified GAMLSS sometimes performs rather differently (sometimes better, sometimes worse) compared to the two data-driven models. Values below zero show that, for some stations, EMOS performs better than the more complex statistical methods.

To assess whether these differences in predictive performance are due to differences in the topography, Figure 2.8 shows a brief spatial summary of all stations. Each station is illustrated by a symbol that conveys which model performed best in terms of CRPS on the last 4 years of the data. Additionally, the color of the symbol indicates the CRPS difference between distributional forest and the best-performing other model. Green signals that the distributional forest performs better than the other models whereas red signals that another model performs better. Overall the distributional forest performs on par (gray) or better (green) for the majority of stations. Only for a few stations in the north-east EMOS performs best, and in East Tyrol the prespecified GAMLSS performs particularly well in the validation period (2009–2012). Partially, this can be attributed to random variation as the differences at several stations are mitigated when considering a full cross-validation rather than a single split into learning and validation period (see Supplement B, Schlosser et al., 2019c, and the corresponding discussion in the next section). Further differences are possibly due to East Tyrol lying in a different climate zone, south of the main Alpine Ridge. Hence, long-term climatological characteristics as well as the precipitation patterns in 2009–2012 differ from North Tyrol, conforming particularly well with the additive effects from the prespecified GAMLSS.
Figure 2.7. CRPS skill score for each station (gray lines with boxplots superimposed). Station Axams is highlighted in green and the horizontal orange line pertains to the reference model EMOS. The models are learned on 1985–2008 and validated for 2009–2012.

Figure 2.8. Map of Tyrol coding the best-performing model for each station (type of symbol) when learned on 1985–2008 and validated for 2009–2012. The color codes whether the distributional forest had lower (green) or higher (red) CRPS compared to the best of the other three models. The gray background shows the local topography (Robinson, Regetz, and Guralnick, 2014). Station Axams is highlighted in bold.
2.4 Discussion

Distributional regression modeling is combined with tree-based modeling to obtain a novel and flexible method for probabilistic forecasting. The resulting distributional trees and forests can capture abrupt and nonlinear effects and interactions in a data-driven way. By basing the split point and split variable selection on a full likelihood and corresponding score function, the trees and forests can not only pick up changes in the location but also the scale or shape of any distributional family.

Distributional forests are an attractive alternative when prespecifying or boosting all possible effects and interactions in a GAMLSS model is challenging. Distributional forests are rather straightforward to specify requiring only little prior subject matter knowledge and also work well in the presence of many potential covariates. The application to precipitation forecasting in complex terrain illustrates that distributional forests often perform on par or even better than their GAMLSS counterparts. Hence, they form a useful addition to the already available toolbox of probabilistic forecasts for disciplines such as meteorology.

Variable selection

Generally, there are many possibilities how to specify the variables that are to be included in a distributional regression model. Especially for a low number of covariates, the GAMLSS approach offers a powerful framework in which penalized estimation of both smooth main effects and corresponding interaction surfaces yields models that often balance good predictive performance with high interpretability (see for example Wood, Scheipl, and Faraway, 2013; Goicoa, Adin, Ugarte, and Hodges, 2018; Ugarte, Adin, and Goicoa, 2017). However, if the number of covariates is high, including all (or many) main effects and interactions in a GAMLSS typically becomes challenging both in terms of interpretability and computational complexity/stability (see also Hofner et al., 2016).

In the precipitation forecasting application, as presented in Section 2.3, 80 covariates are considered which corresponds to 3160 potential pairwise interactions (and even more higher-order interactions). Therefore, only main effects are considered for the boosted GAMLSS while the prespecified GAMLSS also includes selected interactions chosen based on meteorological expert knowledge. In contrast, the distributional forest requires no prespecification as covariates and corresponding interactions are selected automatically. Thus, distributional forests are an appealing alternative to (boosted) GAMLSS in weather forecasting tasks as the main concern is typically not so much interpretability but forecasting skill and (semi-)automatic application on a larger domain (see also the discussion in Rasp and Lerch, 2018).

Distributional specifications for precipitation modeling

Choosing an adequate distributional family is an important step for establishing a well-fitting model. A zero-censored Gaussian distribution is employed in this manuscript as this has been found to be an appropriate choice for precipitation modeling in earlier literature (e.g., Stauffer et al., 2017a). To test for robustness against distributional misspecification, two alternative distributional specifications have been considered in Supplement A (Schlosser, Hothorn, Stauffer, and Zeileis, 2019a): Using the same evaluations as in Section 2.3.4, all models are additionally fitted for 15 meteorological stations using a zero-censored logistic distribution in order to account for heavier tails and a two-part Gaussian hurdle model combining a binary model for zero vs. positive precipitation and a separate Gaussian model, truncated at zero, for the positive precipitation observations. Both specifications yield qualitatively similar results as for the zero-censored Gaussian distribution. For some stations the two-part hurdle model leads to small improvements, however at the expense of increased variability across stations (especially for
EMOS and the boosted GAMLSS). Overall, the results from this manuscript are quite robust across these distributional specifications, especially for the distributional forests.

Moreover, one could consider a distribution including an additional parameter for capturing skewness (as in Scheuerer and Hamill, 2015; Baran and Nemoda, 2016). However, this would go beyond the mean/variance specification of the NGR that is widely used in ensemble post-processing. Therefore, this contribution investigates the effects of using the same distributional family with a novel strategy for specifying dependence on covariates.

More general distributional specifications

Beyond the task of modeling precipitation it is of interest how well distributional forests perform in combination with other more general distributional specifications. It has been shown previously in the literature that using a score- or gradient-based selection of splitting variables outperforms a mean-based selection with subsequent flexible distributional modeling: For example, both Athey et al. (2019, Figure 2) and Hothorn and Zeileis (2017, Figure 1) demonstrated (independently) that their respective score-based random forest algorithms outperform the mean-based quantile regression forests of Meinshausen (2006) in a setup where only the variance of a normal response variable changes across the considered covariates. However, if all distribution parameters are closely correlated with the distribution mean the forests with different splitting strategies all perform similarly, provided a sufficiently flexible distribution is employed for the final predictions (see Hothorn and Zeileis, 2017, Section 7).

Similarly, the score-based distributional forests introduced in this manuscript proved to be quite robust to the different distributional specifications considered. While all specifications focus on capturing mean-variance effects note that these parameters are never fully orthogonal but can actually become quite closely correlated due to the censoring (or truncation and/or zero-inflation considered in Supplement A (Schlosser et al., 2019a)).

However, exploring extensions to more flexible parametric distributions (e.g., such as the Dagum distribution considered by Klein et al., 2015, in GAMLSS-type models) as well as transformation model specifications (e.g., as in Hothorn and Zeileis, 2017) are of interest for future research.

Axams vs. other meteorological stations

Axams was chosen as the meteorological station for the more extensive evaluations in Section 2.3.3 as it yields fairly typical results and is geographically in the center of the study area and closest to Innsbruck, the capital of Tyrol and the work place of three of the authors. To show that qualitatively similar results are obtained for other meteorological stations, Supplement B (Schlosser et al., 2019c) carries out the same evaluation for 14 further stations. These cover a wide range of geographical locations/altitudes and a mix of different best-performing models in the single-split setting reported in Section 2.3.4.

The supplement shows that some of the differences in forecast skill from Figure 2.8 even out in the cross-validation with distributional forests typically performing at least as well as the best of the other models at most stations. In particular, this also includes three stations in East Tyrol where the prespecified GAMLSS performs best in the single-split setting (learning based on 1985–2008 and validation for 2009–2012).

Tuning parameters

Selecting tuning parameters for flexible regression models is important not only in terms of predictive accuracy but also computational complexity. For the application in Section 2.3 tuning parameters are selected based on advice from the literature as well as our own experiences. As
Hastie, Tibshirani, and Friedman (2001) and Breiman (2001a) recommend to build full-grown trees, early stopping upon non-significance is disabled ($\alpha = 1$) and low values are used for minsplit (= 50) and minbucket (= 20), while assuring that minsplit is sufficiently large for reasonably obtaining MLEs of all parameters in each segment of the tree.

Applying the Law of Large Numbers it can be shown that random forests do not overfit as the number of trees increases (Breiman, 2001a; Hastie et al., 2001; Biau and Scornet, 2016). Therefore, in principle, forests can be built with a very large number of trees (ntree) as this cannot deteriorate the predictions. However, “[…] the computational cost for inducing a forest increases linearly with the number of trees, so a good choice results from a trade-off between computational complexity and accuracy” (Biau and Scornet, 2016, p. 205). Following this advice, we decided to build forests consisting of 100 trees.

Computational difficulties
As stated by Hofner et al. (2016) the AIC-based variable selection methods implemented in the R package gamlss “[…] can be unstable, especially when it comes to selecting possibly different sets of variables for multiple distribution parameters.” We have noticed computational problems when applying gamlss in certain settings within the cross-validation framework as it did not succeed in fitting the model. In these cases the prespecified GAMLSS was not taken into consideration in the comparison of all applied models.

Computational details
The proposed methods are in the R package disttree (version 0.1.0) based on the partykit package (version 1.2.3), both available on R-Forge at (https://R-Forge.R-project.org/projects/partykit/). The function distforest learns the distributional forests proposed in this manuscript by combining the general cforest function from partykit with the function distfit for fitting distributional models by maximum likelihood. Analogously, disttree can learn a single distributional tree by combining ctree with distfit. All functions can either be used with GAMLSS family objects from the R package gamlss.dist (Stasinopoulos and Rigby, 2007, version 5.0.6) or with custom lists containing all required information about the distribution family.

In addition to disttree, Section 2.3 employs R package crch (Messner, Mayr, and Zeileis, 2016, version 1.0.1) for the EMOS models, gamlss (Stasinopoulos and Rigby, 2007, version 5.1.0) for the prespecified GAMLSS, and gamboostLSS (Hofner et al., 2016, version 2.0.1) for the boosted GAMLSS.

The fitted distributional forest for July 24 and observation station Axams (including Figure 2.3) is reproducible using demo("RainAxams", package = "disttree"). This also includes fitting the other zero-censored Gaussian models considered in this paper and generating the corresponding QQ plots (Figure 2.4) and PIT histograms (Schlosser et al., 2019c, Supplement B). Full replication of all results can be obtained with demo("RainTyrol", package = "disttree") requiring the companion R package RainTyrol (version 0.1.0), also available within the R-Forge project. The results presented in Supplement A (Schlosser et al., 2019a) and Supplement B (Schlosser et al., 2019c) can be reproduced using demo("RainDistributions", package = "disttree") and demo("RainStationwise", package = "disttree"), respectively.
Supplements

The two following supplementary files to this article can be found in the online version of the publication at https://doi.org/10.1214/19-AOAS1247#supplemental:

**Supplement A: Different Response Distributions** (DOI: 10.1214/19-AOAS1247SUPPA)
To assess the goodness of fit of the Gaussian distribution, left-censored at zero, this supplement employs the same evaluations as in the main manuscript but based on two other distributional assumptions: A logistic distribution, left-censored at zero, is employed to potentially better capture heavy tails – and a two-part hurdle model combining a binary model for zero vs. positive precipitation and a Gaussian model, truncated at zero, for the positive precipitation observations.

**Supplement B: Stationwise Evaluation** (DOI: 10.1214/19-AOAS1247SUPPB)
To show that Axams is a fairly typical station and similar insights can be obtained for other stations as well, this supplement presents the same analysis as in Section 2.3.3 of the main manuscript for 14 further meteorological stations.

Acknowledgements

An extended research stay of Torsten Hothorn in Innsbruck (August 2017 to January 2018) was financially supported by the Swiss National Science Foundation, grant number SNF IZSEZ0 177091.

2.5 Appendix: Tree algorithm

In the following, the tree algorithm applied in the empirical case study discussed in this paper is explained. For notational simplicity, the testing and splitting procedure is described for the root node, i.e., the entire learning sample with observations \( \{ y_i \}_{i=1}^n \), \( n \in \mathbb{N} \). In each child node the corresponding subsample depends on the foregoing split(s).

After fitting a distributional model \( D(Y, \theta) \) to the learning sample with observations \( \{ y_i \}_{i=1}^n \) as explained in Section 2.2.1 the resulting estimated parameter \( \hat{\theta} = (\hat{\theta}_1, \ldots, \hat{\theta}_k) \), \( k \in \mathbb{N} \) can be plugged in the score function \( s(\theta, Y) \). In that way a goodness-of-fit measurement is obtained for each parameter \( \theta_j \) and each observation \( y_i \). To use this information, statistical tests are employed to detect dependencies between the score values

\[
s(\hat{\theta}, y) = \begin{pmatrix}
  s(\hat{\theta}, y_1) & s(\hat{\theta}, y_1) & \ldots & s(\hat{\theta}, y_1) \\
  \vdots & \vdots & \ddots & \vdots \\
  s(\hat{\theta}, y_n) & s(\hat{\theta}, y_n) & \ldots & s(\hat{\theta}, y_n)
\end{pmatrix}
\]  
(2.12)

and each variable \( Z_l \in \{ Z_1, \ldots, Z_m \} \). More formally, the following hypotheses are assessed with permutation tests:

\[
H^l_0 : s(\hat{\theta}, Y) \perp Z_l.
\]  
(2.13)

The permutation tests are based on the multivariate linear statistic

\[
T_l = vec \left( \sum_{i=1}^n v_l(Z_{li}) \cdot s(\hat{\theta}, Y_i) \right)
\]  
(2.14)

where \( s(\hat{\theta}, Y_i) \in \mathbb{R}^{1 \times k} \) and the type of the transformation function \( v_l \) depends on the type of the split variable \( Z_l \). If \( Z_l \) is numeric then \( v_l \) is simply the identity function \( v_l(Z_{li}) = Z_{li} \) and
therefore \( T_l \in \mathbb{R}^k \) as the “vec” operator converts the \( 1 \times k \) matrix into a \( k \) column vector. If \( Z_l \) is a categorical variable with \( H \) categories then \( v_l(Z_{li}) = (I(Z_{li} = 1), \ldots, I(Z_{li} = H)) \) such that \( v_l \) is a \( H \)-dimensional unit vector where the element corresponding to the value of \( Z_{li} \) is 1. In this case the statistic \( T_l \in \mathbb{R}^{H \cdot k} \) as the “vec” operator converts the \( H \times k \) matrix into a \( H \cdot k \) column vector by column-wise combination. Observations with missing values are excluded from the sums.

With the conditional expectation \( \mu_l \) and the covariance \( \Sigma_l \) of \( T_l \) as derived by Strasser and Weber (1999) the test statistic can be standardized. The observed multivariate linear statistic \( t_l \) which is either a \( k \)- or \( k \cdot H \)-dimensional vector, depending on the scale of \( Z_l \), is mapped onto the real line by a univariate test statistic \( c \). In the application of this paper a quadratic form is chosen, such that

\[
c_{\text{quad}}(t_l, \mu_l, \Sigma_l) = (t_l - \mu_l)\Sigma_l^+(t_l - \mu_l)^\top
\]

where \( \Sigma_l^+ \) is the Moore-Penrose inverse of \( \Sigma_l \). Alternatively, the maximum of the absolute values of the standardized linear statistic can be considered (\( c_{\text{max}} \)).

Strasser and Weber (1999) showed that the asymptotic conditional distribution of the linear statistic \( t_l \) is a multivariate normal with parameters \( \mu_l \) and \( \Sigma_l \). Hence, the asymptotic conditional distribution of \( c(t_l, \mu_l, \Sigma_l) \) is either normal (for \( c_{\text{max}} \)) or \( \chi^2 \) (for \( c_{\text{quad}} \)).

The smaller the \( p \)-value corresponding to the standardized test statistic \( c(t_l, \mu_l, \Sigma_l) \) is the stronger the discrepancy from the assumption of independence between the scores and the split variable \( Z_l \). After Bonferroni-adjusting the \( p \)-values it has to be assessed whether any of the resulting \( p \)-values is beneath the selected significance level. If so, the partitioning variable \( Z_l \) with the lowest \( p \)-value is chosen as splitting variable. Otherwise no further split is made in this node as the stopping criterion of no \( p \)-values being below the significance level is fulfilled.

The breakpoint that leads to the highest discrepancy between score functions in the two resulting subgroups is selected as split point. This is measured by the linear statistic

\[
T_{qr} = \sum_{i \in \mathcal{B}_{qr}} s(\hat{\theta}, Y_i) \quad (2.16)
\]

for \( q \in \{1, 2\} \) where \( \mathcal{B}_{1r} \) and \( \mathcal{B}_{2r} \) are the two new subgroups, without any particular ordering, that are defined by splitting in split point \( r \) of variable \( Z_l \). The split point is then chosen as follows:

\[
r^* = \arg\min_r \min_{q=1,2} (c(T_{qr}, \mu_{qr}, \Sigma_{qr})).
\]

One repeats the testing and splitting procedure in each of the resulting subgroups until some stopping criterion is reached. This criterion can for example be a minimal number of observations in a node or a minimal \( p \)-value for the statistical tests. In that way pre-pruning is applied in order to find right-sized trees and hence avoid overfitting.

This permutation-test-based tree algorithm is presented in Hothorn et al. (2006b) as the CTree algorithm. A different framework to build a likelihood-based tree is provided by the MOB algorithm which is based on M-fluctuation tests (Zeileis et al., 2008).
Chapter 3

Paper II: The Power of Unbiased Recursive Partitioning

The Power of Unbiased Recursive Partitioning: A Unifying View of CTree, MOB, and GUIDE

submitted to Advances in Data Analysis and Classification

Lisa Schlosser*, Torsten Hothorn**, Achim Zeileis*

*University of Innsbruck, Department of Statistics, Universitätsstr. 15, A-6020 Innsbruck, Austria
**University of Zurich, Epidemiology, Biostatistics and Prevention Institute, Hirschengraben 84, CH-8001 Zürich, Switzerland

Abstract A core step of every algorithm for learning regression trees is the selection of the best splitting variable from the available covariates and the corresponding split point. Early tree algorithms (e.g., AID, CART) employed greedy search strategies, directly comparing all possible split points in all available covariates. However, subsequent research showed that this is biased towards selecting covariates with more potential splitting points. Therefore, unbiased recursive partitioning algorithms have been suggested (e.g., QUEST, GUIDE, CTree, MOB) that first select the covariate based on statistical inference using p-values that are adjusted for the possible split points. In a second step a split point optimizing some objective function is selected in the chosen split variable. However, different unbiased tree algorithms obtain these p-values from different inference frameworks and their relative advantages or disadvantages are not well understood, yet. Therefore, three different popular approaches are considered here: classical categorical association tests (as in GUIDE), conditional inference (as in CTree), and parameter instability tests (as in MOB). First, these are embedded into a common inference framework encompassing parametric model trees, in particular linear model trees. Second, it is assessed how different building blocks from this common framework affect the power of the algorithms to select the appropriate covariates for splitting: observation-wise goodness-of-fit measure (residuals vs. model scores), dichotomization of residuals/scores at zero, and binning of possible split variables. This shows that specifically the goodness-of-fit measure is crucial for the power of the procedures, with model scores without dichotomization performing much better in many scenarios.

Keywords: classification and regression trees, independence tests, recursive partitioning, simulation.
3.1 Introduction

In many situations fitting one global model to a given data set can be very challenging, especially if the data contains lots of different features with strong variation and complex interactions. Therefore, separating the data into more homogeneous subgroups based on a set of covariates first can simplify the task and fitting a local model to each of the resulting subgroups often leads to better results. This separation can be done by applying a tree algorithm. While almost all algorithms proposed in the literature follow the general idea of splitting the data such that some objective function is optimized locally, they differ in their specific approaches to selecting a split variable and the corresponding split point. Some of the first tree algorithms (e.g., AID, Morgan and Sonquist, 1963; CART, Breiman et al., 1984) rely on exhaustive search procedures to find both, the best split point and split variable in one step by directly comparing all possible split points in all possible split variables. However, it has been shown that this is not only computationally expensive but also biased towards split variables with many possible split points (Doyle, 1973; Kim and Loh, 2001). Therefore, selecting a split variable in a first step and then searching for the best split point only within this variable in a separated second step is a more promising strategy as applied for example by the algorithms QUEST (Loh and Shih, 1997), GUIDE (Loh, 2002), CTree (Hothorn et al., 2006b) and MOB (Zeileis et al., 2008).

For the first step of selecting a split variable they all share the same basic concept of choosing the covariate which shows the highest association to the response variable based on \( p \)-values provided by a statistical test. While QUEST and GUIDE employ statistical significance tests for contingency tables, CTree applies permutation tests in a conditional inference framework and MOB uses fluctuation tests based on central limit theorems for the parameter estimators. All these approaches have been shown to work well for various situations, however, the relative (dis)advantages of the testing strategies have not yet been investigated and compared in detail. Therefore, in this paper the focus will be put on that first step of tree algorithms, i.e., the task of selecting the best split variable in a given (sub)sample. In particular, the approach of the GUIDE algorithm is compared to the one of the CTree algorithm and the MOB algorithm by investigating the building blocks of their testing strategies in which they differ: (1) Variation of the goodness-of-fit measure for the response: using residuals or full model scores. (2) Dichotomization of these residuals or scores. (3) Categorization of possible split variables. Apart from these three main factors further aspects such as the approximation of the null distribution (conditional vs. unconditional) or the type of test statistic (maximally selected vs. sum of squares) will be considered as well. For this purpose, a unifying framework for testing strategies in unbiased model-based tree algorithms is presented such that each of the three strategies in GUIDE, CTree, and MOB can be obtained by a specific combination of the available building blocks. This allows to systematically vary the building blocks and assess the power of the resulting inference procedure. Moreover, it is investigated whether the performance of the inference impacts the performance of the trees differently under pre- vs. post-pruning.

In many of the considered scenarios the choice of goodness-of-fit measure heavily influences the performance of testing strategies. In particular, using model scores leads to overall clearly better results than employing residuals only. Moreover, the original values of the goodness-of-fit measure are preferred over dichotomized versions of them. Also regarding the effects of categorizing possible split variables and the selection of a pruning strategy clear recommendations can be given based on the presented results.

The remainder of this manuscript is structured as follows: Section 3.2 reviews unbiased tree models, starting with the general algorithmic idea (Section 3.2.1) followed by the specific algorithms CTree (Section 3.2.2), MOB (3.2.3), and GUIDE (3.2.4), before possible pruning strategies (3.2.5) are discussed. In Section 3.3 the unifying framework for testing strategies
in unbiased model-based recursive partitioning algorithms is presented. The setting for the simulation study is introduced in Section 3.4 and the results are illustrated and discussed in Section 3.5.

3.2 Unbiased recursive partitioning

3.2.1 Generic algorithm

The basic idea of building a regression tree model is to partition the data into smaller and more homogeneous subgroups based on a set of covariates. Various tree algorithms have been developed, following essentially the same general structure, employing the covariates as split variables in the tree induction. Starting at the root of the tree, pertaining to the full available data sample, the algorithms proceed in the following steps:

1. For the current sample a (possibly simple) model is fitted by optimizing some objective function (or loss function) that reflects the goodness of fit.
2. Among all available split variables one is selected as the split variable, choosing the split point such that goodness of fit is maximized in the resulting subgroups.
3. Steps 1 and 2 are repeated within each subgroup until some stopping criterion is attained.

The term “model” is used here in a very broad sense and encompasses not only least-squares or maximum-likelihood models but also simple constant fits such as means or average proportions. Therefore, depending on the type of model, the employed objective function can for example be the sum of deviations from a typical/average value, but it can also be based on a model for a response along with potential split variables. For instance, different types of residuals (or the signs thereof) can be employed (see e.g., Loh, 2002) as well as rank sums or logrank scores (as in Hothorn et al., 2006a).

As explained in Section 3.1 the considered tree algorithms CTree, MOB, and GUIDE first select the split variable and then, in a separate step, the split point. For the first step of selecting a split variable they all apply statistical tests following the same basic strategy:

1. To capture how the objective function changes with the observations in the current subgroup, a disaggregated, observation-wise goodness-of-fit measure is obtained. More formally, this is an $N \times K$ matrix where $N \in \mathbb{N}$ is the number of observations and $K \in \mathbb{N}$ the number of goodness-of-fit measures per observation.
2. The dependency or association of this goodness-of-fit matrix with each possible split variable $Z_j, j \in \{1, \ldots, J\}$, is assessed using some suitable test statistic. The corresponding $p$-values allow for a comparison of all $J$ split variables on a standardized or unified scale and in that way for an unbiased split variable selection.
3. The split variable corresponding to the smallest $p$-value – and thus the highest influence on the goodness of fit of the model – is selected for splitting the data into subgroups.

As an example – and explained in more detail below – consider a linear regression tree. Thus, a linear regression model is fitted in each subgroup, minimizing the residual sum of squares as the aggregated goodness-of-fit measure. The corresponding observation-wise goodness-of-fit measure can be given by the residuals or the scores (gradient contributions). Analogously, the log-likelihood and corresponding score function could be used.
While this basic approach is the same for the GUIDE, CTree, and MOB algorithms, they differ in their strategies on how to calculate test statistics. In order to point out these specific characteristics in Section 3.3 the strategies of the three tree algorithms are first explained in more detail in Sections 3.2.2, 3.2.3, and 3.2.4.

3.2.2 CTree

The CTree algorithm (Hothorn et al., 2006b) is based on the idea of providing non-parametric regression tree models in a conditional inference framework by applying permutation tests. To select a split variable it is tested whether there is any association between the transformed response $h(Y)$ and each possible transformed split variable $g(Z_j)$, $j = 1, \ldots, J$. The only requirement for the function $h$ is to depend on $Y$ in a permutation-symmetric way but this encompasses ranks, scores, indicator functions, etc. and can also be multidimensional. For a numeric response the identity function $h(Y) = Y$ is a common choice while a categorical response can be mapped to a unity vector by an indicator function $h(Y) = (0, \ldots, 1, \ldots, 0)^T$. Alternatively, the function $h$ can capture location and scale of $Y$ via $h(Y) = (Y, (Y - \bar{Y})^2)^T$. If a parametric model is fitted to the response $Y$ with some covariate(s) $X$ employed as regressor variable(s), then a model-based transformation $h(Y) = s(Y, X, \hat{\beta})$ can be used, e.g., the residuals in a linear model with regression coefficients $\beta$. Moreover, $s$ can be the score function pertaining to the objective function (or loss function) $\ell$:

$$s(Y, X, \beta) = \frac{\partial \ell(Y, X, \beta)}{\partial \beta} \hspace{1cm} (3.1)$$

The estimate of the model parameters $\hat{\beta}$ is obtained by optimizing the sum of the objective function $\ell$, aggregated over all observations. This framework includes many different M-type estimators as special cases, including maximum likelihood and ordinary least squares estimation. For a $K$-dimensional parameter $\beta$ the score function evaluated for the $i$-th observation $s(y_i, x_i, \beta)$ is also a $K$-dimensional vector, i.e., the gradient contribution of the $i$-th observation. Thus, the $N \times K$-matrix consisting of these scores or gradient contributions for all $i = 1, \ldots, N$ observations is a natural candidate for the observation-wise goodness-of-fit measure as described in the previous Section 3.2.1.

Similarly, different types of functions can be chosen for the influence function $g$ depending on a possible split variable $Z$. A simple choice in case of $Z$ being a numeric variable is again the identity function $g(Z) = Z$. For categorical variables $g$ can also map its values to the corresponding unity vectors by an indicator function $g(Z) = (0, \ldots, 1, \ldots, 0)^T$.

To test for independence of $h(Y)$ and $g(Z)$ CTree calculates a linear association test statistic, following the framework of Strasser and Weber (1999). The corresponding conditional expectation and covariance given all permutations of the response variable can be calculated and used to standardize the test statistic. This standardized statistic has an asymptotic normal distribution which is in fact multivariate if either of the transformation $h(Y)$ and/or $g(Z)$ is multivariate. The actual test is carried out by mapping this standardized statistic to the real line either by taking the absolute maximum or using a quadratic form – with $p$-values being computed from the analogous transformation of the normal distribution (see Hothorn et al., 2006a, for a hands-on introduction and Appendix 3.7.1 for more details on the linear test statistic).

If both variables $Y$ and $Z$ are numeric the default independence test corresponds to a Pearson correlation test. For one numeric and one categorical variable essentially a one-way ANOVA (analysis of variance) is employed while for two categorical variables a $\chi^2$ test is performed. Thus, in the general CTree framework many types of tests can be specified by selecting suitable transformations $g$ and $h$. While originally conceived for nonparametric models, it is
3.2 Unbiased recursive partitioning

easy to adapt CTree to model-based testing and recursive partitioning by choosing a model-based \( h \) transformation as argued above (see the concrete examples in Zeileis and Hothorn, 2013; Seibold, Zeileis, and Hothorn, 2016).

3.2.3 MOB

In contrast to CTree the MOB algorithm (Zeileis et al., 2008) was explicitly designed for a model-based goodness-of-fit measure in order to embed parametric models into a regression tree framework. Thus, MOB is based on an objective/loss function \( \ell \) and corresponding score function \( s \). The original paper considered generalized linear models (GLMs) and survival regression models but subsequently various other models have been applied as well, including beta regression (Grünn, Kosmidis, and Zeileis, 2012), psychometric item response theory models (Strobl, Kopf, and Zeileis, 2015), or mixed effects models (Fokkema, Smits, Zeileis, Hothorn, and Kelderman, 2018). But just like CTree can be applied to parametric models, MOB conversely also encompasses simple regression and classification trees, e.g., by choosing an intercept-only model.

For selecting a split variable MOB employs a score-based test that relies on the central limit theorem for the parameter estimate \( \hat{\beta} \). The test assesses whether the scores – when ordered by the potential split variable \( Z \) – fluctuate randomly around their zero mean or differ systematically in certain subgroups. The latter would indicate a parameter instability that could be captured by fitting separate models (optimizing \( \ell \)) in the resulting subgroups. In case of a numeric split variable \( Z \) both the score-based statistic and the partitioned objective function \( \ell \) are maximized over all possible splits in \( Z \) (subject to certain minimal subgroup size constraints). Unlike the CTree framework, MOB relies on classical unconditional inference. For more details see Zeileis and Hornik (2007) and Appendix 3.7.1.

3.2.4 GUIDE

Building on earlier work for the QUEST algorithm (Loh and Shih, 1997), Loh (2002) proposed the GUIDE algorithm blending trees with parametric regression models and encompassing simpler classification and regression trees as special cases. Thus, linear regression models could be fitted in the nodes of a tree as well as constant fits such as simple mean response \( \hat{\beta} = \bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i \). The tests for selecting the split variable are then based on the corresponding residuals, e.g., for a simple linear regression (as used in Loh, 2002):

\[
r(Y, X, \hat{\beta}) = Y - \hat{\beta}_0 - \hat{\beta}_1 \cdot X.
\]

In subsequent work other models together with an appropriate choice of residuals have been applied such as in regression trees for longitudinal and multiresponse data (Loh and Zheng, 2013), quantile regression models (Chaudhuri and Loh, 2002), and proportional hazards modeling via Poisson regression (Loh, He, and Man, 2015), among others.

To construct a statistical test two additional transformations are carried out: (1) The residuals \( r(Y, X, \hat{\beta}) \) are dichotomized at zero, yielding an indicator for positive vs. negative residuals. (2) Each possible split variable \( Z \) is categorized, i.e., unless \( Z \) is already categorical it is split at its quartiles into four bins. Subsequently, a \( \chi^2 \) test of independence is performed for the dichotomized residuals and each categorized/categorical split variable. After choosing the split variable showing the highest dependency by yielding the lowest \( p \)-value, the split point minimizing the overall goodness-of-fit measure is selected. Note that the split point in numeric variables \( Z \) is searched over all possible splits, not just the four bins that were constructed for the \( \chi^2 \) test. More details on the applied test statistic can be found in Appendix 3.7.1.
Table 3.1. Default combinations of fitted model type, test type and pruning strategy for the algorithms CTree, MOB, and GUIDE.

<table>
<thead>
<tr>
<th></th>
<th>Fit</th>
<th>Test</th>
<th>Pruning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTree</td>
<td>Non-parametric</td>
<td>Conditional inference</td>
<td>Pre</td>
</tr>
<tr>
<td>MOB</td>
<td>Parametric</td>
<td>Score-based fluctuation</td>
<td>Pre (or post with AIC/BIC)</td>
</tr>
<tr>
<td>GUIDE</td>
<td>Parametric</td>
<td>Residual-based $\chi^2$</td>
<td>Post (cost-complexity pruning)</td>
</tr>
</tbody>
</table>

3.2.5 Pruning

To avoid overfitting recursive partitioning algorithms need to assure that trees do not grow too large. Apart from certain minimal subgroup size or maximal tree depth constraints this is classically accomplished by so-called “pruning” approaches. As the three tree algorithms considered here (CTree, MOB, and GUIDE) differ in their default choice of pruning approach, we briefly discuss these here. However, as all three algorithms can in principle be combined with any of the pruning approaches this is done only relatively briefly.

The classical CART algorithm (Breiman et al., 1984) proposed to first grow a large tree and then prune those splits in the tree that did not increase predictive performance in a cross-validation. This is also known as post-pruning (after growing the initial tree) and more specifically cost-complexity pruning.

In the unbiased recursive partitioning literature this post-pruning approach is also used frequently (e.g., in Loh and Vanichsetakul, 1988; Loh and Shih, 1997; Kim and Loh, 2001; Loh, 2002) and the $p$-values from the association tests are only employed for selecting the split variable on a unified scale. However, Hothorn et al. (2006b) proposed to also use these $p$-values for a so-called pre-pruning strategy which stops growing the tree as soon as no significant association can be found in a given subgroup. This approach is the default in CTree and also in MOB. However, Zeileis et al. (2008) also pointed out that a natural strategy for post-pruning in model-based partitioning is to use information criteria such as AIC (Akaike information criterion) or BIC (Bayes information criterion), following the ideas of Su, Wang, and Fan (2004).

Clearly, for inference-based pre-pruning it is crucial that the association tests employed for split selection work well as statistical significance test, i.e., conform with their nominal size and have high power. In contrast, when using post-pruning (either based on cross-validation or information criteria) it might not be as crucial that the significance test works well and has high power.

Due to these considerations we first evaluate the significance tests underlying CTree, MOB, and GUIDE by themselves, i.e., without growing an actual tree in combination with a pruning strategy. Subsequently we combine the tests with a cost-complexity post-pruning approach in order to assess whether shortcomings of the tests are mitigated by pruning.

3.3 Unifying framework

Each of the algorithms CTree, GUIDE, and MOB can be characterized by its combination of the type of model fits, tests, and pruning strategy employed to grow the tree. Table 3.1 provides an overview of the default combinations. However, as discussed above, subsequent publications have emphasized that all three algorithms can be combined with different model fitting approaches and to some degree different pruning strategies have been explored as well. Thus,
Table 3.2. Testing strategies of CTree, MOB, and GUIDE with the corresponding setting of the building blocks in the unifying framework and the type of test statistic.

<table>
<thead>
<tr>
<th></th>
<th>Scores</th>
<th>Dichotomization</th>
<th>Categorization</th>
<th>Statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTree</td>
<td>Model scores</td>
<td>–</td>
<td>–</td>
<td>Sum of squares</td>
</tr>
<tr>
<td>MOB</td>
<td>Model scores</td>
<td>–</td>
<td>–</td>
<td>Maximally selected</td>
</tr>
<tr>
<td>GUIDE</td>
<td>Residuals</td>
<td>✓</td>
<td>✓</td>
<td>Sum of squares</td>
</tr>
</tbody>
</table>

the class of tests employed for the unbiased splitting variable selection forms the core of each of the algorithms: conditional inference (CTree) vs. score-based fluctuation tests (MOB) vs. residual-based \( \chi^2 \) tests (GUIDE). Therefore, we consider a standard class of model fits (namely, linear regression trees) and investigate the relative advantages and disadvantages of the tests themselves (which are essential to pre-pruning) as well as the combination with post-pruning. Thus, subsequently the names CTree, MOB, and GUIDE distinguish the test-based variable and split selection rather than the entire algorithm with all default settings.

3.3.1 Building blocks of testing strategies

Even though CTree, MOB, and GUIDE differ in the specific tests they apply, their approaches for split variable selection follow the same basic structure as explained in Section 3.2.1. In fact, the tests can be embedded in a unifying conceptual framework that yields the different tests by combining various building blocks. These mostly differ in the way the model for the dependent variable on the one hand and the splitting variables on the other are prepared or transformed:

- **Goodness-of-fit measure:**
  Different variations of the disaggregated, observation-wise goodness-of-fit measure of the model for the response \( Y \) and possible regressors \( X \) can be considered. Either residuals \( r(Y, X, \hat{\beta}) \) can be used as proposed for GUIDE or model scores \( s(Y, X, \hat{\beta}) \) as proposed for MOB. All three algorithms can, in principle, use both goodness-of-fit measures though, which is probably brought out most clearly in the CTree algorithm that explicitly allowed for different transformations \( h(Y) \) in its original description already.

- **Dichotomization of residuals/scores:**
  Rather than testing independence between the split variables and the residuals/scores themselves, it is possible to dichotomize residuals/scores at 0 so that only their signs are assessed (as proposed for GUIDE).

- **Categorization of split variables:**
  Similarly, the split variables can also be categorized (for testing only). This was proposed for GUIDE, employing binning at the quartiles yielding four categories of approximately equal size.

The three algorithms combine these building blocks in different ways as shown in Table 3.2: When applying CTree for unbiased model-based recursive partitioning it has been suggested to use the model scores without dichotomization and assess their association with the untransformed split variables using a conditional inference test. This is similar to a squared correlation test statistic. MOB also employs the scores without dichotomization and maximally selects a score statistic over all potential split points in the split variable. GUIDE employs the
dichotomized residuals and assesses their association with the categorized split variable in a classical (unconditional) \( \chi^2 \) test.

But the building blocks could be easily recombined to yield new types of tests. For example, in the GUIDE approach, a one-way ANOVA can be used for assessing the association of the residuals (without dichotomization) with the categorized split variable. Or alternatively, a multivariate one-way ANOVA can be used for the non-dichotomized scores as opposed to the residuals etc.

Note that there are further differences in the testing strategies between the three algorithms, e.g., using conditional vs. unconditional approximations of the null distributions. However, this difference has relatively little influence compared to the other building blocks considered in detail here. Moreover, both similarities and relative differences between these approaches have been previously discussed, e.g., in Hothorn and Zeileis (2008) and Zeileis and Hothorn (2013).

### 3.3.2 Linear model tree

To focus on the unified testing framework, as described in the previous section, we employ the same model fits for all three algorithms. To do so, we employ linear regression models because it is such a basic and widely used model and linear model trees were the leading illustrations in both the original MOB (Zeileis et al., 2008) and GUIDE (Loh, 2002) papers. However, the conclusions drawn from this example also hold for many other model types.

To fix notation, we consider the following models for the simulation study in Section 3.4:

\[
Y = \beta_0 + \beta_1 \cdot X + \epsilon \tag{3.3}
\]

with response variable \( Y \), regressor variable \( X \), and error term \( \epsilon \). In particular, in the investigated tree models the coefficients \( \beta_0 \) and \( \beta_1 \) can depend on the possible split variables \( Z_j, j = 1, \ldots, J \), such that

\[
Y = \beta_0(Z_1, \ldots, Z_J) + \beta_1(Z_1, \ldots, Z_J) \cdot X + \epsilon. \tag{3.4}
\]

This model is fitted, as usual, by ordinary least squares (OLS) to the observations in each notation, yielding the parameter estimates \( \hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1)^\top \). To keep notation simple we present the following equations for the root node with all observations \( \{(y_i, x_i)\}_{i=1}^N \) and \( N \in \mathbb{N} \). In subsequent nodes the same equations are used but just for a smaller subgroup.

The aggregated goodness-of-fit measure (or objective or loss function) in OLS estimation is the sum of squared residuals: \( \sum_{i=1}^N \ell(y_i, x_i, \beta_0, \beta_1) \) where

\[
\ell(y_i, x_i, \beta_0, \beta_1) = r(y_i, x_i, \beta_0, \beta_1)^2 \tag{3.5}
\]

is the squared residual which is defined as

\[
r(y_i, x_i, \beta_0, \beta_1) = y_i - \beta_0 - \beta_1 \cdot x_i. \tag{3.6}
\]

These residuals can also intuitively be used as the corresponding disaggregated observationwise goodness-of-fit measure. Another natural candidate for this is the score function for the \( i \)-th observation:

\[
\ell(y_i, x_i, \beta_0, \beta_1) = -2 \cdot r(y_i, x_i, \beta_0, \beta_1) \cdot (1, x_i)^\top \tag{3.7}
\]

Thus, up to a constant scaling factor of \( -2 \) (that could also be omitted) the first component of the scores is in fact the residual. However, it is complemented by a second component that
3.4 Simulation setting and evaluation

In this simulation study, two different scenarios are considered for the linear model trees presented in Section 3.3.2. First, the underlying tree structure based on which the data is generated is a stump, i.e., a tree with only one split (“stump” scenario, see Figure 3.1). By keeping the tree so simple we can focus on the testing strategy only giving focus to their power in terms of selecting the correct split variable.

In the second scenario, the true tree structure contains two splits in two different variables yielding a tree with three terminal nodes (“tree” scenario, see Figure 3.2). It employs the same basic structure as the first scenario but simply adds another split. This allows to evaluate the power of the three testing strategies in a more complex setting, in combination with using a post-pruning strategy.

3.4.1 Data generating process

“Stump” scenario

Each generated data set consists of the response and regressor variables, one true split variable and nine noise split variables as listed in Table 3.3 together with the corresponding distributions.

Table 3.3. Variables included in the data generating process as used for the “stump” scenario. In the “tree” scenario, $Z_2$ is also a true split variable, not only $Z_1$, and hence $\beta_0$ and $\beta_1$ are functions $\beta_{k-1}(Z_1, Z_2)$, $k = 1, 2 = K$.

<table>
<thead>
<tr>
<th>Name</th>
<th>Notation</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Variables:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Response</td>
<td>$Y$</td>
<td>$= \beta_0(Z_1) + \beta_1(Z_1) \cdot X + \epsilon$</td>
</tr>
<tr>
<td>Regressor</td>
<td>$X$</td>
<td>$U([-1, 1])$</td>
</tr>
<tr>
<td>Error</td>
<td>$\epsilon$</td>
<td>$N(0, 1)$</td>
</tr>
<tr>
<td>True split variable</td>
<td>$Z_1$</td>
<td>$U([-1, 1])$</td>
</tr>
<tr>
<td>Noise split variables</td>
<td>$Z_2, \ldots, Z_{10}$</td>
<td>$U([-1, 1])$ or $N(0, 1)$ (alternating)</td>
</tr>
<tr>
<td><strong>Parameters/functions:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intercept</td>
<td>$\beta_0$</td>
<td>0 or $\pm \delta$</td>
</tr>
<tr>
<td>Slope</td>
<td>$\beta_1$</td>
<td>1 or $\pm \delta$</td>
</tr>
<tr>
<td>True split point</td>
<td>$\xi$</td>
<td>$\in {0, 0.2, 0.5, 0.8}$</td>
</tr>
<tr>
<td>Effect size</td>
<td>$\delta$</td>
<td>$\in {0, 0.1, 0.2, \ldots, 1}$</td>
</tr>
</tbody>
</table>
The location parameter \( \mu \) of the normally distributed response variable \( Y \) depends linearly on the regressor variable \( X \). Moreover, the intercept \( \beta_0 \) and/or the slope parameter \( \beta_1 \) can depend on the true split variable \( Z_1 \). More specifically, three different variations are considered for the coefficients \( \beta_0 \) and \( \beta_1 \):

1. The intercept \( \beta_0 \) varies depending on \( Z_1 \) while \( \beta_1 \) is fixed (at 1).
2. The slope coefficient \( \beta_1 \) varies depending on \( Z_1 \) while \( \beta_0 \) is fixed (at 0).
3. Both coefficients \( \beta_0 \) and \( \beta_1 \) vary depending on \( Z_1 \).

See Figure 3.1 for an illustration. More precisely, the coefficient that changes (\( \beta_{k-1} \) with \( k = 1, 2 = K \)) switches between two values at the split point \( \xi \):

\[
\beta_{k-1}(Z_1) = \begin{cases} 
-\delta \cdot (-1)^{k-1} & \text{if } Z_1 \leq \xi \\
+\delta \cdot (-1)^{k-1} & \text{if } Z_1 > \xi 
\end{cases}
\]

In that way, the type of variation is the same for \( \beta_0 \) and \( \beta_1 \), however, in opposite directions.

**“Tree” scenario**

For the second simulation scenario (“tree” scenario, see Figure 3.2) the same basic setting with the same variables as in the “stump” scenario is applied (see Table 3.3). However, not only \( Z_1 \) but
3.4 Simulation setting and evaluation

Also $Z_2$ is used as a true split variable following a uniform distribution on $[-1, 1]$. Therefore, an additional split is preferred yielding a tree with three terminal nodes. The first split (in $Z_2$, at $\xi$) induces a change in the slope parameter $\beta_1$ while the second split (in $Z_1$, also at $\xi$) corresponds to a change in the intercept $\beta_0$. Hence, contrary to the “stump” scenario where three variations are considered for the parameters $\beta_0$ and $\beta_1$, only this one variation with both parameters varying is investigated for the “tree” scenario. In particular, the parameters depend on the split variables in the following way:

$$\beta_0(Z_1, Z_2) = \begin{cases} 0 & \text{if } Z_2 \leq \xi \\ -\delta & \text{if } Z_1 \leq \xi \land Z_2 > \xi \\ +\delta & \text{if } Z_1 > \xi \land Z_2 > \xi \end{cases}$$

$$\beta_1(Z_1, Z_2) = \beta_1(Z_2) = \begin{cases} +\delta & \text{if } Z_2 \leq \xi \\ -\delta & \text{if } Z_2 > \xi \end{cases}$$

3.4.2 Evaluation

The testing strategies are evaluated over a stepwise increasing effect size $\delta$, on 100 replications per step each consisting of 250 observations. To compare the performance of the evaluated testing strategies for each step in the “stump” scenario the following criteria are considered: the $p$-values pertaining only to the true split variable $Z_1$; and the proportion of replications for which the $p$-value of $Z_1$ is the lowest and significant at 5% level (i.e., where $Z_1$ would be selected for splitting in a pre-pruning approach), denoted by the “selection probability”. In that way, the power of the considered statistical tests can be compared as they are all applied as significance tests answering two questions at once: (1) Should a split be performed at all? (2) If so, in which variable? For the first question the $p$-values regarding all available split variables are compared to a predefined level of significance $\alpha = 0.05$. Only if the smallest $p$-value is smaller than $\alpha$ a split is performed and the split variable corresponding to this $p$-value is selected.
For the “tree” scenario the adjusted Rand index (ARI) is calculated as a measure of similarity between the true tree structure and the fitted model tree.

As explained before, the aim of this simulation study is to investigate the effects of each particular building block of the unifying framework presented in Section 3.3 rather than the whole testing strategies. Therefore, other combinations as presented in Table 3.2, hence adapted versions of GUIDE, CTREE, and MOB are evaluated as well as their original versions.

3.5 Results

In the following we first investigate the properties of the testing strategies in the “stump” scenario, focusing on the tests’ p-values for the true split variable \( Z_1 \) and its corresponding selection probability (i.e., the association of \( Y \) and \( Z_1 \) being significant at 5% level and having the lowest p-value among all split variables). Section 3.5.1 begins by highlighting the importance of using full model scores vs. residuals only before Section 3.5.2 considers all building blocks (scores vs. residuals, dichotomization of these, and categorization of the split variable). Subsequently, the “tree” scenario is employed to investigate how the performance of the tests affects growing the trees overall. This is evaluated using the adjusted Rand index for trees grown by pre-pruning and cost-complexity post-pruning (Section 3.5.3).

3.5.1 “Stump” scenario: Residuals vs. full model scores

A crucial difference between the testing strategies of CTree/MOB and GUIDE is the difference between using only the residuals vs. the full model scores. In the literature on structural change tests it is well-established that residual-based tests can only capture parameter differences that affect the conditional mean (see e.g., Ploberger and Krämer, 1992, and further discussion in Section 3.6). Hence we compare the CTree, MOB, and GUIDE algorithms – all three using the default specification as shown in Table 3.2 – and additionally consider a new GUIDE flavor, denoted GUIDE+scores, that uses dichotomized scores rather than residuals.

In Figure 3.3 the performances of the testing strategies are represented by the corresponding selection probability, i.e., a significance level is incorporated. (Appendix 3.7.3 shows that the same qualitative conclusions can be drawn when the significance level is not included.) For a split point at the median (\( \xi = 0 \), top row) all testing strategies perform similarly well as long as the intercept varies (left and right panel) with CTree and MOB being only slightly ahead. However, for the split point at the 90% quantile \( \xi = 0.8 \) (bottom row) the performance of all of the applied strategies decreases. While both GUIDE versions struggle to detect the correct split variable even for a high effect size, MOB is clearly ahead with CTree leading to the second best results. This advantage of MOB over CTree is mainly due to the abrupt shift in the model parameters \( \beta_0 \) and \( \beta_1 \) and turns into an advantage of CTree over MOB for a smooth transition with continuously-changing parameters (see Appendix 3.7.4).

However, in the scenario where only \( \beta_1 \) (but not the intercept \( \beta_0 \)) is affected by the split, the residual-based GUIDE approach has no power at all even for a true split at the median (top middle panel). It is easily possible, though, to substantially mitigate this problem by using scores (sensitive to changes in all parameters) rather than residuals only (sensitive to changes in the conditional mean). The remaining difference between MOB/CTree and GUIDE+scores is due to dichotomizing the scores at zero and due to categorizing the split variables which are investigated in more detail in the following section.
3.5 Results

Figure 3.3. Selection probability of the true split variable $Z_1$ for testing strategies CTree, MOB, GUIDE, and GUIDE+scores in the “stump” scenario. Probabilities are estimated over increasing effect size $\delta$ with 100 replications of 250 observations per step. The true split point in $Z_1$ is either the median 0 (top) or the 90% quantile 0.8 (bottom) for either varying intercept $\beta_0$ (left), varying slope $\beta_1$ (middle), or both coefficients varying (right).

Table 3.4. Available levels of the building blocks.

<table>
<thead>
<tr>
<th>Building block</th>
<th>Levels</th>
<th>Transformation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Residuals vs. scores</td>
<td>residuals</td>
<td>$h(Y) = r(Y, X, \hat{\beta})$</td>
</tr>
<tr>
<td>(h transformation)</td>
<td>scores</td>
<td>$h(Y) = s(Y, X, \hat{\beta})$</td>
</tr>
<tr>
<td>Dichotomization</td>
<td>yes</td>
<td>$\mathbb{1}_{[0,\infty)}(h)$</td>
</tr>
<tr>
<td>(of h)</td>
<td>no</td>
<td>$h$ without dichotomization</td>
</tr>
<tr>
<td>Categorization</td>
<td>cat</td>
<td>$g(Z) = (0, 1, 0, 0)^\top$</td>
</tr>
<tr>
<td>(g transformation)</td>
<td>max</td>
<td>indicating the assigned bin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$g(Z) = (0, \ldots, 0, 1, \ldots, 1)^\top$</td>
</tr>
<tr>
<td></td>
<td>lin</td>
<td>indicating the potential split point</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$g(Z) = Z$</td>
</tr>
</tbody>
</table>

3.5.2 “Stump” scenario: Full factorial analysis of building blocks

To investigate the impact of each of the building blocks separately the most general case of the “stump” scenario where the intercept and slope parameter are both varying is considered. In this evaluation all possible combinations of the building blocks have been included. The different levels of each of the three building blocks are listed in Table 3.4 where $h$ and $g$ refer to the transformation functions applied to the response $Y$ or a split variable $Z$ respectively, both as described in Section 3.2.2.
In the case of categorization, the split variable \(Z\) is binned at the quartiles. This corresponds to a four-dimensional 0/1 transformation function \(g(Z)\) that indicates into which of the bins each observation falls. Maximum selection across potential split points in a variable \(Z\) also corresponds to a multivariate 0/1 transformation function \(g(Z)\). However, in this case for each potential split point an indicator is used that is 0 before and 1 after the respective split point. (See also Table 3.5 in Appendix 3.7.2 for a more detailed overview of all 12 combinations of the building blocks.)

Based on the results displayed in Figure 3.4 it can be stated that dichotomizing residuals/scores decreases the performance as it leads to higher \(p\)-values for the true split variable \(Z_1\) and thus to lower power in the settings considered. For a true split at the median (\(\xi = 0\), left panel) this effect is almost constant across the three types of categorization considered. However, in case of the true split point at the 90\% quantile \(\xi = 0.8\) (right panel), categorizing split variables increases the \(p\)-value of \(Z_1\) even more. Not surprisingly, maximum selection is most advantageous in this case (i.e., for a late abrupt shift) which is harder to detect based on a linear statistic. But overall it depends on the situation whether a linear or a maximum selection of a split variable leads to lower \(p\)-values.

Comparing the two panels suggests that a categorization weakens the performance of the tests unless the true split point is close to one of the breaks from the binning (as in the left panel). Moreover, while the effect of both transformations (dichotomization of residuals/scores and categorization of split variables) can be observed separately in Figure 3.4, combining them increases the negative impact remarkably.

As already shown in Section 3.5.1 the use of scores vs. residuals has a minor effect if there is a change in the intercept which is supported by the small differences between the dashed lines (scores) and the solid lines (residuals) in Figure 3.4.
Note that the effect size $\delta$ in the results from Figure 3.3 has been chosen so that $p$-values for the non-dichotomized tests are roughly comparable: $\delta = 0.3$ for the true split point at the median $\xi = 0$ (left) vs. a stronger effect of $\delta = 1$ for the true split point at the 90\% quantile $\xi = 0.8$ (right). Additional evaluations for varying effect size $\delta$ can be found in Appendix 3.7.5.

### 3.5.3 “Tree” scenario: Pre-pruning vs. post-pruning

So far the testing strategies underlying the different tree algorithms have only been considered as classical significance tests, i.e., in terms of power and $p$-values. However, one could argue that for a tree this is practically not really relevant - at least when combined with a post-pruning strategy such as cost-complexity pruning (Breiman et al., 1984). In the latter case it only matters that the relevant split variables have the lowest $p$-value among all potential split variables – but it is irrelevant whether this is significant or not. To investigate to which extent this is actually true we evaluate the different tree algorithms in the more complex “tree” scenario (see Figure 3.2): once with significance-based pre-pruning and once with cost-complexity post-pruning.

Recall that the true split structure is composed of splits in two different variables, both at the same split point $\xi$. First, the split in $Z_2$ changes the slope from $+\delta$ to $-\delta$. Second, the split in $Z_1$ changes the intercept in the negative slope group from $-\delta$ to $+\delta$. In the simulation the effect size $\delta$ is increased from 0 to 1 for different split points $\xi$ from 0 to 0.8. Here, the performance is not evaluated in terms of test properties but only in terms of tree properties, namely the adjusted Rand index (ARI) in comparison to the true partition of the data.

The top row of Figure 3.5 presents the results for pre-pruning which, not surprisingly, reflect the results from previous sections. Thus, CTree and MOB perform similarly and rather well as the underlying significance tests have good power properties. Only for late abrupt shifts the linear test statistics in CTree clearly have lower power than the maximally-selected statistics in MOB. Again, this picture would reverse for smooth changes instead of abrupt shifts. Compared to CTree/MOB both GUIDE flavors clearly perform worse as the underlying tests are less powerful: notably for the residual-based GUIDE which again has problems picking up the slope change associated with $Z_2$.

When switching from pre-pruning to post-pruning (bottom row of Figure 3.5) it is indeed shown that many of the problems stemming from the low power of the two GUIDE flavors are indeed mitigated. Thus, by first growing a large tree without stopping upon non-significance and then pruning back based on predictive performance instead substantially improves the fit of the two GUIDE flavors. However, using the residuals only in GUIDE still performs clearly worse compared to the GUIDE+scores. The latter is essentially on par with CTree and MOB when the true split matches one of the categorization bins (i.e., $\xi = 0$ and $\xi = 0.5$, respectively) while CTree/MOB still perform somewhat better in the other cases ($\xi = 0.2$ and $\xi = 0.8$).

In summary, there is clear support for the conventional wisdom that the power of the testing strategy in an unbiased tree algorithm is not so important when combined with post-pruning. However, there are limits to this. Consequently, when tests can be made more powerful – e.g., by using scores instead of residuals – then this improves the corresponding tree algorithm. Finally, the simulation also supports using pre-pruning in an unbiased tree algorithm when the underlying testing strategy also works well as a classical significance test.
Figure 3.5. Adjusted Rand index (ARI) for the testing strategies CTree, MOB, GUIDE, and GUIDE+scores in the “tree” scenario, once applying pre-pruning (top row) and once post-pruning (bottom row). The ARI is evaluated over increasing effect size \( \delta \) with 100 replications of 250 observations per step. Each panel represents a different value of the true split point \( \xi \) (0, 0.2, 0.5 and 0.8 from left to right, corresponding to the 50\%, 60\%, 75\%, and 90\% quantile).

3.6 Discussion

The testing strategies underlying the unbiased recursive partitioning algorithms CTree, MOB, and GUIDE have been embedded in a common inference framework, highlighting what the tests have in common and what sets them apart. Concerning the effects of the corresponding building blocks for the tests, three main conclusions can be drawn:

- **Goodness-of-fit measure**: Assessing all dimensions of a model via the full scores is to be preferred over assessing only a subset with the residuals. The former can substantially improve performance while leading only to minor deteriorations when it is not necessary.

- **Dichotomization of residuals/scores**: No scenarios could be found where this is beneficial and tests without dichotomization performed clearly better in several scenarios.

- **Categorization of split variables**: The effects of categorization are not so clear-cut and depend on the true data structure. If there are indeed abrupt shifts close to the breaks from the categorization, it works well. However, for splits close to the margins performance can deteriorate and a maximally-selected test is preferable. Finally, a linear statistic performs better for smooth rather than abrupt changes.

Thus, also when categorization as in GUIDE is used we would recommend to employ the non-dichotomized scores instead of the dichotomized residuals. Note that such a test corresponds to a multiple ANOVAs (for the score components) and is easily available in statistical software.
Moreover, in the R system the coin package provides a convenient toolbox that encompasses multivariate transformation functions \( h \) and/or \( g \).

Linear models as presented in this study have been chosen as they are highly relevant in practice, allow for simple illustrations, and theoretical insights are available for the testing strategies. However, the results can be easily extended to a wide variety of other models where the introduced building blocks can be applied in the same way. Specifically, it has been shown theoretically that certain changes in the parameters do not lead to shifts in the residuals (but in other components of the scores). Ploberger and Krämer (1992) showed that residual-based tests can detect a change in the parameters of a linear model only if it also causes a shift in the expected value \( \mathbb{E}(Y) \). This is not the case if changes are orthogonal to the mean regressor which in our case is \((1, 0)\)\(^\top\). Consequently, if only the slope \( \beta_1 \) but not the intercept \( \beta_0 \) changes, the shift is of type \((0, \delta)\)\(^\top\) and thus orthogonal to the mean regressor. Due to this residual-based tests as in GUIDE break down and do not have power to detect this. Note that this situation does not have to be rare in practice: Especially for binary regressor variables (e.g., as in treatment-subgroup interactions, see Doove, Dusseldorp, Van Deun, and Van Mechelen, 2014) it can easily occur (see Figure 2 in Loh et al., 2015, for an illustration). Testing strategies being capable of detecting all kinds of changes including those orthogonal to the mean are also employed in an extension of (generalised) linear model trees introduced by Seibold et al. (2019) which allows for selected model parameters to be kept fixed at a globally estimated value while only changes in the remaining parameters are considered for splitting.

Also in more general models, residual-based and score-based procedures are expected to perform equally well if all model parameters are highly correlated. But if parameters do change orthogonally this might again be missed when only considering residuals – and full model scores are typically easily available as the appropriate remedy. Note that the score function \( s \) can also simply be seen as a transformation of the response variable \( Y \) (and potentially regressors \( X \)) to a different space in order to allow for a well structured analysis of dependencies. This has been exploited in several tree-based approaches previously published in the literature, e.g., in Hothorn and Zeileis (2017) and Schlosser et al. (2019b). Similarly, it would be of interest to investigate a score-based version of the extended GUIDE algorithms beyond the linear model, e.g., in Loh and Zheng, 2013, Chaudhuri and Loh, 2002, and Loh et al., 2015.

**Computational details**

The applied implementation is based on the R package partykit (version 1.2.4) which is available on R-Forge at https://R-Forge.R-project.org/projects/partykit/. The code to reproduce the simulation study is available at https://eeecon.uibk.ac.at/~zeileis/news/power_partitioning/.

The functions ctree and mob provide an implementation of the two tree algorithms in their original form. For their adapted versions additional modifications have been applied within these functions allowing for a categorization of possible split variables and a dichotomization of scores. To evaluate the GUIDE algorithm a reimplementation of this algorithm has been built using the basic framework of ctree and mob.

**Acknowledgements**

Torsten Hothorn received funding from the Swiss National Science Foundation, grant number 200021_184603.
3.7 Appendix

3.7.1 Test statistics

This section provides further information on the test statistics applied in CTree, MOB, and GUIDE. Most of the presented details have been extracted from the original papers Hothorn et al. (2006b), Zeileis et al. (2008), and Loh (2002), however, notation is adapted to the main manuscript in order to allow for better comparison.

CTree

To measure the association of a response $Y$ and each possible split variable $Z_j$, $j = 1, \ldots, J$, the CTree algorithm applies a linear test statistic $T_j$ which is excerpted from Section 3.1. ("Variable selection and stopping criteria") of the original paper (Hothorn et al., 2006b) and is of the following form:

$$T_j(Y, Z_j, w) = \text{vec} \left( \sum_{i=1}^{N} w_i g_j(Z_{ji}) h(Y_i)^\top \right)$$

where $w$ are optional weights, $g_j : Z_j \rightarrow \mathbb{R}^P$ is a nonrandom transformation of the split variable $Z_j$ and the influence function $h : \mathcal{Y} \times \mathcal{Y}^N \rightarrow \mathbb{R}^Q$ depends on the response $Y = (Y_1, \ldots, Y_N)$ in a permutation symmetric way and is set to $h(Y_i) = s(Y_i, X_i, \hat{\beta})$ for a score-based approach. Moreover, by applying the “vec” operator the resulting $P \times Q$ matrix is converted into a $PQ$ column vector. Following Strasser and Weber (1999) the conditional expectation $\mu_j$ and covariance $\Sigma_j$ of a linear test statistic $T_j$ can be calculated and used to standardize an observed linear test statistic $t_j$ within a function $c$ mapping into the real line. For example, the maximum of the absolute values of the standardized linear statistic

$$c_{\text{max}}(t_j, \mu_j, \Sigma_j) = \max \left| \frac{(t_j - \mu_j)}{\text{diag}(\Sigma_j)} \right|$$

or a quadratic form

$$c_{\text{quad}}(t_j, \mu_j, \Sigma_j) = (t_j - \mu_j)\Sigma_j^+(t_j - \mu_j)^\top$$

can be considered where $\Sigma_j^+$ is the Moore-Penrose inverse of $\Sigma_j$. Since the asymptotic conditional distribution of a linear test statistic $T_j$ is a multivariate normal with parameters $\mu_j$ and $\Sigma_j$ (Strasser and Weber, 1999), the asymptotic distribution of $c_{\text{max}}$ is normal while the quadratic form $c_{\text{quad}}$ follows an asymptotic $\chi^2$ distribution. Based on this knowledge the corresponding $p$-values can be calculated easily.

MOB

The MOB algorithm employs an empirical fluctuation process $W_j$ to measure deviations of the model scores $s$ from zero with respect to an ordering induced by the possible split variable $Z_j$, $j = 1, \ldots, J$. As described in detail in Section 3.2. ("Testing for parameter instability") of the original paper (Zeileis et al., 2008) this process is of the following form:

$$W_j(t) = \hat{V}^{-1/2}N^{-1/2} \sum_{i=1}^{[Nt]} s_{\pi(Z_{ji})} \quad (0 \leq t \leq 1)$$

with the model scores $s_i = s(Y_i, X_i, \hat{\beta})$ being sorted by the possible split variable $Z_j$ by including the ordering permutation $\pi(Z_{ji})$. To scale this partial sum process an estimate $\hat{V}$ of the covariance matrix $\text{cov}(s(Y, X, \hat{\beta}))$ is included. Following Zeileis and Hornik (2007) $W_j(t)$ converges to a Brownian bridge $W^0$ under the null hypothesis of parameter stability. To obtain a test statistic
A scalar functional $\lambda(\cdot)$ capturing the fluctuation in the empirical process can be applied and the corresponding asymptotic distribution of $\lambda(W_j(t))$ can be obtained by employing the same functional to the limit process, i.e., $\lambda(W^0)$.

One possible and intuitive choice for a functional in order to assess instabilities over a numerical split variable is the following:

$$\lambda_{\text{supLM}}(W_j) = \max_{i = \hat{i}, \ldots, \hat{i}} \left( \frac{i}{N} \cdot \frac{N - i}{N} \right)^{-1} \left\| W_j \left( \frac{i}{N} \right) \right\|^2_2$$

where a minimal segment size $\hat{i}$ and then $\hat{i} = N - \hat{i}$ are used to define the interval $[\hat{i}, \hat{i}]$. Other possible functionals, for example also for categorical split variables, and more details, particularly on calculating the corresponding $p$-values can be found in Section 3.2. of the original paper (Zeileis et al., 2008).

GUIDE

The test statistic of the $\chi^2$ test as applied in the GUIDE algorithm is

$$X^2 = \frac{(O_{11} - E_{11})^2}{E_{11}} + \ldots + \frac{(O_{24} - E_{24})^2}{E_{24}}$$

where $O_{lm}$ are the observed frequencies of observations in a certain combination of dichotomized residuals ($l = 1, 2$) and categorized split variables ($m = 1, \ldots, 4$). $E_{lm}$ are the corresponding expected frequencies under independence $E_{lm} = \frac{O_{1l} \cdot O_{m1}}{N}$. Under the null hypothesis of independence the asymptotic distribution of $X^2$ is a $\chi^2$ distribution allowing for a straightforward calculation of $p$-values. If full model scores are used instead of residuals a sum of the test statistic over the number of distribution parameters is considered such that each summand corresponds to one column of the score matrix. For this case the degrees of freedom of the resulting $\chi^2$ distribution is then also the corresponding sum of degrees of freedom over the number of distribution parameters.

3.7.2 Combinations of building blocks applied

For the evaluation of all 12 possible combinations of the presented building blocks the original testing strategies CTree, MOB, and GUIDE have been applied together with modified versions of them. The employed versions and the corresponding combination of building blocks are listed in Table 3.5.

3.7.3 Significance level

In Section 3.5.1 the investigated testing strategies are compared based on the selection probability of $Z_1$, hence, the number of replications in which the true split variable $Z_1$ is detected with a $p$-value smaller than the predefined significance level $\alpha = 0.05$. This choice of measurement is due to the aim of investigating how well the tests perform as significance test. However, ignoring this significance level yields the same conclusions which can be seen when comparing Figures 3.6 and 3.3, both being based on the exact same evaluation, but in Figure 3.6 the proportion of replications in which the true splitting variable $Z_1$ is detected by showing the smallest $p$-value, but not necessarily smaller than the significance level, is illustrated.

3.7.4 Continuous “stump” scenario

In the simulation study presented in Sections 3.4 and 3.5 the intercept and slope parameters $\beta_0$ and $\beta_1$ are both either fixed or binary variables taking either the positive or the negative
Table 3.5. Testing strategies applied for all 12 factorial combinations of the building blocks. For each combination one of the three testing strategies CTree/MOB/GUIDE is applied, possibly after modifying the assessed variables as indicated (i.e., dichotomized or categorized).

<table>
<thead>
<tr>
<th>Residuals/Scores</th>
<th>Dichotomization</th>
<th>Categorization</th>
<th>Via testing strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>residuals</td>
<td>yes</td>
<td>cat</td>
<td>GUIDE (default)</td>
</tr>
<tr>
<td>residuals</td>
<td>yes</td>
<td>max</td>
<td>MOB (modified)</td>
</tr>
<tr>
<td>residuals</td>
<td>yes</td>
<td>lin</td>
<td>CTree (modified)</td>
</tr>
<tr>
<td>residuals</td>
<td>no</td>
<td>cat</td>
<td>MOB (modified)</td>
</tr>
<tr>
<td>residuals</td>
<td>no</td>
<td>max</td>
<td>MOB (modified)</td>
</tr>
<tr>
<td>residuals</td>
<td>no</td>
<td>lin</td>
<td>CTree (modified)</td>
</tr>
<tr>
<td>scores</td>
<td>yes</td>
<td>cat</td>
<td>GUIDE (modified)</td>
</tr>
<tr>
<td>scores</td>
<td>yes</td>
<td>max</td>
<td>MOB (modified)</td>
</tr>
<tr>
<td>scores</td>
<td>yes</td>
<td>lin</td>
<td>CTree (modified)</td>
</tr>
<tr>
<td>scores</td>
<td>no</td>
<td>cat</td>
<td>MOB (modified)</td>
</tr>
<tr>
<td>scores</td>
<td>no</td>
<td>max</td>
<td>MOB (default)</td>
</tr>
<tr>
<td>scores</td>
<td>no</td>
<td>lin</td>
<td>CTree (default)</td>
</tr>
</tbody>
</table>

Figure 3.6. Proportion of replications where the true split variable \(Z_1\) shows the smallest \(p\)-value, hence, where \(Z_1\) is selected without any significance level in the “stump” scenario. The testing strategies CTree, MOB, GUIDE, and GUIDE+scores are evaluated over increasing effect size \(\delta\) with 100 replications of 250 observations per step. The true split point in \(Z_1\) is either the median 0 (top) or the 90% quantile 0.8 (bottom) for either varying intercept \(\beta_0\) (left), varying slope \(\beta_1\) (middle), or both coefficients varying (right).
value of the effect size $\delta$. Therefore, these parameters are step functions for which the maximum selection employed in the testing strategy of MOB has shown to perform better than the linear selection in CTree. However, this changes in case of monotonous functions where CTree is advantageous as it is constructed to detect monotonous effects. To point out that the difference in performance of CTree and MOB depends on the type of effect the same setting for which the results are presented in Figure 3.3 is evaluated again but this time with the varying parameter(s) changing continuously. In particular, the parameters $\beta_0$ and $\beta_1$ are linear functions of the true split variable $Z_1$:

$$
\beta_{k-1}(Z_1) = \begin{cases} 
-\delta \cdot (-1)^{k-1} \cdot Z_1 \\
+\delta \cdot (-1)^{k-1} \cdot Z_1 
\end{cases}
$$

for $k = 1, 2 = K$. Additionally, a modified version of CTree employing a maximum selection, such as MOB, is evaluated (denoted by CTree+max). Looking at the selection probability illustrated in Figure 3.7 it can be observed that in this setting the original version of CTree is ahead while CTree+max and MOB perform almost equally well. Hence, the comparison of Figures 3.3 and 3.7 points out that it clearly depends on the type of effect whether the maximum selection (MOB, CTree+max) or the linear selection (CTree) is to be preferred.

### 3.7.5 Results for increasing effect size

**Dichotomization of residuals/scores**

To elaborate over an increasing effect size whether a dichotomization of the residuals/scores at zero leads to an improvement or a deterioration of performance, CTree and MOB are applied, once in their original version without dichotomization and once in an adapted version with a dichotomization of scores at zero (CTree+dich, MOB+dich). Moreover, they are compared to the adapted GUIDE version which includes all available scores (GUIDE+scores).
Figure 3.8. Selection probability of the true split variable $Z_1$ for testing strategies CTree, CTree+dich, MOB, MOB+dich, and GUIDE+scores in the “stump” scenario with both parameters varying. CTree+dich and MOB+dich refer to modifications of CTree and MOB applying a dichotomization of residuals/scores. Probabilities are estimated over increasing effect size $\delta$ with 100 replications of 250 observations per step. Each panel represents a different value of the true split point $\xi$ (0, 0.2, 0.5 and 0.8 from left to right corresponding to the 50%, 60%, 75%, and 90% quantile).

Figure 3.8 shows the effect of dichotomizing the score values over different values for the true split point $\xi$, however, all four situations lead to the same conclusions: Dichotomizing the score values decreases the selection probability of the true split variable $Z_1$, and hence reduces the power of the testing strategy.

Categorization of splitting variables

To investigate the effect of categorizing the possible splitting variables CTree and MOB are applied, once in their original version without a categorization and once in an adapted version with a categorization of the possible split variables (CTree+cat and MOB+cat). Moreover, they are also compared to GUIDE+scores which includes all available scores.

In Figure 3.9 the impact of categorizing split variables on the performance is illustrated by the selection probability of $Z_1$ over increasing effect size and for four different values of the true split point $\xi$. For both, CTree and MOB, it can be stated that overall they perform better in their original form without categorization. Only if the true split point $\xi$ is close to the quartiles used as breaks for the categorizations both versions lead to a selection probability of $Z_1$ (e.g., for CTree and $\xi = 0.5$).

Therefore, these results support the conclusions drawn in the main manuscript: Categorizing the values of the split variable does not lead to any advantages unless the true split point corresponds (or is at least close) to one of the quartiles used for the categorization. In most situations it even causes the power of the testing strategy to decrease.
Figure 3.9. Selection probability of the true split variable $Z_1$ for testing strategies CTree, CTree+cat, MOB, MOB+cat, and GUIDE+scores in the “stump” scenario with both parameters varying. Probabilities are estimated over increasing effect size $\delta$ with 100 replications of 250 observations per step. Each panel represents a different value of the true split point $\xi$ (0, 0.2, 0.5 and 0.8 from left to right corresponding to the 50%, 60%, 75%, and 90% quantile).
Chapter 4

Paper IIIa: Distributional Trees for Circular Data

Distributional Trees for Circular Data


*University of Innsbruck, Department of Statistics, Universitätsstr. 15, A-6020 Innsbruck, Austria
**University of Zurich, Epidemiology, Biostatistics and Prevention Institute, Hirschengraben 84, CH-8001 Zürich, Switzerland
***University of Innsbruck, Department of Atmospheric and Cryospheric Science, Innrain 52f, A-6020 Innsbruck, Austria

Abstract For probabilistic modeling of circular data the von Mises distribution is widely used. To capture how its parameters change with covariates, a regression tree model is proposed as an alternative to more commonly-used additive models. The resulting distributional trees are easy to interpret, can detect non-additive effects, and select covariates and their interactions automatically. For illustration, hourly wind direction forecasts are obtained at Innsbruck Airport based on a set of meteorological measurements.

Keywords: distributional trees, circular response, von Mises distribution.

4.1 Motivation

Circular data can be found in a variety of applications and subject areas, e.g., hourly crime rate in the social-economics, animal movement direction or gene-structure in biology, and wind direction as one of the most important weather variables in meteorology. Circular regression models were first introduced by Fisher and Lee (1992) and further extended by Jammalamadaka and Sengupta (2001) and Mulder and Klugkist (2017) among others. While most of the already existing approaches are built on additive regression models, we propose an adaption of regression trees to circular data by employing distributional trees.
4.2 Methodology

Distributional trees (Schlosser et al., 2019b) fuse distributional regression modeling with regression trees based on the unbiased recursive partitioning algorithms MOB (Zeileis et al., 2008) or CTree (Hothorn et al., 2006b). The basic idea is to partition the covariate space recursively into subgroups such that an (approximately) homogeneous distributional model can be fitted to the response in each resulting subgroup. To capture dependence on covariates, the association between the model’s scores and each available covariate is assessed using either a parameter instability test (MOB) or a permutation test (CTree). In each partitioning step, the covariate with the highest significant association (i.e., lowest significant p-value, if any) is selected for splitting the data. The corresponding split point is chosen either by optimizing the log-likelihood (MOB) or a two-sample test statistic (CTree) over all possible partitions.

In this study distributional trees are adapted to circular responses by employing the von Mises distribution, also known as “the circular normal distribution”. Based on a location parameter $\mu \in [0, 2\pi]$ and a concentration parameter $\kappa > 0$ the density for $y \in [0, 2\pi]$ is given by:

$$f_{\text{vM}}(y; \mu, \kappa) = \frac{1}{2\pi I_0(\kappa)} e^{\kappa \cos(y-\mu)}$$

(4.1)

where $I_0(\kappa)$ is the modified Bessel function of the first kind and order 0 (see, e.g., Jammalamadaka and Sengupta, 2001, for a more detailed overview).

In each subgroup maximum likelihood estimators $\hat{\mu}$ and $\hat{\kappa}$ are obtained by maximizing the corresponding log-likelihood $\ell(\mu, \kappa; y) = \log(f_{\text{vM}}(y; \mu, \kappa))$. The model scores are given by $s(y; \hat{\mu}, \hat{\kappa}) = (\partial_\mu \ell(\mu, \kappa; y), \partial_\kappa \ell(\mu, \kappa; y))$. In a subgroup of size $n$, evaluating the scores at the individual observations and parameter estimates $s(y_i; \hat{\mu}, \hat{\kappa})$ yields an $n \times 2$ matrix that can be employed as a kind of residual, capturing how well a given observation conforms with the estimated location $\hat{\mu}$ and precision $\hat{\kappa}$, respectively. Hence MOB or CTree can assess whether the scores change along with the available covariates. If so, by maximizing a partitioned likelihood the parameter instabilities are incorporated into the model. This procedure is repeated recursively until there are no significant parameter instabilities or until another stopping criterion is met (e.g., subgroup size or tree depth).

4.3 Application

Wind is a classical circular quantity and accurate forecasts of wind direction are of great importance for decision-making processes and risk management, e.g., in air traffic management or renewable energy production. This study employs circular regression trees to obtain hourly wind direction forecasts at Innsbruck Airport. Innsbruck lies at the bottom of a deep valley in the Alps. Topography channels wind along the west-east valley axis or along a tributary valley intersecting from the south. Hence, pressure gradients to which valley wind regimes react both west and east of the airport are considered as covariates along with other meteorological measurements at the airport (lagged by one hour), such as wind direction and wind speed at Innsbruck Airport. Note that in the meteorological context wind direction is defined on the scale [0, 360] degree and increases clockwise from North (0 degree).

Figure 4.1 depicts the resulting distributional tree, including both the empirical (gray) and fitted von Mises (red) distribution of wind direction in each terminal node. Based on the fitted location parameters $\hat{\mu}$, the subgroups can be distinguished into the following wind regimes: (1) Up-valley winds blowing from the valley mouth towards the upper valley (from east to west, nodes 4 and 5). (2) Downslope winds blowing across the Alpine crest along the intersecting valley towards Innsbruck (from south-east to north-west, nodes 7 and 8). (3) Down-valley winds
Figure 4.1. Fitted tree based on the von Mises distribution for wind direction forecasting. In each terminal node the empirical histogram (gray) and fitted density (red line) are depicted along with the estimated location parameter (red hand). The covariates employed are wind direction (degree), wind speed ($\text{ms}^{-1}$), and pressure gradients ($\text{dpressure; hPa}$) west and east of the airport, all lagged by one hour.
Paper IIIa: Distributional Trees for Circular Data

blowing in the direction of the valley mouth (from west to east, nodes 12, 14, and 15). Node 11 captures observations with rather low wind speeds that cannot be distinguished clearly into wind regimes and consequently are associated with a very low estimated concentration \( \hat{\kappa} \). In terms of covariates, the lagged wind direction (“persistence”) is mostly responsible for distinguishing the broad wind regimes listed above while the pressure gradients and wind speed separate between subgroups with high vs. low precision.

4.4 Discussion and outlook

Distributional trees for circular responses are established by coupling model-based recursive partitioning with the von Mises distribution. The resulting trees can capture nonlinear changes, shifts, and potential interactions in covariates without prespecification of such effects. This is particularly useful for modeling wind direction in mountainous terrain where wind shifts can occur due to turns of the pressure gradients along a valley.

4.4.1 Ensembles and random forests

A natural extension are ensembles or forests of such circular trees that can improve forecasts by regularizing and stabilizing the model. Random forests introduced by Breiman (2001a) average the predictions of an ensemble of trees, each built on a subsample or bootstrap of the original data. A generalization of this strategy is to obtain weighted predictions by adaptive local likelihood estimation of the distributional parameters (Schlosser et al., 2019b). More specifically, for each possibly new observation \( x \) a set of “nearest neighbor” weights \( w_i(x) \) is obtained that is based on how often \( x \) is assigned to the same terminal node as each learning observation \( y_i, i \in \{1, \ldots, n\} \).

The parameters \( \mu \) and \( \kappa \) are then estimated for each (new) observation \( x \) by weighted maximum likelihood based on the adaptive nearest neighbor weights:

\[
\argmax_{\mu, \kappa} \frac{1}{n} \sum_{i=1}^{n} w_i(x) \cdot \ell(\mu, \kappa; y_i).
\]

Therefore, the resulting parameter estimates can smoothly adapt to the given covariates \( x \) whereas \( w_i(x) = 1 \) would correspond to the unweighted full-sample estimates and \( w_i(x) \in \{0, 1\} \) corresponds to the abrupt splits from the tree.

4.4.2 Splits in circular covariates

In order to obtain more parsimonious and more stable trees another possible extension for circular covariates (with or without a circular response) is to consider their circular nature when searching the best split into two segments. In general, searching the best separation of a covariate into a “left” and “right” daughter node tries to maximize the segmented log-likelihood:

\[
\max \left( \sum_{y \in \text{left}} \ell(\hat{\mu}_1, \hat{\kappa}_1; y) + \sum_{y \in \text{right}} \ell(\hat{\mu}_2, \hat{\kappa}_2; y) \right)
\]

where \( \hat{\mu}_1, \hat{\kappa}_1, \hat{\mu}_2, \hat{\kappa}_2 \) are the estimated parameters of the von Mises distribution in the two daughter nodes. Searching a single split point \( \nu \) in a circular covariate \( \in [0, 2\pi) \) only considers linear splits into the intervals \( \text{left} = [0, \nu] \) and \( \text{right} = (\nu, 2\pi) \), thus enforcing a potentially unnatural separation at zero. This can be avoided by searching for two split points \( \nu \) and \( \tau \) considering a split into one interval \( \text{left} = [\nu, \tau] \) and its complement \( \text{right} = [0, \nu] \cup (\tau, 2\pi) \),
encompassing zero. The latter strategy is invariant to the (often arbitrary) definition of the direction at zero.

When one split point $\nu$ is sufficiently close to zero and the other $\tau$ sufficiently far away, a simple linear split typically suffices to capture such a split (as seen for the lagged wind direction in Figure 4.1). If both $\nu$ and $\tau$ differ clearly from zero, two linear splits should also lead to a reasonable (but less parsimonious) fit. However, if both $\nu$ and $\tau$ are rather close to zero, a linear split strategy might miss such a pattern.

The required test statistic to maximally select two split points simultaneously is straightforward to accommodate in the CTree framework by providing all binary indicators corresponding to the splits into left/right intervals. However, this will become increasingly slow for larger sample sizes but it might be possible to speed up computations by exploiting the particular covariance structure similar to Hothorn and Zeileis (2008). In the MOB framework the extension is not quite as straightforward but one strategy could be to adapt double maximum tests à la Bai and Perron (2003).

Hence, the splitting idea can be naturally extended to a two-point search, however, for an unbiased and inference-based selection the corresponding testing strategies might need further adaption.

**Computational details:**

R packages implementing the proposed methods are currently under development at [https://R-Forge.R-project.org/projects/partykit/](https://R-Forge.R-project.org/projects/partykit/).

**Acknowledgements**

This project was partially funded by the Austrian Research Promotion Agency (FFG), grant number 858537.
Circular Regression Trees and Forests with an Application to Probabilistic Wind Direction Forecasting

submitted to the *Journal of the Royal Statistical Society C*

Moritz N. Lang*, Lisa Schlosser*, Torsten Hothorn**,
Georg J. Mayr***, Reto Stauffer*, Achim Zeileis*

*University of Innsbruck,*
Department of Statistics, Universitätsstr. 15, A-6020 Innsbruck, Austria

**University of Zurich,*
Epidemiology, Biostatistics and Prevention Institute, Hirschengraben 84, CH-8001 Zürich, Switzerland

***University of Innsbruck,*
Department of Atmospheric and Cryospheric Science, Innrain 52f, A-6020 Innsbruck, Austria

**Abstract** While circular data occur in a wide range of scientific fields, the methodology for distributional modeling and probabilistic forecasting of circular response variables is rather limited. Most of the existing methods are built on the framework of generalized linear and additive models, which are often challenging to optimize and to interpret. Therefore, we suggest circular regression trees and random forests as an intuitive alternative approach that is relatively easy to fit. Building on previous ideas for trees modeling circular means, we suggest a distributional approach for both trees and forests yielding probabilistic forecasts based on the von Mises distribution. The resulting tree-based models simplify the estimation process by using the available covariates for partitioning the data into sufficiently homogeneous subgroups so that a simple von Mises distribution without further covariates can be fitted to the circular response in each subgroup. These circular regression trees are straightforward to interpret, can capture nonlinear effects and interactions, and automatically select the relevant covariates that are associated with either location and/or scale changes in the von Mises distribution. Combining an ensemble of circular regression trees to a circular regression forest yields a local adaptive likelihood estimator for the von Mises distribution that can regularize and smooth the covariate effects. The new methods are evaluated in a case study on probabilistic wind direction forecasting at two Austrian airports, considering other common approaches as a benchmark.

**Keywords:** circular data, regression trees, random forests, distributional regression, von Mises distribution.
5.1 Introduction

Circular data can be found in a variety of applications and subject areas, e.g., hourly crime rate in socio-economics, animal movement direction or gene-structure in biology, and wind direction as one of the most important weather variables in meteorology. Fitting a statistical model to this type of data requires the incorporation of its specific feature of periodicity. For example, angular data are restricted to an interval such as $[0, 2\pi)$ with 0 being equivalent to $2\pi$.

5.1.1 Circular regression: Conditional mean vs. distributional models

Many approaches to model circular data assume that the circular variable of interest follows a circular distribution, in particular the von Mises distribution which is also known as the “circular normal distribution”. One of the first regression models with a circular response variable and linear covariates was presented by Gould (1969) where the circular mean is predicted by a linear combination of covariates. Johnson and Wehrly (1978) refined this idea by plugging in a link function transforming the linear predictor to a restricted interval of length $2\pi$. This generalized linear model (GLM) type approach was further extended by Fisher and Lee (1992) and subsequently by Fisher (1993) introducing independent GLMs for either the location or scale parameter with appropriate link functions while keeping the other parameter constant. Additionally, they developed a combined heteroscedastic or distributional version with alternating reestimation of the location and scale parameters conditional on the respective sets of covariates until convergence. While all of these models are built on well-elaborated theory, their application in practice remains very challenging, mainly due to the complexity encountered in optimizing the corresponding log-likelihood function which is not globally concave. Therefore, highly-informative starting values are crucial for such circular GLMs to converge (Pewsey, Neuhäuser, and Ruxton, 2013; Gill and Hangartner, 2010). In order to avoid this strong dependence on appropriate initial values, Mulder and Klugkist (2017) present a Bayesian alternative of a homoscedastic GLM for circular data. However, apart from potential difficulties in the optimization procedure of circular GLMs, the interpretation of the underlying additive effects is often challenging as well because the link function is highly nonlinear and the representation of smooth transitions on the unit circle is not straightforward. For example, the same rotation can be obtained in either positive or negative direction on the circle leading to an ambiguous interpretation.

As a very intuitive and data-driven alternative, we propose a flexible tree-based regression approach for modeling circular data by applying the von Mises distribution within the methodology of distributional trees and forests (Schlosser et al., 2019b). The resulting circular regression trees and forests avoid the discussed difficulties of circular GLMs by using the available covariates for partitioning the data into sufficiently homogeneous subgroups so that a simple von Mises distribution without further covariates can be fitted to the circular response in each of these subgroups. This obviates the need for a link function or for iterating between models for the separate distribution parameters. By leveraging the distributional modeling approach, the trees can automatically detect and capture differences in both distribution parameters, providing a fully specified circular response distribution in each terminal node offering a wide range of statistical inference. In addition, the employed tree structure allows to capture non-additive effects while forests enable the modeling of smooth changes. Furthermore, covariates and their possible interactions do not need to be specified in advance as they are selected automatically in the recursive partitioning algorithm.

This novel approach to circular regression trees and forests complements the literature on tree-based circular modeling. Lund (2002) already introduced a circular regression tree algorithm where binary splits are made based on an angular distance measure capturing node
homogeneity. However, this only models changes in the conditional mean but not the conditional variance or full probabilistic distribution which would allow for also considering uncertainty in forecasts (Gneiting, 2008). In contrast, we introduce a distributional tree approach that considers splits in all distribution parameters and yields a full probabilistic predictive model. In addition, as a natural extension, ensembles or forests of circular regression trees are presented in this study, showing that these can further improve predictive power.

5.1.2 Motivating example

To provide a first impression of the presented methodology, a circular regression tree is employed for probabilistic wind direction forecasting. Wind direction is a classical circular quantity and accurate forecasts are of great importance for decision-making processes, e.g., in air traffic management as considered in this study. Figure 5.1 shows an estimated tree for 1-hourly forecasts at Innsbruck Airport, located at the bottom of a narrow valley within the European Alps. Topography channels the wind along the west-east valley axis or along a tributary valley intersecting from the south. Hence, pressure gradients to which valley wind regimes react are considered as covariates along with other meteorological measurements (lagged by one hour) and their derivatives, such as wind direction and wind speed at the airport itself as well as spatial and temporal differences.

Figure 5.1 illustrates the resulting tree along with the empirical (gray) and fitted von Mises (red) wind direction distribution in each terminal node. Based on the fitted location parameters \( \hat{\mu} \), the subgroups can be distinguished into the following wind regimes: (1) Up-valley winds blowing from the valley mouth towards the upper valley (from east to west, nodes 4 and 5); (2) Down-slope winds blowing across the Alpine crest along the intersecting valley towards Innsbruck (from south-east to north-west, node 8); (3) Down-valley winds blowing in the direction of the valley mouth (from west to east, nodes 10, 12 and 13). Node 7 captures observations with rather low wind speeds that cannot be clearly distinguished into specific wind regimes and are consequently associated with a very low estimated concentration parameter \( \hat{\kappa} \), i.e., a high estimated variance. In terms of covariates, the lagged wind direction (“persistence”) is mostly responsible for distinguishing the broad range of wind regimes listed above while the pressure gradients and wind speed separate the data into subgroups with high vs. low precision. A more extensive case study of circular regression trees and forests applied to probabilistic wind direction forecasting at Innsbruck Airport and Vienna International Airport is presented in Section 5.4, along with a benchmark against commonly-used alternative approaches.

The remainder of the paper is structured as follows: The theory on probabilistic circular modeling introducing the von Mises distribution, and circular regression models are discussed in Section 5.2. The methodology of circular regression trees and forests and their features are introduced in Section 5.3. After the case study presented in Section 5.4, a comprehensive summary and conclusions are given in Section 5.5.

5.2 Probabilistic circular modeling

Probabilistic modeling of circular data requires the selection of a probability distribution which accounts for the periodicity of circular data. Generally, this feature can be obtained by “wrapping” the probability density function of any continuous distribution around the unit circle (Mardia and Jupp, 1999). In that way, the wrapped Cauchy distribution or the wrapped normal distribution can be employed to model symmetric unimodal circular data. A close approximation to the wrapped normal distribution that is mathematically simpler and hence easier to use is provided by the von Mises distribution (Fisher, 1993), a purely circular distribution which is also
Figure 5.1. Fitted tree based on the von Mises distribution for 1-hourly wind direction forecasts at the airport of Innsbruck. In each terminal node the empirical histogram (gray) and fitted density (red line) are depicted along with the estimated location parameter (red hand). The covariates selected for splitting are wind direction (meteorological degree), wind speed (ms$^{-1}$), and pressure gradients (dpressure; hPa) west, east and south of the airport, all lagged by one hour. Note that in the meteorological context wind direction is defined on the scale $[0, 360]$ degree and increases clockwise from North (0 degree).
5.2 Probabilistic circular modeling

Known as “the circular normal distribution” and is a common choice for probabilistic modeling of circular data. Based on a location parameter \( \mu \in [0, 2\pi) \) and a concentration parameter \( \kappa > 0 \) the density of the von Mises distribution for an observation \( y \in [0, 2\pi) \) is given by

\[
f_{vM}(y; \mu, \kappa) = \frac{1}{2\pi I_0(\kappa)} e^{\kappa \cos(y - \mu)},
\]

(5.1)

where \( I_0(\kappa) \) is the modified Bessel function of the first kind and order 0 (see, e.g., Mardia and Zemroch 1975, Jammalamadaka and Sengupta 2001, or Ley and Verdebout 2017 for a more detailed overview).

The corresponding log-likelihood function is defined as

\[
\ell(\mu, \kappa; y) = \log(f_{vM}(y; \mu, \kappa)) = - \log(2\pi I_0(\kappa)) + \kappa \cos(y - \mu).
\]

(5.2)

To fit a probabilistic model \( vM(y; \mu, \kappa) \) to a circular response \( y \), the distribution parameters \( \mu \) and \( \kappa \) need to be estimated. This can be done by maximizing the log-likelihood function \( \ell(\mu, \kappa; y) \)

\[
(\hat{\mu}, \hat{\kappa}) = \arg\max_{\mu, \kappa} \sum_{i=1}^{n} \ell(\mu, \kappa; y_i)
\]

(5.3)

yielding maximum likelihood estimators \( \hat{\mu} \) and \( \hat{\kappa} \) such that a fully specified distributional model is fitted to the learning data \( \{y_i\}_{i=1,...,n} \).

The score function

\[
s(\mu, \kappa, y) = \left( \frac{\partial \ell}{\partial \mu}(\mu, \kappa; y), \frac{\partial \ell}{\partial \kappa}(\mu, \kappa; y) \right)
\]

\[
= \left( \kappa \sin(y - \mu), \frac{I_1(\kappa)}{2\pi I_0(\kappa)} + \cos(y - \mu) \right)
\]

(5.4)

provides a way to obtain a measure of goodness of fit of the model for each observation and fitted parameter. Then, the optimization problem in Equation 5.3 can alternatively be specified as

\[
\sum_{i=1}^{n} s(\hat{\mu}, \hat{\kappa}, y_i) = 0.
\]

(5.5)

Figure 5.2 depicts a von Mises model for circular data in \( [0, 2\pi) \) fitted by maximum likelihood, either using a linearized (left) or circular (right) scale. In both cases, the empirical histogram (gray bars) and fitted density (red line) are depicted along with the estimated location parameter (red hand).
bars) is shown along with the fitted density (red line) and estimated location parameter (red
hand). However, this distributional model only considers the circular response variable but no
covariate. Of course, including covariates is of interest in a regression setup for forecasting.

In most generalized linear model (GLM) or generalized additive model (GAM) approaches to
circular regression the location parameter $\mu$ depends on covariates $z$ through a link function $g(\cdot)$,
circular intercept $\mu_0$ and coefficient vector $\beta$:
$$
\mu = \mu_0 + g \left( \beta^\top z \right).
$$
(5.6)
The link function transforms the additive predictor to an interval of length $2\pi$. Typically
$g(x) = 2 \cdot \arctan(x)$ is employed, as suggested by Fisher and Lee (1992). They also developed
a heteroscedastic version by combining two individual GLMs, each for one of the parameters $\mu$
and $\kappa$. This provides a first approach to a fully probabilistic regression model for circular data,
albeit the parameters are not regressed simultaneously on covariates as in the more general
framework provided by generalized additive models for location, scale, and shape (GAMLSS,
Rigby and Stasinopoulos, 2005). Nevertheless, other circular additive models share the previ-
ously discussed difficulties induced by the characteristics of the log-likelihood function and the
strongly nonlinear link function. Referring to additive models in general, it has to be considered
that a proper model specification can be very challenging, particularly for a high number of
covariates and no information on possible interactions. Moreover, the additive structure might
impose a smooth effect even if the true underlying effect is an abrupt shift, which occurs, e.g.,
in atmospheric wind fields.

In contrast, the tree-based circular regression models proposed in the next sections largely
avoid the problems above by employing recursive partitioning in combination with local adaptive
likelihood estimation.

5.3 Circular regression trees and forests
Starting out from the ideas of Lund (2002), we introduce circular regression trees and forests
considering splits in all distribution parameters of the von Mises distribution and providing a
full probabilistic model. Moreover, the resulting tree-based models provide a very intuitive and
data-driven alternative to commonly used GLMs for circular data.

5.3.1 Circular regression trees
Fitting a global model to a full data set can be very challenging, particularly for complex
data with substantial variations. Therefore, separating the data set into more homogeneous
subgroups based on covariates before fitting a local model in each of these subgroups allows
to capture (potential) group-specific effects more precisely and hence can result in an overall
better-fitting model. This is the general idea of regression trees which are combined with dis-
tributional modeling in Schlosser et al. (2019b). Specifying a full distributional model in each
node of the tree yields a distributional regression tree, where selecting the von Mises distribution
enables an application to circular data. The crucial step of how and where to split the data can
be accomplished with the unbiased recursive partitioning algorithms MOB (Zeileis et al., 2008)
or CTree (Hothorn et al., 2006b). For this purpose, model scores are obtained by evaluating the
score function $s(\cdot)$ for each individual observation at the parameter estimates (Equation 5.4).
For the von Mises distribution with its two distribution parameters ($\mu$ and $\kappa$) and a data set
of $n$ observations, this yields an $n \times 2$ matrix that can be employed as a discrepancy measure,
capturing how well each given observation conforms with the estimated location $\hat{\mu}$ and preci-
sion $\hat{\kappa}$, respectively. To capture dependence on covariates, the association between the model’s
scores and each available covariate is assessed using either a parameter instability test (MOB) or a permutation test (CTree). By doing so in each partitioning step, the covariate with the highest significant association (i.e., lowest significant p-value, if any) is selected for splitting the data. The corresponding split point is chosen either by optimizing the log-likelihood (MOB) or a two-sample test statistic (CTree) over all possible partitions. This procedure is repeated recursively until there are no significant parameter instabilities or until another stopping criterion is met (e.g., subgroup size or tree depth). A more detailed description of the applied tree-building algorithm can be found in Appendix 5.6.

Once a distributional tree model is fitted it can be employed to obtain probabilistic predictions for a possibly new set of observed covariates \( z = (z_1, \ldots, z_m) \). Starting at the root node, the tree structure leads the observation to a terminal node where the parameter pair \( (\hat{\mu}, \hat{\kappa}) \) is estimated for the corresponding subset of learning observations. This can also be expressed by employing weights which indicate whether the \( i \)-th learning observation and the observation \( z \) belong to the same terminal node:

\[
w_{\text{tree}}^i(z) = \sum_{b=1}^{B} 1((z_i \in B_b) \wedge (z \in B_b)). \tag{5.7}
\]

Here, \( 1(\cdot) \) is the indicator function and \( B_b \) is the \( b \)-th out of \( B \) segments partitioning the covariate space in disjoint subsets. Then the estimated parameter pair \( (\hat{\mu}, \hat{\kappa})(z) \) specifying the predicted von Mises distribution for a given \( z \) is obtained by a weighted maximum likelihood estimator:

\[
(\hat{\mu}, \hat{\kappa})(z) = \arg \max_{\mu, \kappa} \sum_{i=1}^{n} w_{\text{tree}}^i(z) \cdot \ell(\mu, \kappa; y_i). \tag{5.8}
\]

Therefore, the same parameter pair is estimated for all observations belonging to the same terminal node, which speeds up computation since the parameter estimates do not need to be recalculated for each (new) observation via maximum likelihood but can be extracted directly from the learning sample and the fitted model.

While tree models can capture non-additive effects, their structure and the consequent strict separation of data into subgroups hinders an adequate depiction of smooth effects. They can be included by combining an ensemble of trees in order to obtain a regression forest, which also stabilizes the model.

### 5.3.2 Circular regression forests

A natural extension of (circular) regression trees are ensembles or forests that can improve forecasts by regularizing and stabilizing the model. Random forests introduced by Breiman (2001a) average the predictions of an ensemble of trees, each built on a subsample or bootstrap of the original data. A generalization of this strategy is to obtain weighted predictions by adaptive local likelihood estimation of the distributional parameters (Section 2.3. of Schlosser et al., 2019b; Hothorn and Zeileis, 2017). More specifically, for each (possibly new) observation \( z \) a set of averaged “nearest neighbor” weights \( w_{\text{forest}}^i(z) \) is obtained that is based on the number of trees in which \( z \) is assigned to the same terminal node as each learning observation \( y_i, i \in \{1, \ldots, n\} \). Hence, for a forest of \( T \) trees, the weights are calculated via

\[
w_{\text{forest}}^i(z) = \frac{1}{T} \sum_{t=1}^{T} \sum_{b=1}^{B^t} \frac{1((z_i \in B^t_b) \wedge (z \in B^t_b))}{|B^t_b|}, \tag{5.9}
\]

where \( |B^t_b| \) denotes the number of observations in the \( b \)-th segment of the \( t \)-th tree. Therefore, similar observations ending up more often in the same terminal node have higher weights and thus a stronger influence in the weighted maximum likelihood estimation.
In that way a specific set of weights can be calculated for each observation yielding its specific parameter estimates for the von Mises distribution

$$(\hat{\mu}, \hat{\kappa})(z) = \arg\max_{\mu, \kappa} \sum_{i=1}^{n} w^\text{forest}_i(z) \cdot \ell(\mu, \kappa; y_i).$$

(5.10)

Therefore, the resulting parameter estimates can smoothly adapt to the given covariates $z$ whereas $w_i(z) = 1$ would correspond to the unweighted full-sample estimates and $w_i(z) \in \{0, 1\}$ corresponds to the subgroup selection from a tree. Thus, circular regression forests can capture both smooth and abrupt changes, while covariates and possible interactions are selected automatically and do not explicitly need to be specified beforehand.

### 5.4 Case study: Probabilistic wind direction forecasting

As motivated in Section 5.1, accurate forecasts of wind directions are of great importance for risk management in various fields such as agriculture, energy production, or aviation. For example, in order to direct airplanes to a safe landing, precise knowledge of wind direction for the next hour(s) at the respective airport is highly desirable and adequate prediction methods are required. This section exemplifies the use of circular regression trees and forests with wind direction forecasts for two Austrian airports – one in flat terrain, the other one in mountainous terrain. The results are benchmarked against alternative probabilistic forecasting methods. The study is based on $+1\,\text{h}$ and $+3\,\text{h}$ forecasts employing lagged observations in the vicinity of the airports as possible predictor variables.

#### 5.4.1 Data

The circular response variable considered in this case study is a 10 min-average of wind direction measurements at Innsbruck Airport (INN) and Vienna International Airport (VIE) on an hourly temporal resolution. Temporal information and 1-hourly resolved 10 min mean observations of various meteorological quantities are used as predictor variables, including wind direction, wind speed, temperature, air pressure and humidity, all lagged by one or three hours according to the respective forecasting step. The meteorological variables are measured either directly at the airports or within their vicinities. For Innsbruck, measurements at the airport and along the intersecting valleys are used, whereas, for Vienna, measurements at the airport and within its vicinity of approximately 30 km are used. Figure 5.3 provides a topographical overview of the airports and their surrounding areas with the station sites employed in this study. In addition, we use derived quantities such as 3-hourly means, minima and maxima, as well as 1-and 3-hourly temporal changes and spatial differences towards the airport of the respective quantities. An overview of the employed data sets can be found in Table 5.1.

The data used in this study consists of five subsequent years from January 2014 to December 2018. After first eliminating predictor variables with more than 5% missing values and then time points with any missing observations, the data set consists of 41979 time points and 260 covariates for Innsbruck, and of 38985 time points and 494 covariates for Vienna, respectively.

#### 5.4.2 Models and evaluation

For a fair evaluation of circular regression trees and forests, and to investigate whether they can be applied as a reasonable alternative to already existing approaches, three additional statistical models are employed in this study for probabilistic forecasting of wind directions. Two of them are based on existing approaches used in the meteorological field, while the third is a state-of-the-art GLM-type model to forecast circular response variables.
5.4 Case study: Probabilistic wind direction forecasting

Figure 5.3. Overview of the study area for Innsbruck Airport (left) and Vienna International Airport (right). For Innsbruck, 4 stations at the airport and 6 stations along the intersecting valleys are used, whereas, for Vienna, 9 stations at the airport and 13 stations within the vicinity of approximately 30 km are used. Elevation data are obtained from the TandDEM-X digital elevation model with a horizontal resolution of 90 m (Wessel, Huber, Wohlfart, Marschalk, Kosmann, and Roth, 2018).

Table 5.1. Overview of the data sets employed in the case study: For Innsbruck and Vienna, various meteorological variables and derived quantities of these are considered at the respective stations, located either directly at the airports or in their vicinities.

<table>
<thead>
<tr>
<th>Data components</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temporal information:</td>
<td>Time of the day, day of the year</td>
</tr>
<tr>
<td>Meteorological variables:</td>
<td>Wind direction, wind (gust) speed, (reduced) air pressure, relative humidity, temperature</td>
</tr>
<tr>
<td>Derived quantities:</td>
<td>3-hourly means/minima/maxima, 1-hourly and 3-hourly temporal changes, spatial differences towards the airport</td>
</tr>
<tr>
<td>Weather stations (Innsbruck):</td>
<td>4 stations at the airport, as well as Igls, Kematen, Kufstein, Landeck, Patscherkofel, and Steinach</td>
</tr>
<tr>
<td>Weather stations (Vienna):</td>
<td>9 stations at the airport, as well as Arsenal, Donaufeld, Exelberg, Gänserndorf, Groß-Enzersdorf, Gumpoldskirchen, Hohe Warte, Innere Stadt, Jubiläumswarte, Mariabrunn, Sebersdorf, Unterlaa, and Wolkersdorf</td>
</tr>
</tbody>
</table>

- Climatological model: Accurate knowledge of weather quantities’ climatologies can be important for a wide range of applications. While forecasts based on climatologies, by construction, do not adapt to the current weather situation they are still a useful baseline for the validation of newly developed forecasting systems (Simon, Umlauf, Zeileis, Mayr, Schulz, and Diendorfer, 2017; Stauffer et al., 2017a).
Specifically, the climatology employed in the following uses all observations at the same
time (to adapt to daily cycles) in a window of 31 days centered around the day of interest
(to adapt to seasonal cycles) in all available years in the sample. Based on these observa-
tions a probabilistic model is obtained by maximum likelihood estimation as described in
Section 5.2. This approach follows Vogel, Knippertz, Fink, Schlüeter, and Gneiting (2018)
and is discussed in a comprehensive summary on different time-adaptive training schemes
in Lang, Lerch, Mayr, Simon, Stauffer, and Zeileis (2019a).

- **Persistency model:** The persistence describes the previous value of a single weather quan-
tity in a time series. Like the climatology it is a very basic prediction model that is often
applied as a baseline reference in weather forecasts (NOAA’s National Weather Service,
2019). Especially in nowcasting tasks with very short forecasting steps, the persistence
can provide very good estimates.

To gain a full probabilistic persistency model, we proceed similarly as for the climatological
model by using maximum likelihood estimation and fitting the distribution parameters of
the von Mises distribution conditional on lagged response values according to the descrip-
tion in Section 5.2. We fit one model for every hour throughout the validation data set
employing the previous six lagged response values as training data. In order to allow for
a stronger influence of observations closer to the time of interest, exponential smoothing
is employed with a smoothing factor of 0.5; accordingly, for every prediction an equal in-
fluence rate of 50 percent is assigned both to the current observation and to the previous
five observations together. Observations with longer time lags have exponential weights
below 0.01 and are therefore omitted from the training data.

- **Generalized linear model:** Traditional approaches to forecast circular response variables
are often based on circular GLM-type models (Fisher, 1993). As discussed in Section 5.1,
circular regression models often experience the problem that the likelihood function can
be strongly irregular which makes optimization rather difficult. Hence, they often do
not converge if no appropriate initial values are provided (Pewsey et al., 2013; Gill and
Hangartner, 2010). In this study, to be able to employ a GLM out of the box as a reference,
we use the Bayesian implementation of Mulder and Klugkist (2017) which depends less on
initial values due to an MCMC sampling algorithm using weakly informative priors.

Following Mulder and Klugkist (2017) the model uses a link function \( g(\cdot) \) to keep the
response values within an interval of length 2\( \pi \). As the implementation cannot handle
circular covariates, we use the components of the lagged 2-dimensional wind vector \((u, v)^\top\)
and the lagged wind speed \(spd\) as predictor variables. The model formula for the location
parameter \(\mu\) of the von Mises distribution can be written as:

\[
\mu = \beta_0 + g(\beta_1 \cdot u + \beta_2 \cdot v + \beta_3 \cdot spd)
\]  

(5.11)

with \(\beta_0\) being a circular intercept, \(\beta\), the regression coefficients and the link function
\(g(x) = 2 \cdot \arctan(x)\). In addition, a constant concentration parameter \(\kappa\) is fitted to the full
learning sample. To allow for seasonally varying error characteristics in both bias and slope
coefficients, and to allow for seasonal heteroscedasticity captured by the concentration
parameter, we use the same time-adaptive training approach as for the climatological
model; hence, separate models are estimated over all observation dates, using the same
time of 31 days centered around the day of interest over all available years in the training
data (Lang et al., 2019a).

- **Circular regression tree:** For the circular regression tree introduced in Section 5.3.1, all
covariates provided in the learning data can be considered due to an intrinsic automatic
variable selection performed in the tree estimation. The tree is built with the newly
developed R package `cirtree` employing the CTree algorithm (Hothorn et al., 2006b) using a minimal number of 2000 observations in each terminal node (argument `minbucket`).

- **Circular regression forest**: Following the description in Section 5.3.2, the circular regression forest used in this study is constructed based on 100 individual trees employing the R package `cirtree`. Each of these trees is again built by the CTree algorithm on a subsample containing 30 percent of the original learning data. All covariates are included for building each tree which ensures that the lagged response variable is always considered for splitting. This bagging approach can be applied in `cirtree` by setting the argument `mtry` to the total number of covariates. Since a high number of possible split points leads to high computational costs, the covariates are binned in a maximum of 50 classes (argument `nmax = c(yx = Inf, z = 50)`). Contrary to a single-tree model, forests usually consist of very large trees as they are not prone to overfitting the data due to the stabilization obtained by combining the individual trees. Therefore, we use the following control arguments to build rather large trees: The minimal number of observations to perform a split is set to 20 (argument `minsplit`), the minimal number of observations in each terminal node is set to 7 (argument `minbucket`), and the significance level for variable selection is kept at its maximum value of 1 (argument `alpha`).

To compare the predictive performance of all proposed models, a circular analogue of the continuous ranked probability score (CRPS) as introduced by Grimit, Gneiting, Berrocal, and Johnson (2006) is computed. Just as the linear version of the CRPS (for more details see Hersbach, 2000) it is a proper scoring rule (Gneiting and Raftery, 2007) and measures the difference between an observation and the corresponding predicted distribution function in order to assess the probabilistic goodness of fit for the estimated model. Hence, the lower the CRPS value the better the predictive performance. Contrary to the linear version, the circular CRPS reduces not to the absolute error but to the angular distance when the forecast is deterministic.

In addition to the raw CRPS, corresponding skill scores are computed to assess differences in the improvement of the various statistical models over the climatological model used as a reference:

\[
\text{CRPSS}_{\text{model}} = 1 - \frac{\text{CRPS}_{\text{model}}}{\text{CRPS}_{\text{climatology}}}. \tag{5.12}
\]

All scores presented in the next section are computed out-of-sample based on five years of data. For the persistency model only dates prior to the time of interest are used and the validation is performed rolling over all observations. For all other models a five-fold cross-validation is employed using up to four calendar years for model training and the remaining single calendar year for validation. Due to the large sample size of 24 hourly values per day over five years, some kind of temporal aggregation is needed to ensure a correct visual comparison of the individual methods. The analyses performed have shown that for the employed models the variability of the predictive performance over the five years is lower than over a single day or over a single year. Hence, CRPS and CRPS skill scores are aggregated over the respective five validation years which yields 24 hourly scores per month averaged over the five validation years.

### 5.4.3 Results

This section provides a detailed analysis on the predictive performance of the different proposed statistical models applied to probabilistic wind direction forecasting. To ensure a comprehensive comparison of the models, wind direction forecasts are evaluated for two different lead times at two airports with different climatological site characteristics. Figure 5.4 shows the CRPS values of the employed models at forecast steps +1h and +3h for Innsbruck (Panels a, c) and
Figure 5.4. CRPS skill scores of wind direction forecasts based on the full predictive von Mises distribution for +1 h and +3 h forecasts at the Innsbruck Airport and Vienna International Airport. Each box-and-whisker contains 24 hourly scores for each of the 12 months averaged over the 5 validation years which yields a total of 288 yearly mean values. The scores are shown for the climatological, persistency and the linear model as well as for the circular regression tree and forest.

Vienna (Panels b, d). The scores are aggregated over the five validation years, yielding yearly mean values for every hour per calendar month, with a lower score indicating better performance. The circular regression forest overall provides the best predictive performance, followed by the circular regression tree and the persistency model for both stations at both forecast steps; except for the 3 h forecasts at Innsbruck where the persistency model is outperformed by all others. In comparison to the circular regression tree and forest, for both stations and forecast steps, the climatological model and the linear model show clearly higher CRPS values and hence a lower predictive performance. The different site characteristics of the airports Innsbruck (Figure 5.4 a, c) and Vienna (Figure 5.4 b, d) seem to have an effect on the absolute level of the model performances and on their respective predictive performance variances. At Innsbruck, due to the surrounding mountains only a limited number of possible wind directions exists, namely the three wind regimes discussed for Figure 5.1 in Section 5.1. Therefore, for Innsbruck the wind direction remains rather constant in one of these possible states, but once a change takes place it is mostly a major wind direction shift, e.g., from up-valley to down-valley. Due to the few wind regimes the rather inflexible climatological and linear models score relatively well with similar CRPS values as the other models (Figure 5.4 a, c). In addition, at Innsbruck the potential high prediction errors in case of a change of the wind regime seem to lead to a higher variation in the predictive performance for all models in comparison to Vienna; this variation is especially high for the persistency model due to its strong vulnerability to abrupt wind shifts. On the contrary, at Vienna smaller and less abrupt changes in the wind direction as well as less pronounced wind regimes are observed due to the less mountainous surrounding. This seems to weaken the predictive performance of the climatological and linear models, and to reduce the performance variability for all models (Figure 5.4 b, d).

The different forecast steps have apparently only a minor effect on the predictive performance
5.5 Summary and conclusion

Extending the toolbox for modeling circular data, circular regression trees and forests are established by coupling model-based recursive partitioning with the von Mises distribution. By separating the data into more homogeneous subgroups, possible difficulties in circular regression are bypassed as covariates are solely considered for splitting and group-specific models are fitted without further covariates. In addition, by specifying the von Mises distribution for each node and allowing for splits in both distribution parameters \( \mu \) and \( \kappa \), fully probabilistic forecasts are provided.

Figure 5.5. As Figure 5.4, but showing CRPS skill scores with the climatology model as reference. Skill scores are in percent; positive values indicate improvements over the reference.

of the climatological model and the linear model at both stations. As expected, for the persistency model, at both stations, higher scores for the 3 h forecast (Figure 5.4c,d) reveal a lower performance for longer lead times; this is due to the lower information content of 3-hourly instead of 1-hourly lagged response values employed as covariates in the persistency model. The circular regression tree and forest seem to partially compensate for the lower skill of the lagged response values by other covariates, hence their predictive performance only slightly decreases for the longer lead time. This compensation is especially evident for Innsbruck, where the performance difference between the persistency model and the tree-based methods significantly increases from the 1-hourly to the 3-hourly forecast.

In addition to the raw CRPS (Figure 5.4), CRPS skill scores with the climatological model as a reference are provided in Figure 5.5. Skill scores are in percent, where positive values indicate an improvement in the predictive performance over the reference. For all setups, the circular regression forest has the highest skill scores with a mean performance gain of 13–25% and 58–71% for Innsbruck and Vienna, respectively. As discussed for Figure 5.4, this improvement over the climatological model is lower for Innsbruck due to the low number of predominant wind regimes and hence a relatively good performance of the climatological model. Additionally, Figure 5.5 shows that while the persistency model’s performance is lower than the reference (Panel c) the tree-based models can compensate for the low skill of the lagged response values employed as covariates and, hence, are still significantly superior to the reference.

5.5 Summary and conclusion

Extending the toolbox for modeling circular data, circular regression trees and forests are established by coupling model-based recursive partitioning with the von Mises distribution. By separating the data into more homogeneous subgroups, possible difficulties in circular regression are bypassed as covariates are solely considered for splitting and group-specific models are fitted without further covariates. In addition, by specifying the von Mises distribution for each node and allowing for splits in both distribution parameters \( \mu \) and \( \kappa \), fully probabilistic forecasts are provided.
The performance of the novel circular regression trees and forests is assessed in a case study for short-term probabilistic wind direction forecasting at two airports with different site characteristics. As benchmark models, probabilistic climatology and persistency models, as well as a state-of-the-art circular GLM-type model are evaluated based on proper scoring rules. In summary, the circular regression trees and forests have the highest predictive performance in this setting. For cases without changes in the wind regime, lagged response values provide already highly skillful estimates leading to a good performance of the persistency model as observed for short-term wind direction forecasts in this study. While in these cases the trees and forests also benefit from the highly informative lagged response, they can compensate for a lower information of this covariate by incorporating other quantities and possible interactions of these, contrary to the persistency model (see Figure 5.5). Hence, the tree-based models provide reliable forecasts in all tested meteorological settings. For operational use, a possible extension could be the incorporation of numerical weather predictions as (additional) covariates. While this probably only slightly improves the predictive skill for short leadtimes, it possibly extends the potential forecast range of the different methods.

For the specific task of wind forecasting, the wind direction is often only relevant if the wind speed is sufficiently high. Hence, it is of interest to account for both quantities simultaneously, e.g., by considering a bivariate normal distribution for wind vectors (from which wind speed and wind direction can be obtained). The parameters of this bivariate normal distribution could then be linked to available covariates using an additive regression framework (as proposed by Lang, Mayr, Stauffer, and Zeileis, 2019b) or using a tree-based approach, similar to the one proposed in this paper. Moreover, a rather different approach for a combined response of wind speed and wind direction would be a two-step or hurdle model: In the first step this could build on the truncated normal model of Thorarinsdottir and Gneiting (2010) to capture wind speed; in the second step a circular wind direction model is leveraged given that a certain hurdle for the wind speed is crossed.

Another possible improvement for obtaining more parsimonious circular regression trees is to consider splitting circular covariates into two circle segments by searching two split points simultaneously rather than sequentially at different depths. While this might slightly improve the predictive performance of circular regression trees, this should not affect the performance of the forests, as they consist of very large trees with many different splits.

To conclude, in general the tree structure can capture nonlinear changes, shifts, and potential interactions in covariates without prespecification of such effects. As supported by the presented case study, this can be particularly useful for modeling a highly fluctuating response, such as typically observed for wind direction, or/and in case of a large number of possible covariates. Moreover, the case study shows that building ensembles of circular regression trees can even improve the forecasting performance, as the resulting forests allow for modeling smooth effects and stabilize the model.

Computational details

The corresponding implementation of the proposed methodology for circular regression trees and forests is provided in the R package circtree (version 0.1.0). The package is based on the disttree package (version 0.2.0) which applies the main tree building functions from the partykit package (version 1.2.5). All three packages are available on R-Forge at https://R-Forge.R-project.org/projects/partykit/.

For the circular GLM considered as reference model the corresponding implementation is provided in the R package circglmbayes by Mulder and Klugkist (2017). In particular the function circGLM is applied to estimates the intercept and regression coefficient along with the concentration parameter.
5.6 Appendix: Tree algorithm

Acknowledgments

This project was partially funded by the Austrian Research Promotion Agency (FFG) grant number 858537. Torsten Hothorn received funding from the Swiss National Science Foundation, grant number 200021_184603. Lisa Schlosser received a PhD scholarship granted from the University of Innsbruck.

5.6 Appendix: Tree algorithm

This section provides a more detailed overview on the permutation-test-based CTree algorithm (Hothorn et al., 2006b), specifically for circular data as applied for building circular regression trees and forests presented in this case study. An alternative tree-building framework is provided by the MOB algorithm, which is based on M-fluctuation tests (see Zeileis et al., 2008, for more details).

In the following, the testing and splitting strategy is described for the root node of the tree which contains the entire learning sample. For a complete tree model, the same procedure is applied iteratively to all resulting child nodes with the corresponding subsamples.

First, employing the von Mises distribution, a distributional model \( vM(y; \mu, \kappa) \) is fitted to the learning sample of circular observations \( \{y_i\}_{i=1}^n \) as explained in Section 5.2. In a next step, a goodness-of-fit measurement is obtained for each parameter and each observation by evaluating the score function \( s(\mu, \kappa, y) \) at the estimated location and concentration parameter \( \hat{\mu} \) and \( \hat{\kappa} \). To detect dependencies between the resulting scoring matrix

\[
\begin{bmatrix}
  s(\hat{\mu}, \hat{\kappa}, y_1)_1 & s(\hat{\mu}, \hat{\kappa}, y_1)_2 \\
  \vdots & \vdots \\
  s(\hat{\mu}, \hat{\kappa}, y_n)_1 & s(\hat{\mu}, \hat{\kappa}, y_n)_2
\end{bmatrix}
\]

(5.13)

and each possible split variable \( z_l \in \{z_1, \ldots, z_m\} \) a permutation test is applied. In particular, the null hypotheses of independence of each split variable and the scores is assessed by employing the multivariate linear statistic

\[
t_l = vec \left( \sum_{i=1}^n v_l(z_{li}) \cdot s(\hat{\mu}, \hat{\kappa}, y_i) \right)
\]

(5.14)

with \( s(\hat{\mu}, \hat{\kappa}, y_i) \in \mathbb{R}^{1 \times 2} \). For a numeric split variable \( z_l \) the transformation function \( v_l \) is simply the identity function \( v_l(z_{li}) = z_{li} \) such that \( t_l \in \mathbb{R}^2 \) as the “vec” operator converts the matrix of dimension \( 1 \times 2 \) into a \( 2 \) column vector. If \( z_l \) is a categorical variable with \( h \) categories then \( v_l(z_{li}) = (I(z_{li} = 1), \ldots, I(z_{li} = h)) \), hence, \( v_l \) returns a unit vector of dimension \( h \) where the entry 1 indicates the category of \( z_{li} \). In this case the “vec” operator converts the \( h \times 2 \) matrix into a \( h \cdot 2 \) column vector by column-wise combination such that \( t_l \in \mathbb{R}^{h \cdot 2} \). If there are any observations with missing values these are not included in the calculation of \( t_l \).

To map the multivariate linear statistic \( t_l \) onto the real line a univariate test statistic \( c \) is employed, for example in a quadratic form

\[
c_{\text{quad}}(t_l; \mu_l, \Sigma_l) = \left( t_l - \mu_l \right) \Sigma_l^+ \left( t_l - \mu_l \right)^\top
\]

(5.15)

where \( \mu_l \) and \( \Sigma_l \) are the conditional expectation and the covariance of \( t_l \), as derived by Strasser and Weber (1999) and used for standardization, and \( \Sigma_l^+ \) is the Moore-Penrose inverse of \( \Sigma_l \). As an alternative, also a maximum form \( c_{\text{max}} \) can be considered such that the maximum of the absolute values of the standardized linear statistic is returned.
The asymptotic conditional distribution of $c(t_l, \mu_l, \Sigma_l)$ is either normal (for $c_{\text{max}}$) or $\chi^2$ (for $c_{\text{quad}}$) owing to the asymptotic conditional distribution of the linear statistic $t_l$ being a multivariate normal with parameters $\mu_l$ and $\Sigma_l$ (Strasser and Weber, 1999). With this knowledge at hand, the corresponding $p$-values can be calculated and used to select the best splitting variable. A small $p$-value corresponding to $c(t_l, \mu_l, \Sigma_l)$ indicates a strong discrepancy from the assumption of independence between the scores and the split variable $z_l$. Therefore, if any of the Bonferroni-adjusted $p$-values is beneath the selected significance level, the partitioning variable $z_l^*$ with the lowest $p$-value is selected as split variable, otherwise no split is performed. This early stopping induced by the significance level is referred to as “pre-pruning” which is often avoided for forest models by setting the significance level to 1.

To select the best split point within the already chosen split variable, again, a linear test statistic is employed. In particular, for a breakpoint $r$ of the variable $z_l^*$ leading to two subgroups $B_{1r}$ and $B_{2r}$, the discrepancy between score functions in the subgroups is measured by evaluating

$$t_{qr} = \sum_{i \in B_{qr}} s(\hat{\mu}, \hat{\kappa}, y_i)$$

for $q \in \{1, 2\}$. The breakpoint that leads to the highest discrepancy is then selected as split point as defined by

$$r^* = \arg\min_r \left( \min_{q=1,2} \left( c(t_{qr}^{\mu_l}, \mu_l, \Sigma_l) \right) \right).$$

Subsequently, the same testing and splitting procedure is repeated in each of the resulting subgroups until some stopping criterion is reached. Next to the already mentioned significance-level-based stopping criterion, i.e., a minimal $p$-value for the statistical tests, also a maximal tree depth or a minimal number of observations in a node can be employed as stopping criteria.


BIBLIOGRAPHY


