Confidence Limits for Prediction Performance

Doctoral Thesis

by Pascal Rink

A thesis submitted in partial fulfillment of the requirements for the degree of
doctor rerum naturalium
(Dr. rer. nat.)
in the
Applied Statistics and Biometry Working Group in
Faculty 3: Mathematics and Computer Science at the
University of Bremen

First reviewer:	Prof. Dr. Werner Brannath
Second reviewer:	Prof. Dr. Marvin N. Wright
Date of submission:	27 March 2025
Date of defense:	30 April 2025

Summary

As machine learning algorithms become increasingly integrated into critical systems, assessing the reliability of their predictions is essential, especially when errors can have severe consequences. Incorporating statistical methods can help to quantify the inherent uncertainty and improve decision-making. This work proposes a new method for estimating confidence limits for prediction performance.

In the first part, we introduce fundamental concepts and findings from the machine learning and statistical inference literature, framing the selection and evaluation of prediction models as a statistical inference problem. In particular, we consider the simultaneous evaluation of multiple candidate models and interpret this as a multiple testing problem.

We also explore the bootstrap and nonparametric bootstrap tilting, which provides a reliable approach for estimating confidence intervals without the need to assume a specific underlying distribution.

The second part integrates these concepts and presents the proposed multiplicity-adjusted bootstrap tilting lower confidence limits for conditional prediction performance. This approach is computationally undemanding and universally applicable to any combination of prediction models, model selection strategies, and performance measures.

We prove that the proposed interval asymptotically achieves the nominal coverage probability and conduct simulation experiments to assess its goodness in finite samples. Specifically, we investigate the prediction accuracy of LASSO and random forest classifiers. The proposed approach shows reliable coverage and competitive lower confidence limits. In contrast, we also show that recent alternative methods such as bootstrap bias-corrected cross-validation and nested cross-validation may fail to accurately track conditional performance.

Finally, we apply the proposed approach to real-world data, where it demonstrates stability when model selection is highly sensitive to the allocation of the sample data to the learning and evaluation sets or in the presence of a distribution shift.

Zusammenfassung

Machine Learning und Vorhersagemodelle werden zunehmend in kritische Anwendungsbereiche integriert. Damit wird gleichzeitig die Bewertung der Zuverlässigkeit der Vorhersagen immer wichtiger, insbesondere wenn Fehler schwerwiegende Folgen haben können. Diese Arbeit schlägt eine neue Methode zur Schätzung von Konfidenzgrenzen für die Vorhersagegüte vor.

Im ersten Teil werden grundlegende Konzepte aus der Literatur zu Machine Learning und statistischer Inferenz vorgestellt. Ziel ist es, die Auswahl und Bewertung von Vorhersagemodellen als statistisches Inferenzproblem zu verstehen. Insbesondere wird die simultane Gütebewertung mehrerer konkurrierender Modelle als multiples Testproblem interpretiert.

Außerdem wird der Bootstrap und das sogenannte Nonparametric Bootstrap Tilting beschrieben, eine Methode zur zuverlässigen Schätzung von Konfidenzintervallen ohne starke Verteilungsannahmen.

Im zweiten Teil werden diese Konzepte zusammengeführt und die vorgeschlagenen Multiplizitäts-adjustierten Bootstrap Tilting Konfidenzintervalle für die bedingte Vorhersagegüte vorgestellt. Der Ansatz ist recheneffizient und universell anwendbar, unabhängig von der Kombination der Vorhersagemodelle, den Strategien zur Modellselektion und den Gütemaßen.

Es wird bewiesen, dass die vorgeschlagenen Konfidenzintervalle asymptotisch die nominale Überdeckungswahrscheinlichkeit erreichen. In Simulationsexperimenten wird ihre Güte in endlichen Stichproben bewertet. Insbesondere wird die Vorhersagegenauigkeit von LASSO und Random Forest Klassifikatoren untersucht. Die vorgeschlagenen Intervalle erreichen zuverlässig die gewünschte Überdeckungswahrscheinlichkeit und bieten konkurrenzfähige untere Konfidenzgrenzen. Im Vergleich dazu schätzen alternative Methoden wie Bootstrap Bias-Corrected Cross-Validation und Nested Cross-Validation nicht immer zuverlässig Konfidenzintervalle für die bedingte Vorhersagegüte.

Abschließend wird der vorgeschlagene Ansatz auf echte Daten angewendet, wo er sich als stabil erweist, selbst wenn die Modellselektion empfindlich auf die Aufteilung der Daten in Lern- und Evaluationsdaten reagiert oder wenn sich die Verteilung zwischen den beiden Phasen ändert.

Contents

List of Abbreviations

BBC-CV	Bootstrap bias-corrected cross-validation
CP	Clopper-Pearson
FWER	Familywise error rate
LASSO	Least absolute shrinkage and selection operator
MABT	Multiplicity-adjusted bootstrap tilting
NCV	Nested cross-validation

List of Figures

List of Symbols

$\xrightarrow{\text{a.s.}}$	Almost sure convergence
$\xrightarrow{\mathbb{P}}, \xrightarrow{\mathcal{L}}$	Convergence in probability and in distribution
$\mathbb{1}$	Indicator function
$\alpha, \alpha_{\text{adj}}, \hat{\alpha}_{\text{adj}}$	Global significance level, true and estimated adjusted level
B	Number of bootstrap samples
$\boldsymbol{\beta}, \hat{\boldsymbol{\beta}}$	True and estimated coefficient vector
$\boldsymbol{\eta}$	Vector of nuisance parameters
$\hat{f}$	Prediction function
F_n	Empirical distribution function
$F_{\boldsymbol{w}}, F_\tau$	Reweighted distribution function, with tilting parameter τ
$H_n, \hat{H}_n^*$	True and bootstrap distribution function of test statistic
$L, \hat{L}$	Lower confidence limit and its estimate
$\lambda, \lambda_{\max}$	ℓ_1 regularization parameter and maximum regularization parameter
m	Number of null hypotheses, number of competing prediction models
m_0	Number of true null hypotheses
$\mathcal{H}^0, \mathcal{H}^A$	Null and alternative hypothesis
Φ	Cumulative distribution function of the standard-normal distribution
T, T^*	Test statistic and bootstrap test statistic
τ	Tilting parameter
$\theta, \hat{\theta}_n, \hat{\theta}_n^*$	Goal of inference, and sample and bootstrap sample estimates
$\hat{\theta}_{\text{V}}$	Estimate of validation performance
$\boldsymbol{w}, w_i(\tau)$	Vector of sampling weights, exponential tilting weights
X_i, X_i^*	Samples and bootstrap samples
$\bar{X}_n, \bar{Y}_n$	Sample means
ξ	Reference value in null hypothesis
$y_i, \hat{y}_i$	True and predicted class label
z_q	$q \cdot 100\%$-quantile of the standard-normal distribution

List of Tables

Chapter 1

Introduction

1.1 Motivation

In recent years, machine learning has transitioned from specialized applications to becoming omnipresent. Today, machine learning is seamlessly integrated in everyday life, powering personalized recommendation algorithms on streaming platforms and e-commerce websites, as well as voice assistants and social media apps, controlling what content we see and interact with.

However, machine learning influences not only consumer technology, but has entered even more critical areas. In healthcare, it supports doctors in decision-making, diagnosing conditions, predicting treatment outcomes, and optimizing medication. In finance, machine learning assesses risks and possibilities, and detects fraud and money laundering. In cybersecurity, it helps to identify and prevent threats, from e-mail spam to hacking attempts.

As machine learning algorithms evolve and grow more powerful, they are increasingly integrated into critical systems affecting both personal and public well-being and security. In many of its applications, machine learning is utilized to make a decision whether an observation should be categorized into the one or the other of two competing classes. For example, a spam filter labels an e-mail as *spam* or *not spam*; fraud detection systems classify transactions as *fraudulent* or *legitimate*; and diagnostic tests determine whether a medical condition is *present* or *absent*.

As the consequences of wrong decisions grow progressively more severe, there is an increasing need to evaluate the reliability of such predictions. A missed malicious e-mail can compromise data security, while incorrectly flagged legitimate transactions can disrupt a customer's financial stability. A false positive in medical diagnosis may lead to unnecessary treatments, while a false negative may prevent timely intervention; both may eventually harm patient's health.

Machine learning algorithms face inherent uncertainty due to various factors such as noisy, incomplete, or scarce data. This variability means that basic performance metrics may not be sufficient for capturing a model's probabilistic nature. To assess and quantify its reliability, more sophisticated approaches are necessary.

Statistical methodology is crucial in this context, as it provides tools to quantify uncertainty and draw meaningful conclusions about the reliability of predictions. However, statistical methods are not always applied in machine learning. Many practitioners focus on empirical performance metrics and heuristics rather than incorporating formal statistical reasoning. As Breiman (2001) pointed out, statisticians and machine learning practitioners often approach problems from different perspectives. Statistical modeling emphasizes understanding the data generating process and interpreting underlying patters, while machine learning typically prioritizes effective prediction, often neglecting to quantify the confidence in those predictions.

Understanding a prediction model's confidence is as important as the prediction itself, especially when potential consequences of wrong decisions are severe. Statistical reasoning, including hypothesis testing and confidence intervals, provides valuable insights into a prediction model's reliability. Breiman (2001) argued that, while machine learning led to revolutionary advances in prediction from complex, high-dimensional, and noisy real-world data, drawing statistically valid conclusions should not be ignored. By incorporating statistical methods, machine learning practitioners can better understand uncertainty and make more informed decisions.

In the present work, we will propose a universal method for estimating confidence limits for prediction performance measures, particularly in situations where data is scarce.

1.2 Outline

The present work is designed to address both statisticians and machine learning practitioners. It avoids unnecessary technical depth in areas where such detail is not absolutely essential. Instead, it focuses on explaining key concepts in an illustrative manner, and provides references to more detailed literature where appropriate. The goal is to offer a coherent and self-contained presentation that is of use to readers from both fields.

This work both builds and expands on our comprehensive publication Rink & Brannath (2025), which can be accessed online at `https://doi.org/10.1007/s10994-024-06632-w`. Specifically, while the principle idea remains the same, in this work, we additionally investigate theoretical properties that also slightly

affect the methodology of the proposed method. We will discuss the differences at a later point.

We divide our presentation into two parts. In Part I we will predominantly introduce well-established concepts and findings from the literature. In Chapters 2 and 3 we will begin with a brief overview of foundational ideas and results from machine learning and statistical inference, notably conditional performance and multiple testing. The objective will be to frame the selection and evaluation of a prediction model in terms of a statistical inference problem. In addition, in Chapter 4, we will discuss the bootstrap method and the nonparametric bootstrap tilting confidence interval, which our proposed method is based on.

In Part II, in contrast, we will almost entirely present only our own contributions, integrating the various introduced concepts into a multiplicity-adjusted version of the nonparametric tilting interval. In Chapter 5, we will develop its methodology and prove asymptotic properties. Then, in Chapter 6, we will assess its goodness in simulation experiments, and apply it to real-world data in Chapter 7. We will conclude Part II with a discussion of our findings in Chapter 8.

Furthermore, the present work contains comprehensive appendices. They supplement the presentations in the various chapters with additional key concepts, proofs, and results. Moreover, the `R` code to all plots, computations, and simulations is provided online via a public GitHub repository at `https://gitlab.informatik.uni-bremen.de/s_opbgf3/clfpp`.

The main body of this work was written entirely by Pascal Rink. Any use of the pronoun *we* is purely for stylistic reasons.

Part I

Foundations

Chapter 2

Machine Learning

In this chapter, we will explore essential concepts in machine learning for the development of effective prediction models. We will begin this chapter in Section 2.1 with a fundamental comparison of the two primary fields of machine learning, supervised and unsupervised learning. Then, in Section 2.2, we will discuss various aspects of performance estimation, which is a critical aspect of model development and goodness assessment. We will conclude Section 2.2 with contrasting conditional and unconditional performance. These two concepts will be of special importance later in Part II of this work. Finally, in Section 2.3, we will turn to model selection and address the challenges of choosing the best model for a particular problem. This section examines the issue of overfitting and ends with two approaches to model selection and evaluation, one standard approach and one that has only recently been proposed.

In this chapter, we will only present selected aspects of machine learning. There are numerous references that provide comprehensive coverage and discussion on the field. They include Hastie et al. (2009), Murphy (2012), and Shalev-Shwartz & Ben-David (2014).

2.1 Supervised and unsupervised learning

A major distinction in machine learning is between unsupervised and supervised learning. In unsupervised learning, the goal is to explore and understand inherent patterns and relationships within the data without much prior knowledge of what those might be. Techniques such as clustering or dimensionality reduction are commonly used for such tasks and work largely without human intervention. Therefore, unsupervised learning tasks often appear when there is a large amount of data that is too costly to label or structure by hand.

In supervised learning, in contrast, the data is labeled. In particular, the data

consists of some input values called *features* and an output value, the *label*. One of the primary goals in supervised machine learning is to learn a mapping from the features to the labels. This is typically referred to as *training* of a *prediction model*. This process is considered to be supervised, because the training is guided by the labeled data. For example, linear regression can be understood as a type of supervised learning.

The prediction model will later be utilized to predict the labels of new observations. Of course, its goodness determines its usefulness, and hence, the goal is to train prediction models that maximize performance, that is, the similarity of the predictions and the true labels on new data. The specific way to measure performance depends on the type of data.

This work concerns supervised machine learning in binary classification problems, that is, the labels are limited to two discrete classes. Examples of such problems include e-mail spam recognition, or medical testing to determine whether a patient has a certain disease or not.

2.2 Performance estimation

In this section, we will discuss various aspects of performance estimation. We will start by setting the stage for binary classification problems in Section 2.2.1, introducing both linear and non-linear classifiers, along with an example of each. In Section 2.2.2, we will give an overview of several common measures of prediction performance in classification problems. Then, in Section 2.2.3, we will outline how to estimate the performance on new observations and introduce cross-validation as a popular tool for this purpose in Section 2.2.4. Finally, we will conclude with an important and consequential discussion of conditional and unconditional performance in Section 2.2.5.

2.2.1 Binary classification

In a binary classification problem, the goal is to assign each input data point $\boldsymbol{x}_i$ to one of two discrete labels, which we will represent by the numbers zero and one. The input data point $\boldsymbol{x}_i$ is a k-dimensional vector $(x_{i1}, x_{i2}, \ldots, x_{ik})$, and the x_{ij}'s are called the *features*. They can be both continuously or discretely scaled.

To assign a data point to a label, we learn a function $\hat{f}\colon \boldsymbol{x}_i \mapsto \hat{y}_i \in \{0, 1\}$ from a sample of observations $(\boldsymbol{x}_i, y_i)$, $i = 1, 2, \ldots, n$, that maps an input data point $\boldsymbol{x}_i$ to its prediction $\hat{y}_i$, and y_i denotes the true label. Throughout this work, the $\hat{}$ symbol will denote quantities that we learned from the data in some way. The main challenge is to learn $\hat{f}$ such that it correctly labels new, unseen data based

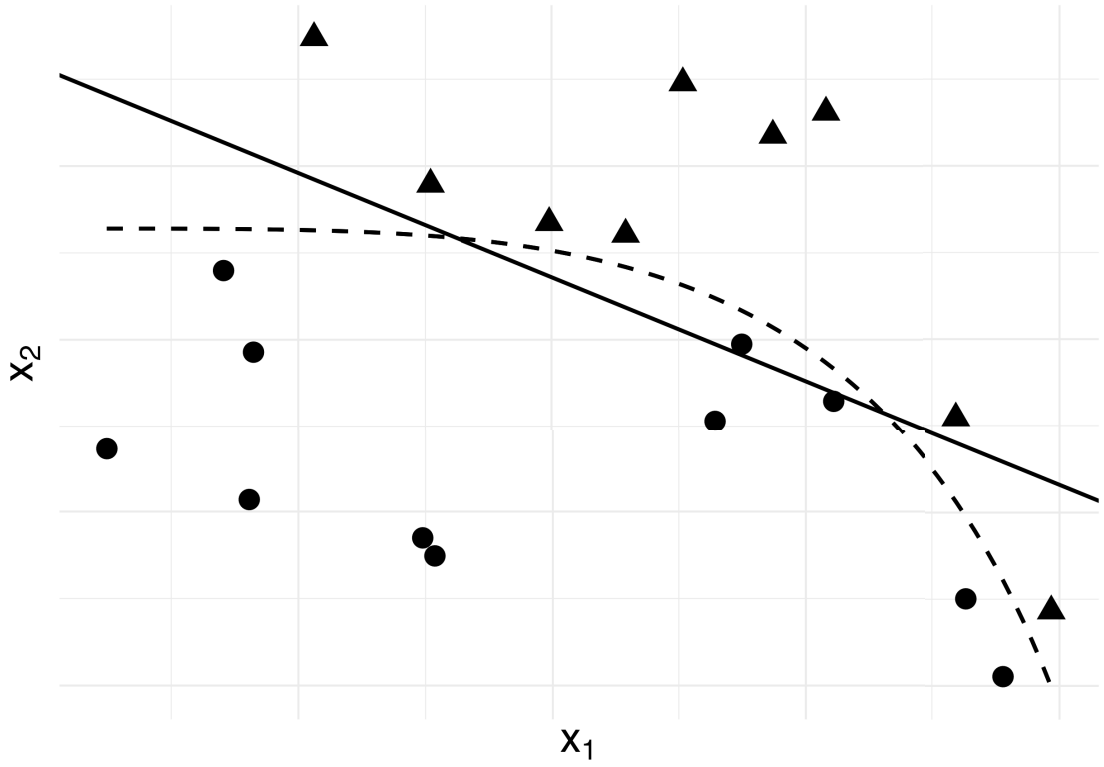

Figure 2.1: Example of a linear (solid line) and a non-linear (dashed line) decision boundary in a binary classification problem with two features

on patterns learned from the sample.

There are numerous ways to learn these patterns, using different algorithms that we will refer to as *models* or *classifiers*. They can broadly be separated into *linear* and *non-linear* classifiers. Linear and non-linear classifiers operate differently on the *feature space*, which is the multi-dimensional space spanned by the features; in other words, the feature space is a geometrical representation of all the possible values that the features can take, and each point in the feature space represents a single data point $\boldsymbol{x}_i$.

In Figure 2.1, we present an example of a two-dimensional feature space, alongside both a linear and a non-linear so-called *decision boundary*, represented by a solid and a dashed line, respectively. The two classes are depicted by circles and triangles. Binary classifiers, in general, try to separate the two classes in the feature space by such a decision boundary, which represents the threshold where the model switches from predicting one class to the other. The shape and complexity of the decision boundary displays how well the model can separate between the classes. In this example, the non-linear classifier achieves perfect separation between the classes.

Linear classifiers Linear classifiers model the decision boundary as a straight line, when there are only two features present, or a hyperplane, when there are more than two features, respectively; that is, linear classifiers assume that the two classes can be separated by a linear function of the features.

One foundational linear classifier is logistic regression. A logistic regression model predicts the probability π_i of an observation to belong to a particular class, say class one, based on its feature values $\boldsymbol{x}_i$. In particular,

$$\pi_i = \frac{e^{\boldsymbol{x}_i\boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}_i\boldsymbol{\beta}}}. \tag{2.1}$$

Note that this model assumes a linear relationship between the features $\boldsymbol{x}_i$ and the coefficients $\boldsymbol{\beta}$. Therefore, logistic regression is considered a linear classifier.

We compare the probability computed in Equation (2.1) to a threshold, typically 0.5, in order to obtain the class prediction. Specifically, if π_i is larger than the threshold, we will predict class one.

Usually, the coefficients $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_k)$ are unknown and we need to estimate them. For a given data set, the coefficients are basically estimated by maximizing the likelihood function using gradient descent to iteratively update the coefficients. For details, see, for example, Hastie et al. (2009).

Non-linear classifiers Non-linear classifiers, in contrast, do not assume linear separability of the feature space. Instead, they model complex dependencies between the features and the classes by learning, for example, curved or even more complex boundaries. This way, non-linear classifiers are more flexible and more versatile.

A notable example of non-linear classifiers are decision trees. A decision tree recursively splits the observations into two sets, based on the feature that provides the best separation, until a stopping criterion is met, such as maximum tree depth or minimum number of observations in a leaf node. This yields a tree-like structure; each internal node represents a feature split, and each leaf node represents a class prediction.

In Figure 2.2 we present an example of a decision tree for the classification of e-mails into the two classes *spam* and *no spam*, due to the features *sender*, *contains suspicious link*, and *number of uppercase characters*. Here, the decision tree learns that, initially, the best separation of the data is based on whether the sender of the e-mail is known or unknown, and in case of the former, the decision tree predicts the e-mail not be spam. In case the sender is unknown, the decision tree will continue to make decisions based on the remaining features. For example,

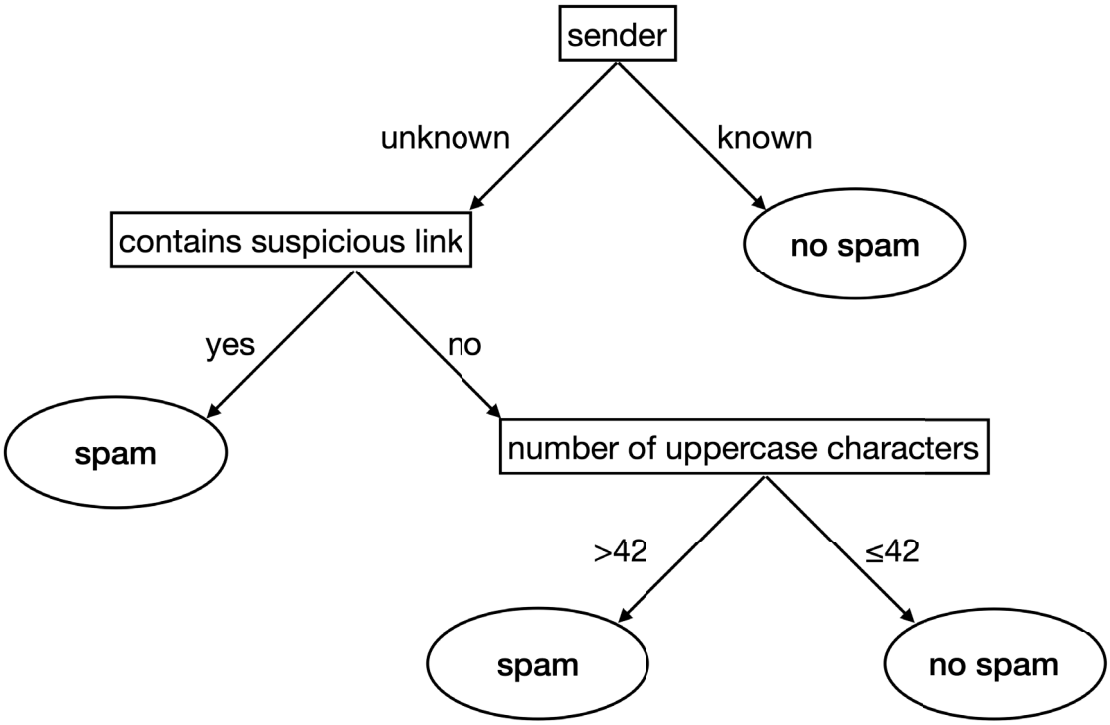

Figure 2.2: Example of a simple decision tree for e-mail spam detection

for the final feature *number of uppercase characters*, the decision tree learns the threshold 42 and predicts the class accordingly.

Linear classifiers are mostly easy to interpret and computationally efficient. In addition to logistic regression, common instances of linear classifiers are linear support vector machines and the least absolute shrinkage and selection operator, which is a variant of logistic regression we will describe in Section 2.3.1. On the other hand, their limitation to linear decision boundaries might lead to ineffective prediction models when the feature space is not really linearly separable.

Non-linear classifiers are in fact able to handle such data. Yet, they are typically less easy to interpret and computationally more expensive. Apart from decision trees, other noteworthy instances of non-linear classifiers are kernelized support vector machines and neural networks, see Murphy (2012) and Goodfellow et al. (2016), respectively, as well as random forests. The latter utilizes multiple individual decision trees, and we will outline how later in Section 2.3.1.

2.2.2 Measures of prediction performance

To make it simple, we will denote the two classes one and zero the *positive class*, and the *negative class*, respectively. Typically, the positive class represents the event that we are interested in, for example, the presence of a medical condition

	$\hat{y}_i = 1$	$\hat{y}_i = 0$	
$y_i = 1$	$\lvert\{\hat{y}_i = 1\} \cap \{y_i = 1\}\rvert$	$\lvert\{\hat{y}_i = 0\} \cap \{y_i = 1\}\rvert$	$\lvert N_+\rvert$
$y_i = 0$	$\lvert\{\hat{y}_i = 1\} \cap \{y_i = 0\}\rvert$	$\lvert\{\hat{y}_i = 0\} \cap \{y_i = 0\}\rvert$	$\lvert N_-\rvert$
	$\lvert\hat{N}_+\rvert$	$\lvert\hat{N}_-\rvert$	n

Table 2.1: General form of a confusion matrix

	$\hat{y}_i = 1$	$\hat{y}_i = 0$	
$y_i = 1$	59	25	$\lvert N_+\rvert = 84$
$y_i = 0$	3	13	$\lvert N_-\rvert = 16$
	$\lvert\hat{N}_+\rvert = 62$	$\lvert\hat{N}_-\rvert = 38$	$n = 100$

Table 2.2: Confusion matrix of the example prediction model

or whether an e-mail is spam. Recall that y_i and $\hat{y}_i$ denote the true and predicted label of the i-th observation from some prediction model, respectively. In the following, we will discuss some common measures to quantify the predictive performance of binary classification models.

A common way to summarize the predictions of a model is to give its *confusion matrix*, which compares the true classes to the predicted ones. In general, a confusion matrix is of the form as presented in Table 2.1. There are four combinations of true class and predicted class that we need to consider: a true-positive prediction occurs if the model correctly predicts the positive class; a false-negative prediction is one where the model incorrectly predicts the negative class; accordingly, a false-positive occurs when the model incorrectly predicts the positive class, and a true-negative is one where the model correctly predicts the negative class. For each of these, there is a corresponding cell in the confusion matrix.

We will introduce some notation in order to denote both the row sums and the column sums of the confusion matrix. Let $N_+ = \{i = 1, 2, \ldots, n \mid y_i = 1\}$ be the index set of observations that belong to the positive class, and, similarly, let $N_- = \{i = 1, 2, \ldots, n \mid y_i = 0\}$ be the index set of observations that belong to the negative class. Also, let $\hat{N}_+ = \{i = 1, 2, \ldots, n \mid \hat{y}_i = 1\}$ denote the index set of positive predictions, and let $\hat{N}_- = \{i = 1, 2, \ldots, n \mid \hat{y}_i = 0\}$ denote the index set of negative predictions.

We will have a running example of a binary prediction model that we will keep coming back to. This model's predictions are summarized in the confusion matrix in Table 2.2.

Prediction accuracy The prediction accuracy is the proportion of correct predictions out of the total number of predictions,

$$\widehat{\mathrm{ACC}} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{\hat{y}_i = y_i\}.$$

Prediction accuracy is a very basic and widely-used measure for the overall performance of a prediction model across all classes. The example prediction model from Table 2.2 has prediction accuracy $(59 + 13)/100 = 72\%$.

However, the prediction accuracy of a model can be misleading in the presence of severe class imbalance. For example, if 95 percent of the observations belong to the positive class and five percent belong to the negative class, a model that always predicts the positive class will have 95 percent prediction accuracy, but performs poorly on the negative class.

Also, prediction accuracy weighs a false-positive prediction equally as bad as a false-negative. Depending on the application, for example if we want to diagnose a medical condition, the implications of a false-positive and a false-negative can vary in severity; a false-negative diagnostic prediction could mean that further cancer screenings are halted and the present condition will not be recognized.

Sensitivity and specificity Two measures that address the issue of class imbalance are the sensitivity,

$$\widehat{\mathrm{SENS}} = \frac{1}{|N_+|}\sum_{i\in N_+} \mathbb{1}\{\hat{y}_i = y_i\},$$

and the specificity,

$$\widehat{\mathrm{SPEC}} = \frac{1}{|N_-|}\sum_{i\in N_-} \mathbb{1}\{\hat{y}_i = y_i\}.$$

of the prediction model. They represent its accuracies in each class. Other names for sensitivity are *recall* or *true positive rate*, and specificity is also known as *true negative rate*.

In the example in Table 2.2, the prediction model has sensitivity $59/84 = 70\%$ and specificity $13/16 = 81\%$. Recall that its prediction accuracy is 72 percent; its sensitivity is only slightly smaller but its specificity is visibly larger, while both, sensitivity and specificity, are quite different.

Revisiting the class-imbalance example from the previous section, a model that always predicts the positive majority class will have a sensitivity of 100 percent, but its specificity will be zero percent.

Balanced prediction accuracy It is not unusual to take the average of the sensitivity and the specificity of a prediction model to summarize both in a single number, the balanced prediction accuracy,

$$\widehat{\text{BACC}} = \frac{1}{2}\left(\frac{1}{|N_+|}\sum_{i\in N_+}\mathbb{1}\{\hat{y}_i = y_i\} + \frac{1}{|N_-|}\sum_{i\in N_-}\mathbb{1}\{\hat{y}_i = y_i\}\right).$$

From the sensitivity and specificity estimates of the prediction model presented in the example from Table 2.2, we obtain a balanced prediction accuracy of $(70\% + 81\%)/2 = 76\%$.

While the balanced prediction accuracy mitigates the issue of class imbalance to some extent, it is still somewhat simplistic and it does not capture all potentially relevant aspects of prediction performance. Therefore, an individual consideration of sensitivity and specificity might be useful, or at least an unequal weighing of sensitivity and specificity,

$$\widehat{\text{WACC}} = \frac{\omega}{|N_+|}\sum_{i\in N_+}\mathbb{1}\{\hat{y}_i = y_i\} + \frac{1-\omega}{|N_-|}\sum_{i\in N_-}\mathbb{1}\{\hat{y}_i = y_i\},$$

where $\omega \in [0, 1]$ is the weight for sensitivity.

Precision The precision of the model is its accuracy among positive predictions,

$$\widehat{\text{PREC}} = \frac{1}{|\hat{N}_+|}\sum_{i\in \hat{N}_+}\mathbb{1}\{\hat{y}_i = y_i\}.$$

Precision is also known as *positive predicted value*.

The precision of the prediction model from the example in Table 2.2 is $59/62 = 95\%$.

Negative predicted value The negative predicted value of the prediction model is its accuracy among negative predictions,

$$\widehat{\text{NPV}} = \frac{1}{|\hat{N}_-|}\sum_{i\in \hat{N}_-}\mathbb{1}\{\hat{y}_i = y_i\}.$$

The prediction model from the example in Table 2.2 has a negative predicted value of $25/38 = 66\%$.

Performance measure	Estimate
Precision	95%
Specificity	81%
Balanced prediction accuracy	76%
Prediction accuracy	72%
Sensitivity	70%
Negative predicted value	66%

Table 2.3: Performance estimates of the example prediction model presented

Choice of measure Table 2.3 lists all the performance estimates from the previous sections of the example prediction model. It is hard to say whether its predictive performance is bad, mediocre, good, or even very good; this heavily depends on what we are particularly interested in and the context in which the model is applied.

For example, when diagnosing a medical condition, it may be more important to identify all patients who are likely to have that condition, even at the cost of some patients being identified as having the condition when they actually do not. This means to aim for a prediction model with high sensitivity. The model from the example in Table 2.2, however, only has a sensitivity of 70 percent, so it might be worth trying to find a predictive model with higher sensitivity.

In another example, when we try to detect spam e-mails, marking a legitimate e-mail as spam can cause important information not being communicated. Here, it is probably wiser to make sure that e-mails marked as spam are in fact spam, even at the cost of some spam e-mails not being detected. This means to aim for a prediction model with a high precision, which applies to the model from the example in Table 2.2, as its precision is 95 percent.

Note that all the presented prediction measures can be written as means. This property will be important later. We will come back to that.

2.2.3 Generalization performance

In the previous section, we explored a variety of measures that capture different aspects of predictive performance. They allow for a deeper understanding of a prediction model's strengths and weaknesses.

However, another important aspect to consider when evaluating prediction performance is the method of evaluating the model itself. A key focus here is *gen-*

eralization performance, which refers to the model's performance on new, unseen data. In rare cases, we can keep a separate hold-out set of data that we use for model fitting; its only purpose is to test the model's performance, providing an independent evaluation of its ability to generalize to new, unseen data. However, more often than not, we only have a single data set at hand that we need to use wisely for both model fitting and estimation of generalization performance.

One approach to do so is data splitting. This is the most straightforward approach to estimate generalization performance. We split the sample at hand randomly into two parts, the *training* and the *test* set. We use the observations in the training set exclusively to fit the model and the observations in the test set exclusively to estimate the model's generalization performance, for example, its prediction accuracy. This way, no information from the test set is used during model fitting.

A major drawback of data splitting is that we need to irrevocably allocate the observations to either the training or the test set. This is particularly challenging when only little data is available. By allocating a greater fraction towards model training, the estimate of generalization performance gets less reliable, and the allocation of a greater fraction towards performance estimation deteriorates the model's predictive performance, as it learns from less data.

Another approach is cross-validation, which we will address in the next section. A third example is *bootstrap aggregating* or *bagging* that we will not discuss here; see Murphy (2012) for more information.

2.2.4 Cross-validation

An alternative option to data splitting is cross-validation, which might be the most popular and widely-used approach to estimate the generalization performance of a prediction model. Instead of a fixed allocation of the observations in the sample at hand into a training and a test set, each is used for both, but not at the same time.

Rather, we split the sample into $K > 1$ equal-sized parts, which are also called *folds*. In each iteration, one of the folds is held out and serves as the test set. The remaining $K - 1$ folds are used to train the prediction model, which is then used to predict the observations in the test fold. This way, for each fold, we obtain an estimate of prediction performance. Finally, the average of these K numbers is an estimate of the generalization performance of the prediction model.

More specifically, let $\mathcal{I}_\ell$ denote the index set of the observations that are allocated to the ℓ-th fold. Let $\hat{\boldsymbol{\beta}}^{(-\ell)}$ denote the coefficient vector that we obtain from training the prediction model on all but the ℓ-th fold, and let the associated

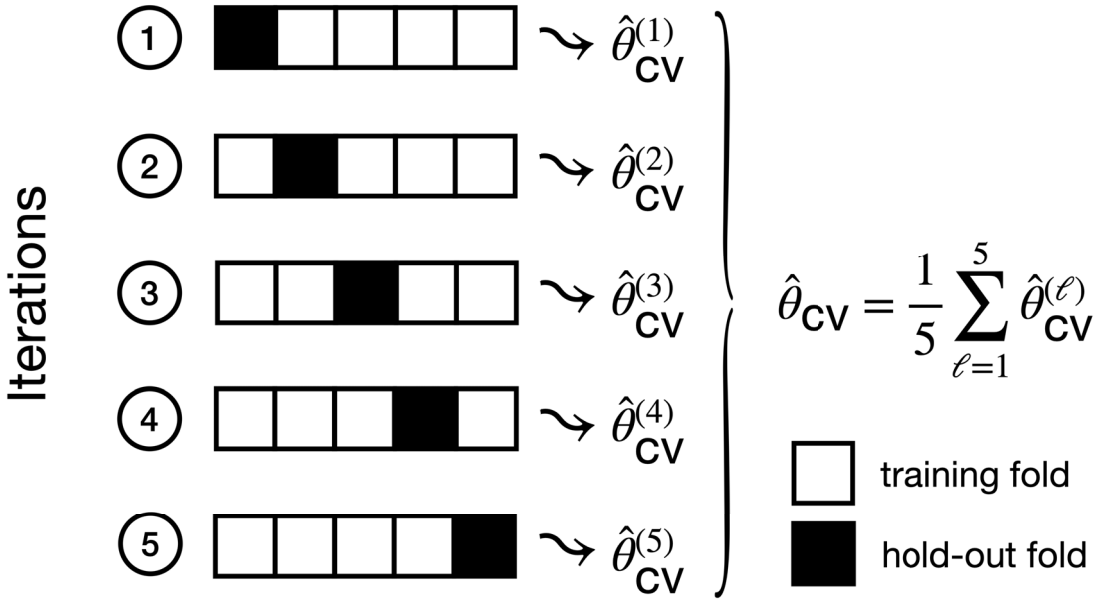

Figure 2.3: Illustration of 5-fold cross-validation

prediction function be denoted by $\hat{f}\colon (\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}^{(-\ell)}) \mapsto \hat{y}_i \in \{0, 1\}$. In addition, let $u\colon (\hat{y}, y) \mapsto \{0, 1\}$ be a utility function that formalizes how well $\hat{f}$ predicts. For example, $u(\hat{y}_i, y_i) = \mathbb{1}\{\hat{y}_i = y_i\}$ is associated with prediction accuracy, and is a typical choice in a classification setting.

Then, we estimate the prediction performance in the ℓ-th fold by

$$\hat{\theta}^{(\ell)}_{\text{CV}} = \frac{1}{|\mathcal{I}_\ell|} \sum_{i \in \mathcal{I}_\ell} u[\hat{f}(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}^{(-\ell)}), y_i],$$

and the average across the K folds

$$\hat{\theta}_{\text{CV}} = \frac{1}{K} \sum_{\ell=1}^{K} \hat{\theta}^{(\ell)}_{\text{CV}} = \frac{1}{n} \sum_{\ell=1}^{K} \sum_{i \in \mathcal{I}_\ell} u[\hat{f}(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}^{(-\ell)}), y_i]$$

estimates the generalization performance.

There are some standard choices for the number of folds K, which include $K = 5$ and $K = 10$. In Figure 2.3, 5-fold cross-validation is illustrated. Another common option is to take $K = n$. This is called *leave-one-out cross-validation*. All of these choices lead to different levels of bias and variance in the estimation. For a discussion, see Hastie et al. (2009), as we will not get into more detail here.

Note that the cross-validation scheme also allows for the computation of a

variance estimate

$$\hat{\sigma}^2_{\text{CV}} = \frac{1}{n-1} \sum_{\ell=1}^{K} \sum_{i \in \mathcal{I}_\ell} \{u[\hat{f}(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}^{(-\ell)}), y_i] - \hat{\theta}_{\text{CV}}\}^2.$$

We will revisit this estimate later.

Cross-validation can also be used for model selection. For that, the cross-validation performances $\hat{\theta}_{\text{CV},1}, \hat{\theta}_{\text{CV},2}, \ldots, \hat{\theta}_{\text{CV},m}$ are computed and compared to each other. Usually, we would select the best-performing among them for future use.

2.2.5 Conditional and unconditional performance

There are different kinds of generalization performances that we might be interested in. When we evaluate the *unconditional* prediction performance of a model, we estimate the average performance of the fitting algorithm across a wide range of hypothetical same-sized data sets from the same underlying distribution. Here, we do not account for specific characteristics of the sample. Rather, the goal is to ensure that the model is able to perform well across different instances of training data. In this way, unconditional performance reflects the interaction between the model and the data-generating process. In practice, it is often utilized to compare different fitting algorithms.

On the other hand, when we are interested in the *conditional* prediction performance, we estimate the generalization performance of the prediction model trained on the sample at hand. Thus, it depends on the particularities of the sample. In many practical applications, though, conditional performance is often the more relevant measure, because it reflects how well we can expect the model to predict future observations, rather than its theoretical average performance across hypothetical instances.

To further illustrate the differences, we will give a mathematical description of conditional and unconditional performance. Suppose we have a utility function $u\colon (\hat{y}, y) \mapsto \{0, 1\}$, and let $\hat{f}\colon (\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}) \mapsto \hat{y}_i \in \{0, 1\}$ be the function that predicts y_i from its corresponding feature vector $\boldsymbol{x}_i$ and estimated coefficients $\hat{\boldsymbol{\beta}}$. Moreover, let $\boldsymbol{X}$ and $\boldsymbol{y}$ denote the feature matrix and vector of true class labels of the sample at hand, respectively. Then, the conditional prediction performance is given by

$$\text{PERF}_{\boldsymbol{X},\boldsymbol{y}} = \mathbb{E}\{u[\hat{f}(\tilde{\boldsymbol{x}}, \hat{\boldsymbol{\beta}}), \tilde{y}] \mid \boldsymbol{X}, \boldsymbol{y}\},$$

where $(\tilde{\boldsymbol{x}}, \tilde{y})$ is an independent new observation from the same distribution.

While $\text{PERF}_{\boldsymbol{X},\boldsymbol{y}}$ is a random quantity that depends on the sample $(\boldsymbol{X}, \boldsymbol{y})$ at

hand, in contrast, the unconditional performance

$$\text{PERF} = \mathbb{E}(\text{PERF}_{\boldsymbol{X},\boldsymbol{y}}).$$

is its expected value across multiple hypothetical samples.

For the further course of this work, we will focus our interest on conditional performance. In fact, since Bates et al. (2024) argued that the cross-validation estimate of generalization performance $\hat{\theta}_{\text{CV}}$ estimates PERF rather than the $\text{PERF}_{\boldsymbol{X},\boldsymbol{y}}$, we will not be able use cross-validation to evaluate prediction models. There are, however, proposals by Bates et al. (2024) and Tsamardinos et al. (2018) to estimate conditional performance using modified variants of cross-validation. We will review them later in Part II of this work. In any case, cross-validation often is a valuable selection tool in the presence of multiple competing prediction models.

2.3 Model selection

Typically, when fitting prediction models in supervised learning applications, there are multiple candidate models available that could potentially provide reliable predictions. This is because different models work well for different types of data, and even within certain applications, there is rarely a single model that is universally best suited to learn from different samples. In addition, a model might perform differently in terms of different measures of prediction performance. It is a major and intricate task to select a prediction model for future use among the candidate models.

The abundance of candidate models is due on the one hand to the fact that there are so many different algorithms that can learn prediction models for binary classification, and on the other hand due to fact that, within the same algorithm, we can often adjust so-called *hyperparameters*, which can lead to completely different prediction models. These hyperparameters usually control the *complexity* of the model. For example, hyperparameters for a decision tree include the maximum tree depth and the minimum number of observations in a leaf node.

Complexity refers to the ability of a model to capture intricate or complicated interactions among the features and between the features and the labels. Less complex models tend to produce reliable predictions when these relationships are rather simple or when there are not many features present. More complicated data requires more complex models. But as complexity increases, so does the risk of *overfitting*, that is, the model fits the data in the sample too closely. It then fits to the noise and even potential outliers. Overfitting generally leads to poor generalization performance. The challenge is, thus, to select a prediction model

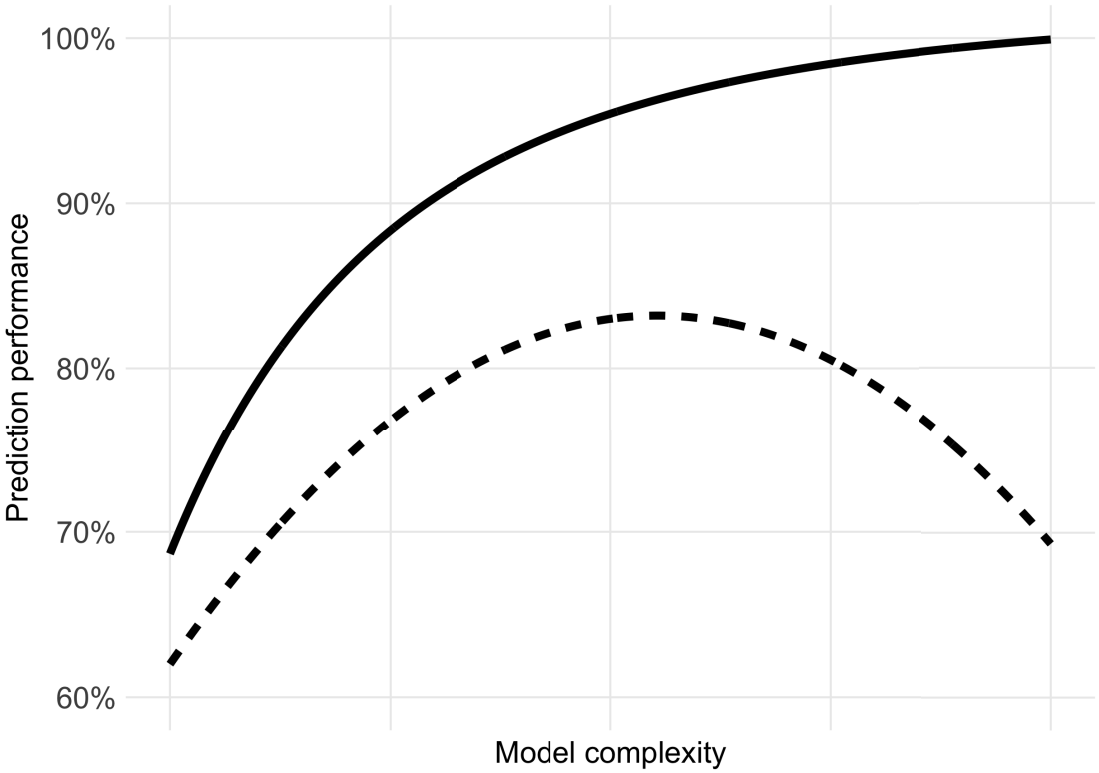

Figure 2.4: Behavior of in-sample (solid line) and out-of-sample prediction performance (dashed line) as model complexity increases

for future use that balances complexity and performance on new, unseen data.

In Section 2.3.1, we will show how to track the occurrence of overfitting and present modifications on both logistic regression and decision trees that mitigate this risk. Then, in Section 2.3.2, we will present both a well-established and a recently proposed approach by Westphal & Brannath (2020) to use the sample for efficient model selection. In this context, we will also address the phenomenon of selection-induced over-optimism.

2.3.1 Overfitting

Generally speaking, overfitting occurs when a model does not only capture the underlying patterns in the data, but also learns from the random fluctuations or noise. As a result, the model becomes overly sensitive to the specific data it is fitted to and, consequently, performs relatively poorly on new, unseen data. The risk of overfitting is particularly high when there are too many features in the data relative to the number observations, or when a model is complex enough to be able to memorize the particularities of the data instead of learning the generalizable trends.

We can track overfitting by comparing a model's *in-sample performance* with

its *out-of-sample performance*. The former means the performance on the data it is fitted to, while the latter means the performance on new, unseen data.

We illustrate the behavior of the in-sample and out-of-sample performance when model complexity increases in Figure 2.4. When model complexity is low, both in-sample and out-of-sample performance are also low, because the model essentially underfits, that is, it fails to capture the underlying patterns. As the complexity increases, both performances increase, because the model becomes better at learning the underlying trends. However, beyond a certain point, while in-sample performance naturally continues to improve, the out-of-sample performance begins to decline, signaling overfitting. The optimal prediction model is the one that achieves the highest out-of-sample performance.

There are basically two approaches to handle overfitting. The first is, as we usually only have a single set of data at hand, to utilize it wisely to obtain estimates of both in-sample and out-of-sample performance. One approach is to use cross-validation. The cross-validation estimates of prediction performance generally give a fairly good indication of which of the candidate models generalize well. Note that *well* means in relative terms and not in absolute, since cross-validation estimates the unconditional performance. Another way is to split the data set at hand into two, use one part for model fitting and in-sample performance estimation, and the other for prediction and out-of-sample estimation.

The second strategy is to penalize the complexity of a regression model, and, therefore, discouraging it from fitting noise or overly complex patterns in the data. In Section 2.2.1 we established logistic regression as a simple approach to classification. While the resulting coefficient estimates are easy to interpret, logistic regression can struggle with overfitting in high-dimensional data sets or in the presence of irrelevant features. Adding a penalty to the complexity is called *regularization*. One such instance is ℓ_1 regularization, also known in this context as the *Least Absolute Shrinkage and Selection Operator*, or LASSO, for short.

LASSO The LASSO penalty term is proportional to the sum of the absolute values of the estimated coefficients $\hat{\beta}_j$. It, thus, penalizes large coefficients and shrinks some of the coefficients of less important features exactly to zero, as shown, for example, in Section 13.3 in Murphy (2012). Shrinking coefficients exactly to zero effectively excludes the corresponding features from the prediction model, which simplifies it and makes it easier to interpret.

The strength of regularization affects the number of features that are excluded from the model and is controlled via the hyperparameter λ. Admissible values for

λ are larger than zero but smaller than

$$\lambda_{\max} = \min \{\lambda > 0 \mid \hat{\boldsymbol{\beta}}_\lambda = \mathbf{0}\}, \tag{2.2}$$

the smallest value for the hyperparameter such that none of the features are selected into the prediction model, and $\hat{\boldsymbol{\beta}}_\lambda$ is the estimate of the true coefficient vector $\boldsymbol{\beta}$ from a LASSO regression with regularization parameter λ.

The LASSO works particularly well in situations where there are many features, but only a small subset of them are truly informative for prediction. However, if the true underlying dependencies between the features and the true class labels are complex or the signal is in fact not sparse, the LASSO may perform poorly because it eliminates features. In such cases, a combination of ℓ_1 and ℓ_2 penalty terms might be a better option, leading to the *elastic net*.

Elastic net The ℓ_2 penalty is proportional to the sum of squares of the estimated coefficients $\hat{\beta}_j$ and encourages the coefficients to be evenly small, but in contrast to the ℓ_1 penalty, it does not force them to be exactly zero. This is useful when we believe that most of the features are informative for the true class label, but some may be noisy, have only little influence, or when some features are highly correlated. Again, the strength of regularization is controlled via a hyperparameter.

Combining the ℓ_1 and ℓ_2 penalties yields the elastic net, which has two hyperparameters $\lambda \in [0, \lambda_{\max}(\gamma)]$ and $\gamma \in [0, 1]$ that control the overall strength of regularization and the balance between the ℓ_1 and ℓ_2 penalty, respectively. In particular, the elastic net penalty can be written as $\lambda[\gamma\|\boldsymbol{\beta}\|_1 + (1-\gamma)\|\boldsymbol{\beta}\|_2^2]$.

Random forest In addition to logistic regression, we mentioned decision trees as an example of a non-linear classifier in Section 2.2.1. While a single decision tree can separate the classes based on feature splits, it is likely to overfit, because the splits can be highly specific, capturing noise rather than underlying patterns. This can happen especially when the tree is deep and complex.

A random forest is a so-called *ensemble learning* method that builds on the idea of decision trees. Ensemble learners combine multiple individual and typically simple prediction models. They aggregate their predictions in order to improve the overall predictive performance.

In particular, a random forest combines multiple decision trees. Random forests address overfitting by averaging the predictions from a collection of de-

cision trees. For each such trees, we randomly draw subsets with replacement from the observations (so-called *bootstrap samples*; we will discuss bootstrap sampling in great detail in Chapter 4) and a random subset of the features. This way, we obtain multiple decision trees and multiple predictions per observation. The final predicted class for an observation is determined by a majority vote among the predictions from the individual trees.

Random forests are useful in prediction problems with many features that have complex dependencies. They usually yield high predictive performance, but this comes at the expense of higher computational cost and worse interpretability than, for instance, a single decision tree or a LASSO model.

Note that the above strategies only help to handle overfitting. They do not directly address model selection or generalization performance. When we need to do both model selection and evaluation, even more careful consideration is required.

2.3.2 Selection-evaluation pipelines

In case we need to first select a model from a collection of competing prediction models before we can evaluate it, the predominant recommendation in the literature is to split the sample into three parts, see, for example, Goodfellow et al. (2016), Hastie et al. (2009), Japkowicz & Shah (2011), Murphy (2012), or Raschka (2018).

Training set The first part, the *training set*, is used to fit all the candidate models in order to learn the underlying patterns and allow the models to adjust in order to accurately map the features to the correct labels. It is typically the largest of the three parts and contains usually at least 50 percent of the observations in the sample.

Validation set The second part is called the *validation set*. Its purpose is to find values for the candidate models' hyperparameters that lead to a high prediction performance. This is known as *tuning*. Keeping the validation set separate from the training set mitigates the risk of overfitting, and ensures that the models do not memorize the training data, but rather generalize well to new, unseen data.

Typically, we prespecify all the hyperparameter values that we want to test, fit the corresponding models on the training set, and compare them based on their predictions in the validation set. Alternatively, we could also validate the

model iteratively by repeatedly going back and forth between the training and the validation set, adjusting the hyperparameters. In any case, the prevailing recommendation is to only select a single model to proceed with to the third and last part.

Evaluation set The remaining portion of the data is called the *evaluation set*. We utilize it to estimate generalization performance, as its observations are truly unseen by the prediction models. Once a model is selected for future use, we fit it again to both the training and the validation data, and apply it to the evaluation set in order to compare the predictions to the true labels. This way, we obtain an unbiased estimate of the selected model's generalization performance, using, for example, any of the measures presented in Section 2.2.2.

The reason to keep the evaluation set separate from both the training and validation set is to prevent *data leakage*, that is, information from outside model development accidentally influencing the model. Something similar occurs when we base the model selection on the evaluation performance. This leads to *selection-induced over-optimism*, that is, inflated performance estimates, and creates a false sense of the reliability of the prediction model.

In many real-world applications, we identify multiple promising models during validation, and often some perform almost equally well. Additionally, another promising model might be easier to interpret or to compute than the most-promising one. When we acknowledge that the performance estimates from the validation set are subject to variability and do not account for potential changes in the distribution of future observations, we might wonder if we should perform the model selection based on the evaluation performances after all, despite the selection-induced over-optimism.

Recent work by Westphal & Brannath (2020) showed that it is indeed beneficial to shift the model selection from the validation phase to the evaluation phase, yielding final models with better prediction performance. The authors also proposed a way to deal with the arising selection-induced over-optimism. This idea is fundamental to the present work and we will elaborate on the details later. We will call this approach the *proposed machine learning selection-evaluation pipeline*, in contrast to the *default pipeline* explained before.

In Figure 2.5, we illustrate both the default and the proposed pipeline. From the training set, we obtain prediction models that we represent by their respective estimated model parameters $\hat{\boldsymbol{\beta}}_1, \hat{\boldsymbol{\beta}}_2, \ldots, \hat{\boldsymbol{\beta}}_M$. Using the validation set, we estimate the validation performances $\hat{\theta}_{V,1}, \hat{\theta}_{V,2}, \ldots, \hat{\theta}_{V,M}$. Alternatively, depending on the

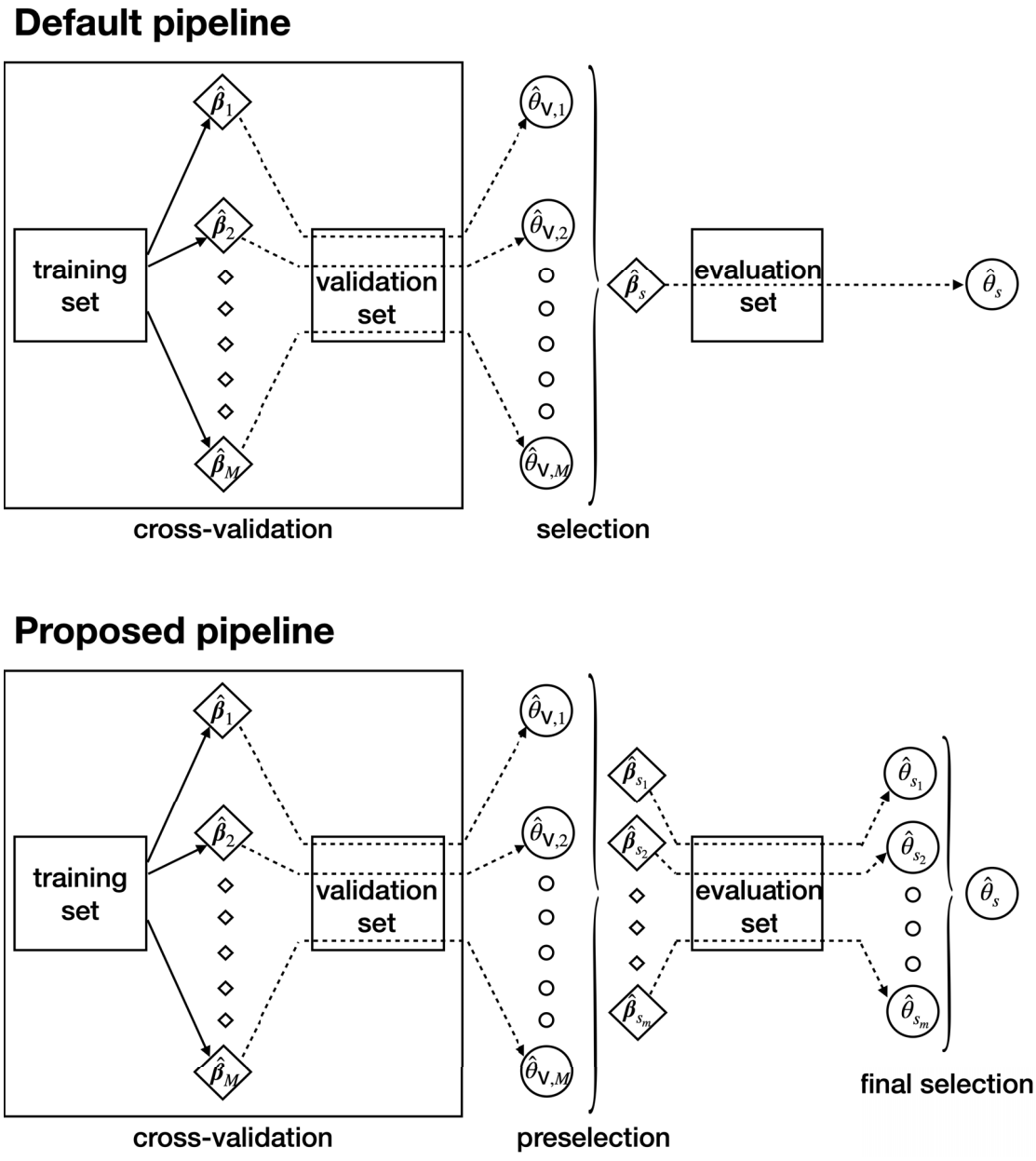

Figure 2.5: Default and proposed machine learning selection-evaluation pipelines

specific application, we could also use cross-validation to estimate the $\hat{\theta}_{V,i}$'s. The next step is where both pipelines differ. In the default pipeline, we select the most-promising model and estimate its generalization performance in the evaluation set; in the proposed pipeline, we *preselect* multiple models and select the final model $s \in \{s_1, s_2, \ldots, s_m\}$ among them, based on the performance estimates $\hat{\theta}_{s_1}, \hat{\theta}_{s_2}, \ldots, \hat{\theta}_{s_m}$ from the evaluation set. Note that the estimate $\hat{\theta}_s$ this way is inflated due to selection-induced over-optimism.

In Chapter 3, we will introduce the statistical concepts needed to better understand the proposed selection-evaluation pipeline.

Chapter 3

Statistical Inference

In Chapter 2, we set the stage for our further considerations: Among a collection of candidate binary classification models, we want to select one or more promising prediction models and assess their generalization performance. This chapter will provide all the fundamental concepts to translate this task into a statistical inference problem.

We will begin this chapter in Section 3.1 with the introduction of statistical hypothesis testing. This is a foundational mathematical framework for evaluating the validity of claims or theories about so-called *populations* of subjects. Building on this, in Sections 3.2 and 3.3, we will turn to test statistics, p-values, and confidence intervals, which provide quantitative measures of how likely the claims are true. In Section 3.4, we will explore in which ways statistical inference is affected when we investigate multiple claims simultaneously. This will ultimately be the key to translating the proposed machine learning selection-evaluation pipeline into a statistical inference framework, which we will do in Section 3.5, before concluding this chapter.

This chapter only briefly addresses the essential aspects of statistical inference, and comprehensive discussions can be found elsewhere. Key references for Sections 3.1, 3.2, and 3.3 include Lehmann & Romano (2005). Section 3.4 loosely follows Dmitrienko & Hsu (2014), and Dickhaus (2014) provides a detailed presentation.

3.1 Hypothesis testing

Statistical hypothesis testing is a general and rigorous statistical framework for drawing conclusions about unknown parameters based on observed data. It contrasts two opposing hypotheses, the so-called *null hypothesis* $\mathcal{H}^0$ and *alternative hypothesis* $\mathcal{H}^A$, only one of which can be in fact true. We use a *test statistic* to

summarize the strength of evidence in the observed data against the null hypothesis $\mathcal{H}^0$. If the evidence is large enough, we may decide to reject $\mathcal{H}^0$ and conclude that we believe the alternative $\mathcal{H}^A$ to be true.

For example, suppose we consider a newly developed model for predicting the success of some medical treatment, and the reference model has a prediction accuracy of, say, $\xi = 90\%$. We could ask whether the new model has higher or lower prediction accuracy θ. We could aim to discover that the newly developed model has higher predictive performance than the reference, and we would only want to replace the reference model when we are pretty sure about that. Because hypothesis testing treats the null hypothesis $\mathcal{H}^0$ and the alternative $\mathcal{H}^A$ asymmetrically in that it collects evidence against the null hypothesis from the observed data, the null hypothesis should be that the predictive performance of the new model is at most the reference performance, that is, $\mathcal{H}^0 \colon \theta \leq \xi$. If we could reject $\mathcal{H}^0$ with the observed data, we would conclude that we believe the opposite is true, that is, that the new model is truly better in predicting the success of the medical treatment than the reference, or $\mathcal{H}^A \colon \theta > \xi$.

We need to be aware of this asymmetry when we define $\mathcal{H}^0$ and $\mathcal{H}^A$. In general, $\mathcal{H}^0$ represents the default state of belief about the parameter of interest or the less severe outcome, while $\mathcal{H}^A$ corresponds to a discovery with potentially extensive consequences.

A null hypothesis such as $\mathcal{H}^0 \colon \theta \leq \xi$ from above is referred to as a *one-sided hypothesis*, because the alternative is one-directional relative to the reference value ξ; if we rejected $\mathcal{H}^0 \colon \theta \leq \xi$, we would believe that in truth the predictive performance of the new model is larger than the reference. Similar applies when testing $\mathcal{H}^0 \colon \theta \geq \xi$ against $\mathcal{H}^A \colon \theta < \xi$.

In general, a null hypothesis can also be *two-sided*. For example, when we test $\mathcal{H}^0 \colon \theta = \xi$ against $\mathcal{H}^A \colon \theta \neq \xi$, we call $\mathcal{H}^0$ a two-sided hypothesis. Here, the alternative is two-directional relative to ξ, and rejecting $\mathcal{H}^0$ does not yield any information whether we believe that $\theta > \xi$ or $\theta < \xi$. In the example above, a two-sided hypothesis would correspond to the question whether the newly developed model has the same predictive performance as the reference model. While this can indeed be an interesting question at times, we will mainly consider one-sided hypotheses in this work.

3.2 Test statistics and p-values

To quantify the level of evidence in the observed data against the null hypothesis $\mathcal{H}^0 \colon \theta \leq \xi$, we construct a so-called *test statistic* T. The specific choice of T depends in different ways on the null hypothesis we want to test. A detailed

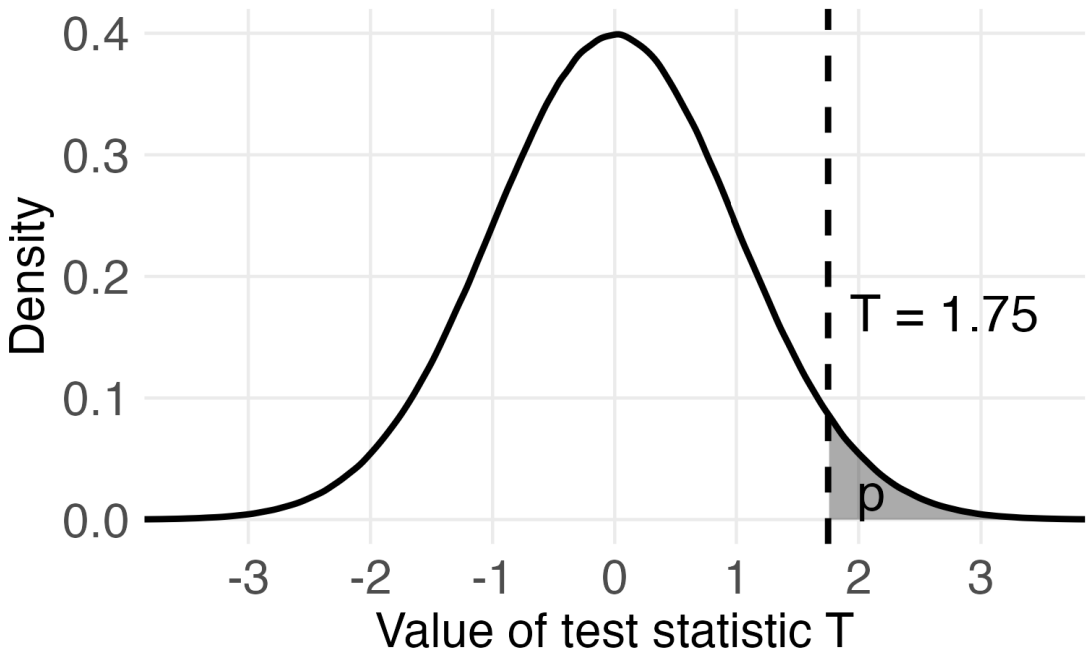

Figure 3.1: Test statistic T and p-value p when T follows a standard-normal distribution under the null hypothesis

summary of the various test statistics and their use is provided in Lehmann & Romano (2005). In many cases, however, test statistics are constructed in such a way that larger values of T correspond to stronger evidence against $\mathcal{H}^0$. We assume that the same applies throughout this work.

The level of evidence against $\mathcal{H}^0$ is measured in terms of the probability of obtaining at least the observed value of T under the assumption that $\mathcal{H}^0$ is actually true. This involves knowledge of the distribution of T given that $\mathcal{H}^0$ is true. This probability p is called the *p-value*. If p is small, the evidence from the observed data against $\mathcal{H}^0$ is strong.

Suppose, for example, that the test statistic T follows a standard-normal distribution under $\mathcal{H}^0$, and from the observed data we compute $T = 1.75$, which corresponds to the 96 percent quantile of the standard-normal distribution, that is, 96 percent of the probability mass lies below 1.75. Thus, the probability to observe a value of T that is 1.75 or even larger is only $p = 4\%$. We illustrate this in Figure 3.1.

In general, we decide against $\mathcal{H}^0$ if the observed value of the test statistic T is larger than a critical value c_α, which we choose such that we reject $\mathcal{H}^0$ when it is actually true only with a small probability α, typically $\alpha = 5\%$. This probability α is called the *significance level*.

In particular, recall that in the example above, we assume that T follows a

standard-normal distribution under $\mathcal{H}^0$. Then, if $\mathbb{P}_0$ denotes probability under $\mathcal{H}^0$ and Φ is the cumulative distribution function of the standard-normal distribution, we want

$$\alpha = \mathbb{P}_0(T \geq c_\alpha) = 1 - \mathbb{P}_0(T < c_\alpha) = 1 - \Phi(c_\alpha).$$

Inverting $\alpha = 1 - \Phi(c_\alpha)$ yields $c_\alpha = \Phi^{-1}(1 - \alpha)$, that is, the critical value is the $(1 - \alpha) \cdot 100\%$-quantile of the standard-normal distribution.

We follow the same idea to find the critical values for different test problems with different distributional assumptions.

3.3 Confidence intervals

An alternative option to measure the evidence against a null hypothesis is confidence intervals, which contain a range of plausible values for the true parameter θ, with a certain level of confidence. In particular, when we consider a so-called $(1 - \alpha) \cdot 100\%$-*confidence interval* for some population parameter θ, we mean the following. If we drew repeated samples from the same population that our observed data is from, and for each sample we computed a $(1-\alpha)\cdot 100\%$-confidence interval, we would expect that the proportion of intervals that contain θ is $(1 - \alpha) \cdot 100\%$.

In practice, however, we usually only have a single sample of observations at hand, and we need to make assumptions on their distribution. This is why in finite samples, even if the so-called *nominal* confidence level is $(1 - \alpha) \cdot 100\%$, the actual *observed* or *estimated* confidence level of the estimated interval may be considerably lower, when the assumptions are not fully or only asymptotically met. It will be a major point of discussion how to estimate the confidence level of an interval method later in Section 6.1.2 in Part II of this work.

Confidence intervals that fall below the nominal level $(1 - \alpha) \cdot 100\%$ are called *anti-conservative*. They are often too narrow, creating an inaccurate perception of the range of plausible values. From an inferential point of view, anti-conservative confidence intervals lead to an increased risk to falsely reject a null hypothesis.

On the other hand, confidence intervals that substantially exceed the nominal level are called *conservative*. They are often too wide, overestimating the present uncertainty and, thus, reducing the precision. This could result in failing to reject a null hypothesis when it is actually false.

Compared to test statistics or p-values, confidence intervals offer a better understanding about the test decision. The range of plausible values they provide gives more information about the effect size, its possible range, and the degree of uncertainty. However, independent of whether we test a null hypothesis with a test statistic or with a confidence interval, the test decision is the same.

3.3.1 Duality to hypothesis testing

In many cases, confidence intervals are based on test statistics. To illustrate this, suppose that we test $\mathcal{H}_\xi^0\colon \theta = \xi$ against $\mathcal{H}_\xi^A\colon \theta \neq \xi$ using the test statistic T_ξ. Note that $\mathcal{H}_\xi^0$ is two-sided. Hence, we need to consider deviations of T_ξ from the critical value in both directions. When we assume that T_ξ follows a standard-normal distribution under $\mathcal{H}_\xi^0$ and reject $\mathcal{H}_\xi^0$ if $|T_\xi| \geq \Phi^{-1}(1-\alpha/2)$, then the probability to falsely reject $\mathcal{H}_\xi^0$ is

$$\begin{aligned}\mathbb{P}_\xi[|T_\xi| \geq \Phi^{-1}(1-\alpha/2)] &= \mathbb{P}_\xi[T_\xi \geq \Phi^{-1}(1-\alpha/2)] + \mathbb{P}_\xi[T_\xi \leq -\Phi^{-1}(1-\alpha/2)] \\ &= 2\,\mathbb{P}_\xi[T_\xi \geq \Phi^{-1}(1-\alpha/2)] = 2\left\{1 - \Phi[\Phi^{-1}(1-\alpha/2)]\right\} \\ &= \alpha,\end{aligned}$$

because of the symmetry of the standard-normal distribution, and $\mathbb{P}_\xi$ denotes the probability under $\mathcal{H}_\xi^0$.

In order to construct a confidence interval, we do not only test one specific ξ this way, but all potential values for $\xi \in \mathbb{R}$, and we reject all ξ's for which $|T_\xi| \geq \Phi^{-1}(1-\alpha/2)$. Those ξ's that we cannot reject remain as the plausible values for θ and form the confidence interval

$$\textsc{ci}(1-\alpha) = \{\xi \in \mathbb{R}\colon |T_\xi| \leq \Phi^{-1}(1-\alpha/2)\}.$$

Therefore, we can reject $\mathcal{H}_\xi^0$ either if $|T_\xi| \geq \Phi^{-1}(1-\alpha/2)$ or if $\xi \notin \textsc{ci}(1-\alpha)$. This is known as the *duality between hypothesis testing and confidence intervals*. Because the probability to falsely reject the null hypothesis is equal to α, $\textsc{ci}(1-\alpha)$ is a $(1-\alpha)\cdot 100\%$-confidence interval, and, hence, $\textsc{ci}(1-\alpha)$ is said to have *coverage probability* $(1-\alpha)\cdot 100\%$.

We will conclude our considerations on confidence intervals in the following Section 3.3.2, where we present some popular standard confidence intervals.

3.3.2 Standard methods

In this section, we will present some standard approaches for constructing confidence intervals when the parameter of interest is a binomial proportion. Hence, let $X_1, X_2, \ldots, X_n \in \{0,1\}$ denote an i. i. d. sample from a Bernoulli distribution. Our goal is to make statistical inferences about the unknown success probability θ using a confidence interval. Specifically, we want to test whether the true success probability θ exceeds some reference value ξ, that is, the null hypothesis we test is $\mathcal{H}^0\colon \theta \leq \xi$.

First, we need to introduce some notation. Let $\hat{\theta}_n = n^{-1}\sum_{i=1}^n X_i$ denote the

estimated success probability, $\hat{\sigma}_n^2 = \hat{\theta}_n(1-\hat{\theta}_n)/n$ is a variance estimate, and $z_{1-\alpha}$ denotes the $(1-\alpha)\cdot 100\%$-quantile of the standard-normal distribution. The following lower confidence limits all induce an interval with nominal coverage probability $(1-\alpha)\cdot 100\%$.

Wald normal approximation interval This popular and widely-used method to estimate confidence intervals assumes that the parameter of interest follows a normal distribution, which is often the case in applications, at least asymptotically. The Wald lower confidence limit is given by

$$\hat{\theta}_n - z_{1-\alpha}\,\hat{\sigma}_n,$$

and we obtain it by testing ξ for inclusion in the confidence interval using the dual test $(\hat{\theta}_n - \xi)/\hat{\sigma}_n \leq z_{1-\alpha}$. It is known to sometimes struggle to reach the nominal coverage probability, especially in small samples or when $\hat{\theta}_n$ is near zero or one.

Wilson interval This interval constitutes an improvement over the Wald interval in many respects as it allows for asymmetric intervals, incorporates a continuity correction, and can be applied to small samples, as well. In addition, it offers better performance than the Wald interval when $\hat{\theta}_n$ is close to zero or one. While the Wald interval uses the estimated standard error $\hat{\sigma}_n$, the Wilson interval incorporates the null hypothesis variance $\xi(1-\xi)/n$; that is, when we test ξ for inclusion in the confidence interval, we use the dual test $(\hat{\theta}_n - \xi)/\sqrt{\xi(1-\xi)/n} \leq z_{1-\alpha}$, and solving for ξ yields the Wilson lower confidence limit

$$\left(\hat{\theta}_n + \frac{z_{1-\alpha}^2}{2n} - z_{1-\alpha}\sqrt{\hat{\sigma}_n^2 + \frac{z_{1-\alpha}^2}{4n^2}}\right) \Big/ \left(1 + \frac{z_{1-\alpha}^2}{n}\right),$$

see, for instance, Example 11.2.7 in Lehmann & Romano (2005). Even though the Wilson interval tends to give more reasonable results than the Wald interval, it still relies on normal approximations. Also, it may yield too conservative estimates when $\hat{\theta}_n$ is close to zero or one.

Clopper-Pearson exact interval The Clopper-Pearson, or, for short, CP interval is based on the binomial distribution of the true success probability θ instead of a normal approximation. In particular, let $S_n = \sum_{i=1}^n X_i$ denote the random variable that counts the number of successes among the n trials $X_1, X_2, \ldots, X_n$. Then, S_n is binomially distributed, and its cumulative distribution function is

given by

$$\mathbb{P}_\theta(S_n \leq s) = \sum_{i=0}^{s} \binom{n}{i} \theta^i (1-\theta)^{n-i}.$$

Let $s_n = n\,\hat{\theta}_n$ denote the number of successes observed in the sample. We utilize the dual test and include ξ in the confidence interval if

$$\mathbb{P}_\xi(S_n \geq s_n) = \sum_{i=s_n}^{n} \binom{n}{i} \xi^i (1-\xi)^{n-i} \geq \alpha. \tag{3.1}$$

Since $\mathbb{P}_\xi(S_n \geq s_n)$ is increasing in ξ, the CP lower confidence limit is given as the smallest value of ξ such that Equation (3.1) holds.

Because the binomial distribution is discrete and the CP approach ensures that the probability in Equation (3.1) is at least α, corresponding to a coverage probability of at least $(1-\alpha)\cdot 100\%$, it tends to yield conservative confidence limits in comparison to the Wald and the Wilson interval; see, for example, Brown et al. (2001). Therefore, it is typically considered a better choice in small samples or when $\hat{\theta}_n$ is close to zero or one.

For now, this concludes our presentation of confidence intervals and their application in statistical inference for a single null hypothesis. Moving forward, in the next section, we will examine how inference is impacted when we test multiple null hypotheses simultaneously.

3.4 Multiple testing

In hypothesis testing, one of the major concerns is the possibility of falsely rejecting a null hypothesis that is actually true. When this happens, we commit a so-called *type-1 error*. There is also a *type-2 error*, which refers to failing to reject the null hypothesis when it is in fact false, but we will not elaborate further on that.

In the context of a single null hypothesis $\mathcal{H}^0$, we prespecify a significance level α, which limits the probability of the occurrence of a type-1 error. Smaller values of α make it increasingly harder to reject $\mathcal{H}^0$, resulting in larger critical values and wider confidence intervals.

However, it is also common to test multiple hypotheses at once. In medical applications, for example, we might examine several outcome variables that describe the condition of a patient or compare different treatments. Each individual test carries its own risk of a type-1 error, and controlling the occurrence of type-1 errors effectively and efficiently is challenging and needs careful consideration. In this section, we will explore different approaches to this.

For this purpose, let $\mathcal{H}_1^0, \mathcal{H}_2^0, \ldots, \mathcal{H}_m^0$ denote a collection of null hypotheses that we want to test against their respective alternatives $\mathcal{H}_j^A$, $j = 1, 2, \ldots, m$.

3.4.1 Per-comparison error rate

Suppose that each individual null hypothesis $\mathcal{H}_j^0$ is tested independently in such a way that the probability of making a type-1 error is at most α. Then the *per-comparison error rate* is said to be controlled at significance level α. It can be shown that the ratio of the number of type-1 errors to the total number m_0 of true null hypotheses is at most α when the number m of null hypotheses approaches infinity. For example, say we test $m = 100$ null hypotheses and control the per-comparison error rate at the significance level $\alpha = 5\%$. If 80 of these null hypotheses are in fact true, we expect that we erroneously reject four of them.

Note, however, that if all m null hypotheses are in fact true, the probability of erroneously rejecting any null hypothesis is equal to $1 - (1 - \alpha)^m$, which can be considerably larger than α, even for small m. Figure 3.2 shows the inflation of the probability for a type-1 error for different numbers m and significance levels α.

3.4.2 Per-experiment error rate

One attempt to mitigate the type-1 error inflation illustrated in Figure 3.2 is to understand the testing of the m null hypotheses as one experimental unit and to define the *per-experiment error rate* as the proportion of experimental units with at least one type-1 error out of the total number of experimental units investigated. It is said to be controlled at significance level α if the probability of at least one type-1 error is at most α when all m null hypotheses are true.

For example, suppose that we have 20 experimental units with 20 different data sets and we test the same null hypotheses, which are in fact all true. The per-experiment error rate is the probability of making any type-1 error in one experimental unit given that all null hypotheses are true. Given a test that controls the per-experiment error rate at $\alpha = 5\%$, we expect to observe rejections in only one of the 20 experimental units.

All null hypotheses being true, however, is not always the worst-case scenario in terms of the probability of making any type-1 error. For example, suppose we have an experimental unit where all but one of the null hypotheses are true, say $\mathcal{H}_1^0, \mathcal{H}_2^0, \ldots, \mathcal{H}_{m-1}^0$ and the remaining null hypothesis $\mathcal{H}_m^0$ is in fact not true. Suppose further that we have a testing procedure for which the probability of making at least one type-1 error is at most α given that all null hypotheses are true, and the probability of rejecting one of $\mathcal{H}_1^0, \mathcal{H}_2^0, \ldots, \mathcal{H}_{m-1}^0$ is larger than α given that they are true and $\mathcal{H}_m^0$ is not true. This testing procedure controls

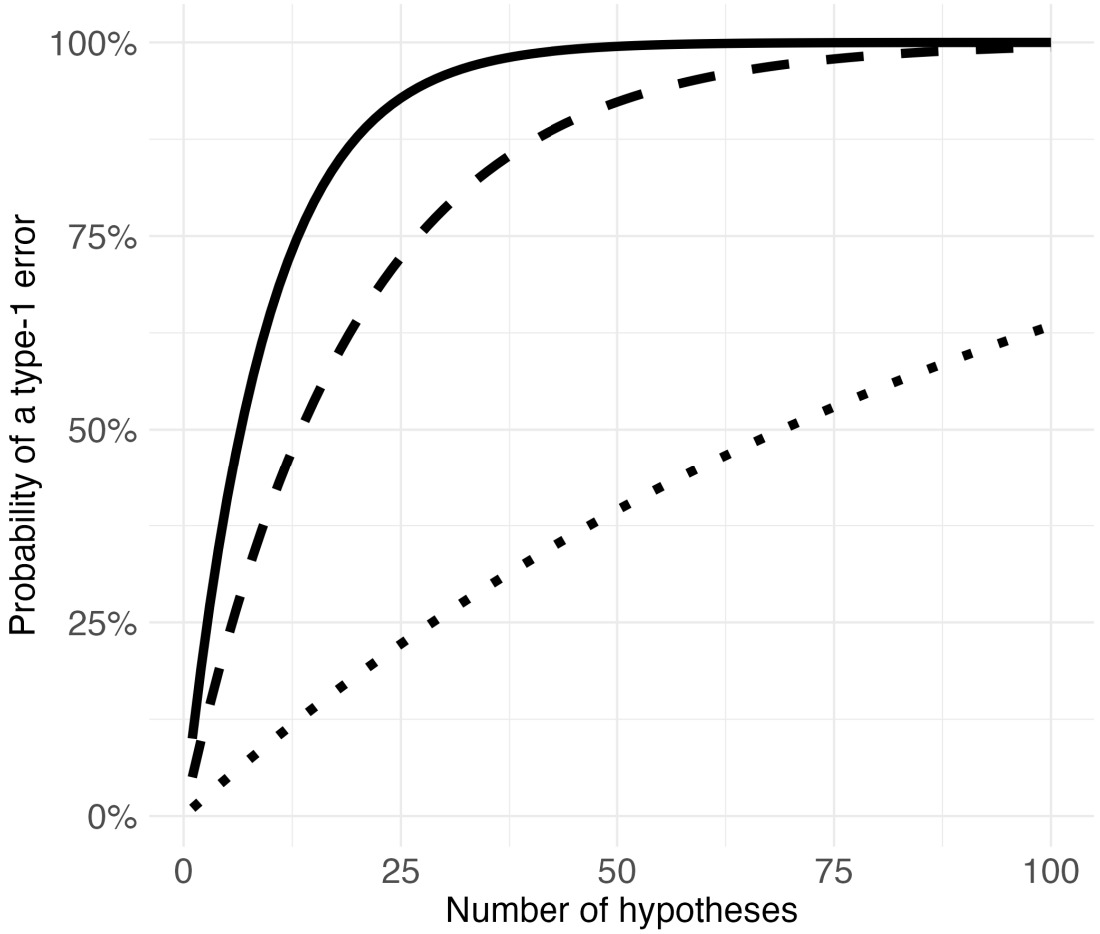

Figure 3.2: Probability of at making least one type-1 error for different numbers m of null hypotheses when each null hypothesis is independently tested at level α. The solid, dashed, and dotted lines correspond to α-levels of ten, five and one percent, respectively

the per-experiment error rate at level α, but this does not guarantee that the probability of making any type-1 error is at most α.

3.4.3 Family-wise error rate

The error rate presented next builds on the notion of experimental units, but it is much more stringent than the per-experiment error rate. The *family-wise error rate*, or, for short, FWER, is said to be controlled at the significance level α if the probability of making any type-1 error among a collection of null hypotheses is at most α, regardless of which and how many of these null hypotheses are in fact true, which is in contrast to the per-experiment error rate.

For example, suppose that we have 20 experimental units with the same null hypotheses on different data sets. If we control the family-wise error rate at the significance level $\alpha = 5\%$, we expect that in only one of the 20 analyses there are any type-1 errors.

More formally, let $\delta_{j,\alpha_j} \in \{0, 1\}$ denote a *test* for the null hypothesis $\mathcal{H}_j^0$ at significance level α_j based on the sample $X_1, X_2, \ldots, X_n$. The test δ_{j,α_j} is equal to one when we reject $\mathcal{H}_j^0$, and it is equal to zero when we cannot reject $\mathcal{H}_j^0$. When $J_0 \subseteq \{1, 2, \ldots, m\}$ denotes the index set of true null hypotheses, then the family-wise error rate $\text{FWER}_\theta(\boldsymbol{\delta})$ of $\boldsymbol{\delta} = (\delta_{j,\alpha_j} : j = 1, 2, \ldots, m)$ under θ is given by

$$\text{FWER}_\theta(\boldsymbol{\delta}) = \mathbb{P}_\theta(\cup_{j \in J_0} \{\delta_{j,\alpha_j} = 1\}).$$

Note that, in general, the significance level α_j might depend on the specific null hypothesis $\mathcal{H}_j^0$. It is thus called a *local* significance level. A testing procedure that controls the family-wise error rate at the *global* significance level α typically adjusts the local levels α_j such that the family-wise error rate is at most α. There are several ways to do this adjustment. We present a brief overview of some popular procedures. They are adequate when there is no particular structure or hierarchy between the null hypotheses such that it is sensible to test all null hypotheses simultaneously, yet individually and independently of the other ones.

Single-step tests Procedures from the class of single-step tests are the simplest in that they are easy to understand and widely applicable with only little assumptions. Two of the most prominent examples are the Bonferroni and the Šidák test. Each of these allocate the same proportion of the global significance level to each null hypothesis such that each $\mathcal{H}_j^0$ is tested at the same local level $\alpha_j = \alpha_{\text{adj}}$, independently of the index j.

For the Bonferroni test, no additional assumptions need to be made and the *adjusted significance level* is given by $\alpha_{\text{adj}} = \alpha/m$. It can easily be shown that

%/m	1	2	5	10	50	100	200	500
Šidák	5	2.53	1.02	0.512	0.103	0.0513	0.256	0.0103
Bonferroni	5	2.5	1	0.5	0.1	0.05	0.25	0.01

Table 3.1: Adjusted significance levels (in percent) for each individual null hypothesis $\mathcal{H}_j^0$ when the Šidák test is used compared to the Bonferroni test

allocating an equal share of the global level to each null hypotheses leads to the control of the family-wise error rate,

$$\begin{aligned} \text{FWER}_\theta(\boldsymbol{\delta}) &= \mathbb{P}_\theta(\cup_{j\in J_0}\{\delta_{j,\alpha_{\text{adj}}}=1\}) \leq \sum_{j\in J_0}\mathbb{P}_\theta\{\delta_{j,\alpha_{\text{adj}}}=1\} \\ &\leq \sum_{j\in J_0}\alpha_{\text{adj}} = \sum_{j\in J_0}\frac{\alpha}{m} = \frac{m_0}{m}\,\alpha \leq \alpha, \end{aligned}$$

due to the Bonferroni inequality and since $m_0 \leq m$, where $m_0 = |J_0|$ is the number of true null hypotheses.

The disadvantage of the Bonferroni test is that the adjusted level $\alpha_{\text{adj}} = \alpha/m$ gets very small when the number m of null hypotheses gets large. It then becomes increasingly difficult to reject any null hypothesis.

In case we can assume that all m individual tests are jointly stochastically independent, the Šidák test allows us to choose α_{adj} slightly larger while still controlling the family-wise error rate. At this point, we should remark, even if we do not elaborate further, that the Šidák test remains valid under certain forms of positive dependence between the individual tests, see Section 4.3.2 in Dickhaus (2014) for details.

In particular, we choose $\alpha_{\text{adj}} = 1-(1-\alpha)^{1/m}$ and test each null hypothesis at this adjusted level. Then,

$$\begin{aligned} \text{FWER}_\theta(\boldsymbol{\delta}) &= \mathbb{P}_\theta(\cup_{j\in J_0}\{\delta_{j,\alpha_{\text{adj}}}=1\}) = 1-\mathbb{P}_\theta(\cap_{j\in J_0}\{\delta_{j,\alpha_{\text{adj}}}=0\}) \\ &= 1-\prod_{j\in J_0}\mathbb{P}_\theta\{\delta_{j,\alpha_{\text{adj}}}=0\} \leq 1-\prod_{j\in J_0}(1-\alpha)^{1/m} \\ &= 1-(1-\alpha)^{m_0/m} \leq 1-(1-\alpha) = \alpha, \end{aligned}$$

due to the joint stochastic independence and because $m_0 \leq m$.

The relative gain of the Šidák over the Bonferroni test increases with the number m of null hypotheses, but admittedly, the gains are small, see Table 3.1. Yet, also with the Šidák test the adjusted level gets very small when m gets large. In addition, even for smaller numbers m, both the Šidák and the Bonferroni test can be overly conservative or anti-conservative, for instance, when the correlation

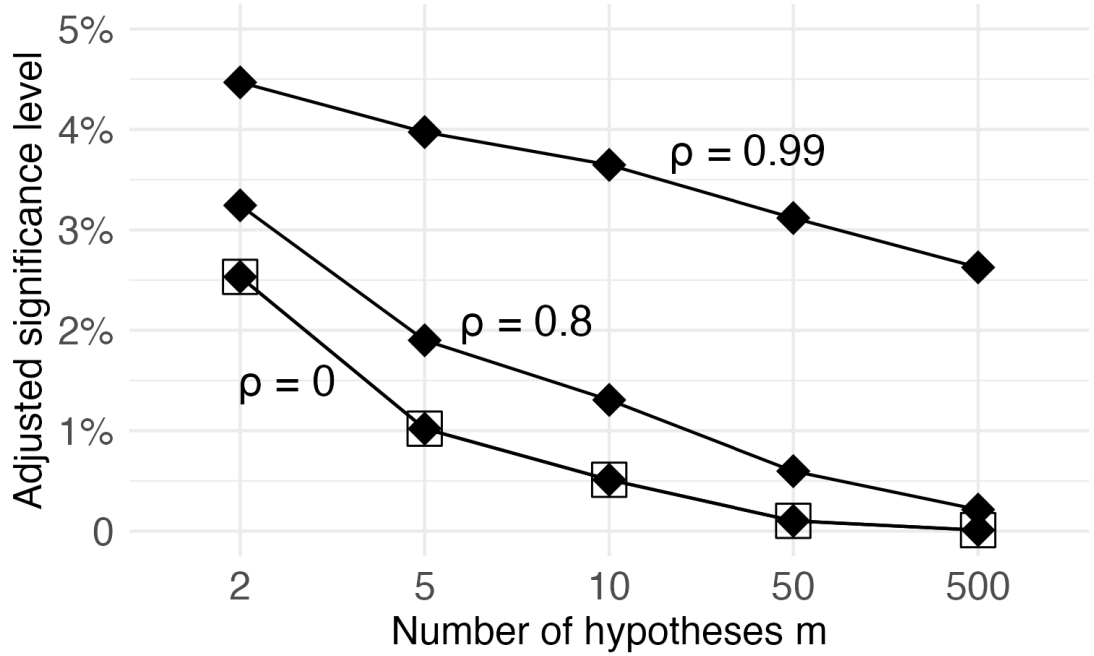

Figure 3.3: Adjusted significance levels from the max-T correction (diamonds) and the Šidák correction (squares) for different levels of correlation ρ

structure between the individual hypothesis tests is complex, see, for example, Section 2.3.1 in Westfall & Young (1993).

Parametric single-step tests If we can make assumptions about the joint distribution of the associated test statistics $T_1, T_2, \ldots, T_m$ and model their covariance structure, we are able to obtain much less conservative tests. For example, if $(T_1, T_2, \ldots, T_m)$ follows a multivariate normal distribution, the so-called *max*-T *test* estimates the critical value c_α as the $(1-\alpha)\cdot 100\%$ quantile of the distribution of the maximum $\max(T_1, T_2, \ldots, T_m)$ under the assumption that $\mathcal{H}_1^0, \mathcal{H}_2^0, \ldots, \mathcal{H}_m^0$ are all simultaneously true.

Figure 3.3 shows the adjusted significance levels for the max-T and the Šidák test for different levels of correlation ρ and different numbers m of simultaneously tested null hypotheses, when the true distribution of $(T_1, T_2, \ldots, T_m)$ is multivariate normal with mean vector $\mathbf{0} = (0, 0, \ldots, 0)$, and covariance matrix $\mathbf{\Sigma} = (\Sigma_{j,j'})_{j,j'}$, given by

$$\Sigma_{j,j'} = \begin{cases} 1, & \text{if } j = j', \\ \rho, & \text{if } j \neq j'; \end{cases}$$

that is, each two distinct test statistics T_j and $T_{j'}$ are equally correlated. The adjusted critical value c_α for the max-T test can then be obtained from the solution to

$\Phi_{\mathbf{0},\mathbf{\Sigma}}(\boldsymbol{c}_\alpha) = 1-\alpha$, where $\Phi_{\mathbf{0},\mathbf{\Sigma}}$ denotes the m-dimensional multivariate normal distribution with mean vector $\mathbf{0}$ and covariance matrix $\mathbf{\Sigma}$, and $\boldsymbol{c}_\alpha = (c_\alpha, c_\alpha, \ldots, c_\alpha)$; see, for example, the discussion around Equation (4.1) in Dickhaus (2014) for details.

There are visible gains when the max-T test is used compared to the Šidák test, in that the adjusted significance level available for each individual hypothesis test is substantially larger, especially when the correlation between the test statistics is large.

The max-T correction is a well-known standard approach in multiple testing, for example, to compute multiplicity-adjusted confidence intervals for linear combinations of parameters, such as mean differences. For an extensive discussion of the max-T correction, see Hothorn et al. (2008).

Resampling-based single step tests In many real-world applications, however, the joint distribution of the test statistics $T_1, T_2, \ldots, T_m$ is unknown. But instead of falling back on the Bonferroni or the Šidák test, Westfall & Young (1993) proposed a universal procedure based on so-called *bootstrap resampling*. This is a statistical method that can be used to estimate the joint distribution of $T_1, T_2, \ldots, T_m$ based only on the sample $X_1, X_2, \ldots, X_n$, and the derived tests are less conservative than the aforementioned single-step tests because they account for the empirical correlation structure of $(T_1, T_2, \ldots, T_m)$. In fact, the multiple testing procedure we propose in this work is based on bootstrap resampling. We will discuss it in much more detail in Chapter 4.

3.5 Reframing model selection and evaluation

In this final section of Chapter 3, we will meet our objective and translate the model selection and evaluation task from Chapter 2 into the framework of statistical inference.

Recall that we are interested in assessing the conditional prediction performance of binary classification models. This, in particular, requires that we hold-out an evaluation set. In the context of statistical inference, this constitutes the sample $X_1, X_2, \ldots, X_n$. Depending on whether we employ the default selection-evaluation pipeline or the proposed one, where we evaluate multiple models simultaneously, we either do not to deal with a multiplicity problem or we do.

Inference in the default pipeline In particular, when we employ the default pipeline, we select a prediction model among a collection of candidate models

based on their validation performances $\hat{\theta}_{V,j}$, $j = 1, 2, \ldots, M$. Suppose we select the model with index s. With this model, we predict the observations in the evaluation set and obtain a performance estimate $\hat{\theta}_s$, such as prediction accuracy. Of course, we want the selected model to perform reasonably well, for example, we might want its performance to exceed a minimum acceptable performance ξ.

In the framework of statistical inference, this corresponds to testing the null hypothesis $\mathcal{H}_s^0 \colon \theta_s \leq \xi$ against the alternative $\mathcal{H}_s^A \colon \theta_s > \xi$. To decide whether we can reject $\mathcal{H}_s^0$, we estimate a lower $(1-\alpha) \cdot 100\%$-confidence limit $\hat{L}(1-\alpha)$, for example, using the Wald method, and check if $\xi < \hat{L}(1-\alpha)$. If this is indeed the case, the evidence from the evaluation is set is strong enough such that we can reject $\mathcal{H}_s^0$ and conclude that $\mathcal{H}_s^A$ is true; that is, with high confidence, the true performance exceeds the minimum acceptable performance. Or in other words: With high confidence, the true performance of prediction model s is at least $\hat{L}(1-\alpha)$.

Inference in the proposed pipeline In contrast, when we employ the proposed pipeline, we preselect multiple models for evaluation. With each of the preselected models, we predict the observations in the evaluation set and estimate their respective prediction performances $\hat{\theta}_{s_1}, \hat{\theta}_{s_2}, \ldots, \hat{\theta}_{s_m}$.

In the context of statistical inference, the simultaneous performance assessment of the m preselected model poses a multiple testing problem with null hypotheses $\mathcal{H}_j^0 \colon \theta_{s_j} \leq \xi_j$ and alternatives $\mathcal{H}_j^A \colon \theta_{s_j} > \xi_j$, for $j = 1, 2, \ldots, m$. Regardless of whether and how we perform the final model selection on the basis of the $\hat{\theta}_{s_j}$'s, we need to correct for multiplicity. To decide whether we can reject each individual hypothesis $\mathcal{H}_j^0$, we need to adjust the significance level, for example using the Bonferroni method, and estimate lower $(1-\alpha_{\text{adj}}) \cdot 100\%$-confidence limits $\hat{L}_j$ and check if $\xi_j < \hat{L}_j$. Let $J_A = \{j = 1, 2, \ldots, m \mid \xi_j < \hat{L}_j\}$ denote the set of indexes for which we can reject the corresponding null hypothesis and conclude that the alternative is true; that is, for $j' \in J_A$, with high confidence, the true performance $\theta_{j'}$ exceeds the respective minimum acceptable performance $\xi_{j'}$. Or in other words: For each $j' \in J_A$, with high confidence, the true performance $\theta_{j'}$ of the corresponding prediction model is at least $\hat{L}_{j'}$.

The translation of the selection-evaluation task into the inference framework allows us to use rigorous statistical methods, efficiently exhausting the permissible type-1 error probability α while drawing valid conclusions about the predictive performance. The main contribution of this work is a sophisticated approach to compute multiplicity-adjusted confidence limits that can be applied in the proposed selection-evaluation pipeline. The idea to interpret this as a multiple testing

problem is based on work on universally valid post-selection confidence limits for regression coefficients by Berk et al. (2013).

We will present our approach in Part II of this work. It extensively uses a popular resampling technique, the bootstrap, which we will explore in great detail in the next Chapter 4.

Chapter 4

Bootstrap

In Chapter 3, we used hypothesis testing to draw conclusions about population characteristics. For example, we might ask whether the mean of some distribution is larger than some reference value, and address this using a test statistic, or equivalently, we could estimate a lower confidence bound.

Usually, we only have a single data sample available to estimate the population characteristic with, using a particular test statistic. But we typically do not know the probability distribution of that test statistic. In order to be able to compute a confidence interval, however, we will need information about this distribution.

If many more replicated samples from the population were available, we could compute the test statistic in each of these samples and use this series of values to estimate the distribution of the test statistic. Typically, though, it is not possible to obtain such additional samples. The bootstrap method offers a practical solution in such cases, allowing for the estimation of the distribution of the test statistic without requiring additional samples from the population.

This chapter is organized as follows. In Section 4.1, we will introduce the basic concept of the bootstrap. Then, in Section 4.2, we will address the question why the bootstrap even works and in which sense. In Section 4.3, we will introduce bootstrap confidence intervals, and in Section 4.4 we will discuss the nonparametric bootstrap tilting confidence interval, which we will base our subsequent considerations on in Part II of this work.

Sections 4.1, 4.2, and 4.3 loosely follow Section 29 in DasGupta (2008). Key references for a comprehensive presentation of the bootstrap include Davison & Hinkley (1997) and Efron & Tibshirani (1994).

4.1 Principle idea

The bootstrap is an approach to estimate a probability distribution based only on the one available sample. The main idea is to regard the sample as the population and generate new samples from the original one. Assuming that the original sample represents the population well, repeated random samples from the original sample can be considered as proxies for repeated samples from the population itself. These proxies are called *bootstrap samples*. In each of the bootstrap samples, we compute the test statistic and use this series of values to obtain an estimate of the distribution.

Nonparametric bootstrap More specifically, let $X_1, X_2, \ldots, X_n$ denote the original sample, where the X_i's are i. i. d. random variables from an unknown distribution represented by its cumulative distribution function F. We want to use the X_i's to make inferences about a population characteristic $\theta = \theta(F)$ using a test statistic T. Hence, we need knowledge about the true distribution function

$$H_n(t) = \mathbb{P}_F[T \leq t]$$

of the test statistic. The idea is to generate bootstrap samples $(X_1^*, X_2^*, \ldots, X_n^*)$ from the X_i's, that is, random samples of size n from the empirical distribution F_n, and to approximate H_n by the *bootstrap distribution function*

$$\hat{H}_n^*(t) = \mathbb{P}_{F_n}[T^* \leq t], \tag{4.1}$$

where T^* denotes the test statistic in the bootstrap sample, and $\mathbb{P}_{F_n}$ in Equation (4.1) indicates probability under all n^n possible bootstrap samples from the original sample, as each X_i can be drawn into the bootstrap sample any number of times. But even for moderately large samples, recalculating T from all n^n bootstrap samples becomes computationally infeasible. We will thus draw only a smaller number B of bootstrap samples, usually a few thousand.

Of course, this introduces a second source of error into the approximation, in addition to the error from pretending that a bootstrap sample is a proxy for a true sample from the true distribution F. However, it turns out that drawing only a much smaller number $B \ll n^n$ of bootstrap samples provides a good balance between accuracy of the estimate and computational cost, with diminishing returns beyond that, such that this second source of error is entirely ignored in any further considerations.

Consequently, we use

$$\hat{H}_n^*(t) = \mathbb{P}_*[T^* \leq t] = \frac{1}{B}\sum_{b=1}^{B} \mathbb{1}\{T_b^* \leq t\}$$

as the bootstrap estimate for the true distribution function H_n, where T_b^* denotes the value of the test statistic in the b-th bootstrap sample.

Note that we replaced $\mathbb{P}_{F_n}$ in Equation (4.1) with $\mathbb{P}_*$ here to indicate probability under bootstrap sampling using B bootstrap samples. Throughout this work bootstrap quantities will carry a subscript or superscript $*$. Such quantities depend on both the sample size n and the sample $X_1, X_2, \ldots, X_n$ itself. Thus, $\hat{H}_n^*$ is a random distribution function that we use to approximate the deterministic but unknown distribution function H_n of the test statistic.

Although the bootstrap is in general not restricted to that case, we will continue to assume that $X_1, X_2, \ldots, X_n$ are i. i. d. random variables. This will give us access to powerful theoretical results on the bootstrap, as we will see later. In addition, we also assume that $X_1, X_2, \ldots, X_n$ are scalars, at least for now.

The bootstrap approach presented in this section is called the *nonparameteric bootstrap*, as it does not make any distributional assumptions on the sample. Rather, the bootstrap samples are generated by repeatedly drawing randomly from the X_i's. The nonparametric bootstrap is one of the two fundamental bootstrap approaches, next to the *parametric bootstrap*, which we will briefly discuss next.

Parametric bootstrap The parametric bootstrap assumes that the sample $X_1, X_2, \ldots, X_n$ comes from a known parametric distribution $F_{\boldsymbol{\eta}}$ with unknown parameter vector $\boldsymbol{\eta}$. The process involves computing an estimate $\hat{\boldsymbol{\eta}}$ from the original sample and generating bootstrap samples from the fitted distribution $F_{\hat{\boldsymbol{\eta}}}$.

We will provide an example in order to illustrate how the parametric bootstrap operates and how it differs from the nonparametric bootstrap. Suppose the i. i. d. sample comes from a normal distribution with unknown population mean $\theta = \theta(F)$ and population variance one, and we want to gain information about the distribution of the test statistic $T = \sqrt{n}(\bar{X}_n - \theta)$, where $\bar{X}_n = n^{-1}\sum_{i=1}^n X_i$ denotes the sample mean. We estimate θ using the sample mean, fit a normal distribution with mean $\bar{X}_n$ and variance one, and draw bootstrap samples from it, in contrast to the nonparametric bootstrap, where we sample from the empirical distribution. In each of the bootstrap samples, we recalculate the test statistic $T^* = \sqrt{n}(\bar{X}_n^* - \bar{X}_n)$, where $\bar{X}_n^* = n^{-1}\sum_{i=1}^n X_i^*$ is the mean in the bootstrap sample. This yields a series of values $T_1^*, T_2^*, \ldots, T_B^*$ that we use to finally estimate

the true distribution H_n of T by

$$\hat{H}_n^*(t) = \frac{1}{B}\sum_{b=1}^{B} \mathbb{1}\{T_b^* \leq t\}.$$

Compared to its parametric counterpart, though, the nonparametric bootstrap is more versatile and, due to its automatic nature, a very popular tool. However, it is not immediately clear why one or the other even works.

4.2 Theoretical justification

We first need to establish what it means for the bootstrap to *work*. Intuitively, when we need to estimate a distribution function, we would want the bootstrap distribution function $\hat{H}_n^*$ to be numerically close to the true distribution function H_n, which can be captured by the idea of *consistency* of $\hat{H}_n^*$ for H_n. For this, we will need some basic concepts from asymptotic statistics, specifically *almost sure convergence* and *convergence in probability*, which we provide in Appendix A.

The bootstrap is called *weakly consistent* under a metric ρ for the test statistic T if $\rho(H_n, \hat{H}_n^*)$ converges to zero in probability as n tends to ∞, and it is *strongly consistent* under ρ for T if $\rho(H_n, \hat{H}_n^*)$ converges to zero almost surely.

Powerful results have been established under the *Kolmogorov distance*,

$$\rho_\infty(H_n, \hat{H}_n^*) = \sup_{t\in\mathbb{R}} |H_n(t) - \hat{H}_n^*(t)|.$$

It captures the maximum difference between the true and the bootstrap distribution function. The following result is Theorem 29.1 in DasGupta (2008), and it shows that if our goal of inference is the expected value μ of F and the test statistic is given by $T = \sqrt{n}(\bar{X}_n - \mu)$, the bootstrap accurately estimates the entire distribution of T; this property is called the *strong consistency of the bootstrap*.

Theorem 4.2.1 (Strong consistency of the bootstrap)**.** *Let $X_1, X_2, \ldots, X_n$ be i. i. d. random variables from some unknown distribution represented by its cumulative distribution function F that has a finite second moment $\mathbb{E}_F(X_i^2) < \infty$. Let $\mu = \mathbb{E}_F(X_i)$ and $\bar{X}_n = n^{-1}\sum_{i=1}^n X_i$ denote the expected value of F and the sample mean of the X_i's, respectively.*

If $T = \sqrt{n}(\bar{X}_n - \mu)$, then the Kolmogorov distance $\rho_\infty(H_n, \hat{H}_n^*)$ converges to zero almost surely as n tends to ∞; that is, the absolute difference between the true distribution function $H_n(t)$ and the bootstrap distribution function $\hat{H}_n^*(t)$ converges almost surely to zero, uniformly over all possible values of t.

We note that the test statistic T in Theorem 4.2.1 has a limiting normal distribution by the Central Limit Theorem, which we provide in a more general multivariate version in Proposition A.3 in Appendix A. When we assume that our goal of inference is a smooth function f of μ instead, the test statistic $T = \sqrt{n}[f(\bar{X}_n) - f(\mu)]$ has a limiting normal distribution, as well (by the Delta Theorem, see Proposition A.4 in Appendix A), and the bootstrap remains strongly consistent, as stated in the next result, which is based on Theorem 29.4 in DasGupta (2008).

Theorem 4.2.2 (Delta theorem for the bootstrap)**.** *Let $X_1, X_2, \ldots, X_n$ be i. i. d. random variables from some unknown distribution represented by its cumulative distribution function F that has a finite second moment $\mathbb{E}_F(X_i^2) < \infty$. Let $\mu = \mathbb{E}_F(X_i)$ and $\bar{X}_n = n^{-1}\sum_{i=1}^n X_i$ denote the expected value of F and the sample mean of the X_i's, respectively. Additionally, let $\theta = f(\mu)$ be a smooth function f of μ, and let $\hat{\theta}_n = f(\bar{X}_n)$ denote the observed value of θ in the sample.*

Then, for $T = \sqrt{n}(\hat{\theta}_n - \theta)$, the Kolmogorov distance $\rho_\infty(H_n, \hat{H}_n^*)$ converges to zero almost surely as n tends to ∞; that is, the absolute difference between the true distribution function $H_n(t)$ and the bootstrap distribution function $\hat{H}_n^*(t)$ converges almost surely to zero, uniformly over all possible values of t.

This Theorem 4.2.2 assures that the probability that the bootstrap distribution accurately reflects the true distribution approaches one, providing reliable estimates for statistical inference, such as confidence intervals.

A natural question regarding the accuracy that comes to mind is whether there is any gain in using the bootstrap when a normal approximation is available. In fact, there are situations in which the bootstrap can be more accurate than the normal approximation. For instance, suppose the true distribution is severely skewed. Because the normal distribution is symmetric around its mean, it is unable to capture this skewness. When our goal is to derive a confidence interval from the approximated distribution, this can considerably reduce its accuracy, that is, the coverage probability of the derived confidence intervals can fall far below the nominal level. Bootstrap confidence intervals, in contrast, are able to correct for skewness.

4.3 Standard confidence intervals

There are different methods to compute bootstrap confidence intervals. Which to use depends very much on the specific application and it is hard to give general recommendations. In this section, we will present an exemplary intuitive method

and describe its characteristics. A detailed review of common bootstrap interval methods can, for example, be found in Carpenter & Bithell (2000).

Recall that a confidence set appropriate for testing the null hypothesis $\mathcal{H}_\xi^0\colon \theta = \xi$ is based on a test statistic $T(\xi)$ that depends on ξ, and it consists of those test values ξ for which we are unable to reject $\mathcal{H}_\xi^0$. In order to decide whether to reject $\mathcal{H}_\xi^0$ or not, we need to know the distribution of $T(\xi)$ for each test value ξ.

4.3.1 Bootstrap pivotal interval

To simplify the process, the test statistic is often constructed in such a way that it is *pivotal*, that is, under the assumption that the null hypothesis $\mathcal{H}_\xi^0$ is true, has a distribution which is independent of ξ.

As an example, consider i. i. d. random variables $X_1, X_2, \ldots, X_n$ from the normal distribution with unknown population mean θ and variance 1. Then, assuming that $\mathcal{H}_\xi^0\colon \theta = \xi$ is true, the asymptotic distribution of the test statistic $\sqrt{n}(\bar{X}_n - \xi)$ is standard-normal and, in particular, independent of ξ and thus pivotal.

Let $T(\xi)$ denote some test statistic which is monotonically decreasing in ξ and pivotal, that is, its unknown true distribution H_n does not depend on the test value ξ. When T^{-1} denotes the inverse of T in ξ, inversion of the test statistic yields

$$1 - \alpha = \mathbb{P}_F\{H_n[T(\xi)] \leq 1 - \alpha\} = \mathbb{P}_F\{\xi \leq T^{-1}[H_n^{-1}(1 - \alpha)]\},$$

and hence a lower $(1 - \alpha) \cdot 100\%$-confidence bound $\ell(1 - \alpha)$ for $\theta(F)$ is given by

$$L(1 - \alpha) = T^{-1}[H_n^{-1}(1 - \alpha)].$$

Inserting the estimated quantile function $\hat{H}_n^{*-1}(q) = \inf\{t \mid \hat{H}_n^*(t) \geq q\}$ of the bootstrap distribution yields the estimate for the lower limit, the *bootstrap pivotal* lower limit

$$\hat{L}^*(1 - \alpha) = T^{-1}\,[\hat{H}_n^{*-1}(1 - \alpha)].$$

There are several other bootstrap interval approaches that are based on the idea of pivotality, including the bootstrap percentile interval and generalizations of it, as well as the bias-corrected bootstrap percentile interval and its accelerated versions, see, for example, Carpenter & Bithell (2000) for details.

4.3.2 Pivotality condition

Let us briefly review the pivotality condition introduced in Section 4.3.1. We can view any distribution of the data to comprise two parts, the parameter of interest θ and a possibly infinite-dimensional vector of nuisance parameters $\boldsymbol{\eta}$. The true distribution F of the X_i's can consequently be represented as $F = (\theta, \boldsymbol{\eta})$.

When we assume that the test statistic is pivotal, we effectively assume that F belongs to a family $\{F_\xi\}_\xi$ of distributions in which each member F_ξ is linked to a test value ξ. However, because the distribution H_n of the test statistic $T(\xi)$ is independent of ξ, for the methods described in Section 4.3.1, it is sufficient to estimate only one member of $\{F_\xi\}_\xi$, the empirical distribution F_n, in order to obtain H_n.

In the next section, we will introduce the nonparametric tilting interval. This instance of bootstrap confidence intervals aims to estimate the entire family $\{F_\xi\}_\xi$ without assuming that the test statistic is pivotal, and will be foundational for our further considerations in Part II of this work. Key references for the nonparametric tilting interval are Csiszár (1975) and Efron (1981).

4.4 Nonparametric tilting confidence interval

When we generate a bootstrap sample from the empirical distribution F_n, the probability that a particular X_i is drawn into the bootstrap sample is n^{-1}. Nonparametric tilting aims to estimate the distribution of the test statistic $T(\xi)$ by adjusting the probabilities to draw bootstrap samples with from the original sample $X_1, X_2, \ldots, X_n$. The confidence set for θ is then formed consisting of those test values ξ that we cannot reject in a test of the null hypothesis $\mathcal{H}_\xi^0\colon \theta = \xi$ using $T(\xi)$.

More precisely, when we restrict our considerations to distributions with support only on $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$, we may represent a distribution in terms of the probabilities $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)$ it puts on $\boldsymbol{X}$. Let $F_{\boldsymbol{w}}$ denote this *reweighted distribution* and let $\hat{\theta}(\boldsymbol{w})$ denote the population parameter of interest estimated from the reweighted distribution $\boldsymbol{w}$.

For example, the empirical distribution F_n can be represented by the vector $\boldsymbol{w}^0 = (n^{-1}, n^{-1}, \ldots, n^{-1})$, and the observed value of θ in the sample is $\hat{\theta}_n = \theta(\boldsymbol{w}^0)$.

Typically, though, we will be interested in distributions $\boldsymbol{w}$ other than $\boldsymbol{w}^0$. Specifically, in order to test a particular test value ξ for inclusion in the confidence set, we need to find the distribution corresponding to the weight vector $\boldsymbol{w}^\xi$ for which it holds true that $\theta(\boldsymbol{w}^\xi) = \xi$. Yet, there may be more than one such distribution.

Moving forward, we reduce the problem to a one-parametric family $\{F_\tau\}_\tau$ of distributions, indexed by the scalar parameter τ. Effectively, this means that the weight vector gets parametrized such that $\boldsymbol{w}(\tau) = [w_1(\tau), w_2(\tau), \ldots, w_n(\tau)]$, and the goal is to find the value of τ such that $\hat{\theta}[\boldsymbol{w}(\tau)] = \xi$. This, however, does not suggest the specific form of the weights. Because we want to sample from $F_{\boldsymbol{w}}$, from a theoretical point of view, it seems sensible that we want $F_{\boldsymbol{w}}$ to be close to the empirical distribution F_n.

In the next section, we will address the question of what we mean by *one distribution is close to another* and derive a closed-form expression for the parametrized weights.

4.4.1 Statistical closeness and exponential tilting weights

The notion of statistical closeness is fundamental when we want to quantify how similar or dissimilar two distributions are. One prominent example of a statistical distance is the *Kullback-Leibler divergence*. It quantifies the amount of information that is lost when we use one distribution to approximate another. Another notable name for the Kullback-Leibler divergence is *relative entropy*. A broader class of divergence measures that includes the Kullback-Leibler divergence as a special case was introduced by Csiszár (1975) through the so-called *I-divergence*.

In particular, when we use a distribution Ψ' to approximate a reference distribution Ψ, the Kullback-Leibler divergence is formally given as

$$d_{\mathrm{KL}}(\Psi, \Psi') = \int_{-\infty}^{\infty} \psi(x) \log \left[\frac{\psi(x)}{\psi'(x)} \right] dx, \tag{4.2}$$

where ψ and ψ' are the probability density functions of Ψ and Ψ', respectively.

From Equation (4.2) we can derive a few important properties of the Kullback-Leibler divergence. Firstly, we notice that d_{KL} is not symmetrical in its arguments, that is, $d_{\mathrm{KL}}(\Psi, \Psi') \neq d_{\mathrm{KL}}(\Psi', \Psi)$. Secondly, by Jensen's inequality,

$$\begin{aligned} d_{\mathrm{KL}}(\Psi, \Psi') &= \int_{-\infty}^{\infty} -\psi(x) \log \left[\frac{\psi'(x)}{\psi(x)} \right] dx = \mathbb{E}_\Psi \left\{ -\log \left[\frac{\psi'(x)}{\psi(x)} \right] \right\} \\ &\geq -\log \left\{ \mathbb{E}_\Psi \left[\frac{\psi'(x)}{\psi(x)} \right] \right\} = -\log \left[\int_{-\infty}^{\infty} \psi(x) \frac{\psi'(x)}{\psi(x)} dx \right] \\ &= 0, \end{aligned}$$

d_{KL} is nonnegative. And thirdly, d_{KL} is equal to zero if and only if the distributions Ψ and Ψ' are identical.

Recall that our goal is to find a distribution $F_{\boldsymbol{w}}$ that is *close* to F_n, that is, a closed-form expression for the weights $\boldsymbol{w}$. Both distributions $F_{\boldsymbol{w}}$ and F_n are dis-

crete and can be identified by $\boldsymbol{w}$ and $\boldsymbol{w}^0$, respectively. As a result, Equation (4.2) can be rewritten as

$$d_{\mathrm{KL}}(\boldsymbol{w}, \boldsymbol{w}^0) = \sum_{i=1}^{n} w_i \log\left(\frac{w_i}{n^{-1}}\right) = \sum_{i=1}^{n} w_i \log(n w_i). \tag{4.3}$$

Note that in Equation (4.3), we use the *backward* Kullback-Leibler divergence, that is, we measure the information loss when using the empirical distribution F_n to approximate the reweighted distribution $F_{\boldsymbol{w}}$. This will yield a closed-form expression for the weights $\boldsymbol{w}$ that can easily be computed, as we will see in the following. In contrast, the *forward* Kullback-Leibler divergence $d_{\mathrm{KL}}(\boldsymbol{w}^0, \boldsymbol{w})$ does not yield a closed-form solution; see, for example, Section 7 in Dickhaus (2018).

As $F_{\boldsymbol{w}}$ should be close to F_n, we minimize $d_{\mathrm{KL}}(\boldsymbol{w}, \boldsymbol{w}^0)$ in the weight vector $\boldsymbol{w}$ subject to the constraints $\theta(\boldsymbol{w}) = \xi$ and $\sum_{i=1}^{n} w_i = 1$, where $w_i \in [0, 1]$. For that, we use the Lagrange multiplier method. Here, the Lagrangian function is

$$\mathcal{L}(\boldsymbol{w}, \tau, \nu) = \sum_{i=1}^{n} w_i \log(n w_i) - \tau[\theta(\boldsymbol{w}) - \xi] - \nu(1 - \sum_{i=1}^{n} w_i) \ .$$

We take the partial derivative of $\mathcal{L}(\boldsymbol{w}, \tau, \nu)$ with respect to w_j and obtain

$$\begin{aligned} \mathbb{D}_{w_j}[\mathcal{L}(\boldsymbol{w}, \tau, \nu)] &= \mathbb{D}_{w_j}[w_j \log(n w_j)] - \tau \mathbb{D}_{w_j}[\theta(\boldsymbol{w})] - \nu \mathbb{D}_{w_j}(w_j) \\ &= \log(n) + \log(w_j) + 1 - \tau U_j(\boldsymbol{w}) - \nu, \end{aligned}$$

where we set $U_j(\boldsymbol{w}) = \mathbb{D}_{w_j}[\theta(\boldsymbol{w})]$. We will comment on $U_j(\boldsymbol{w})$ later in Section 4.4.2.

Solving

$$\mathbb{D}_{w_j}[\mathcal{L}(\boldsymbol{w}, \tau, \nu)] = 0$$

for $\log(w_j)$ yields $\log(w_j) = \tau U_j(\boldsymbol{w}) - \log(n) - 1 - \nu$, and taking the exponent of both sides yields

$$w_j = \mathrm{e}^{\tau U_j(\boldsymbol{w})} \, n^{-1} \mathrm{e}^{-1-\nu}. \tag{4.4}$$

Plugging this into the second constraint $\sum_{i=1}^{n} w_i = 1$ yields

$$\sum_{i=1}^{n} \mathrm{e}^{\tau U_j(\boldsymbol{w})} \, n^{-1} \mathrm{e}^{-1-\nu} = 1,$$

which is equivalent to

$$\sum_{i=1}^{n} \mathrm{e}^{\tau U_j(\boldsymbol{w})} = n \, \mathrm{e}^{1+\nu}.$$

With this, w_j in Equation (4.4) simplifies to

$$w_j = \frac{e^{\tau U_j(\boldsymbol{w})}}{\sum_{i=1}^{n} \mathrm{e}^{\tau U_j(\boldsymbol{w})}}.$$

Evidently, this is a function of the parameter τ, and we will write

$$w_j(\tau) = \frac{\mathrm{e}^{\tau U_j(\boldsymbol{w})}}{\sum_{i=1}^{n} \mathrm{e}^{\tau U_i(\boldsymbol{w})}}. \tag{4.5}$$

Thus, in the backward Kullback-Leibler sense, the reweighted distribution using the *exponential tilting weights* from Equation (4.5) is the closest to the observed sample under the constraint that $\theta[\boldsymbol{w}(\tau)] = \xi$ in a one-parametric family. Note that this family includes the empirical distribution, as for $\tau = 0$, the exponential tilting weight reduces to $1/n$.

4.4.2 Empirical influence function and means

The partial derivative $U_j(\boldsymbol{w}) = \mathbb{D}_{w_j}[\theta(\boldsymbol{w})]$ in Equation (4.5) quantifies the responsiveness of the test statistic to small changes in the weights, that is, how much $\theta(\boldsymbol{w})$ would change in response to a small perturbation in w_j. This partial derivative is also known as the *empirical influence function* and is given by

$$\mathbb{D}_{w_j}[\theta(\boldsymbol{w})] = \lim_{\epsilon \to 0} \frac{\theta[\boldsymbol{w} + \epsilon \boldsymbol{e}_j] - \theta(\boldsymbol{w})}{\epsilon} \tag{4.6}$$

where $\boldsymbol{e}_j$ is the n-dimensional unit vector with the one in the j-th component.

In case the population parameter of interest is a mean, we can obtain a closed-form expression for the influence function, which we will derive next. This will be important when we address the performance measures presented in Section 2.2.2.

We continue by computing the weights $\boldsymbol{w}' = (w'_1, w'_2, \ldots, w'_n) = \boldsymbol{w} + \epsilon \boldsymbol{e}_j$, which are given by

$$w'_i = \begin{cases} w_i, & \text{if } i \neq j \\ w_i + \epsilon, & \text{if } i = j. \end{cases}$$

Using these weights, we compute the weighted mean

$$\theta[\boldsymbol{w} + \epsilon \boldsymbol{e}_j] = w'_j x_j + \sum_{i \neq j} w'_i x_i = (w_j + \epsilon) x_j + \sum_{i \neq j} w_i x_i.$$

It follows that

$$\frac{\theta[\boldsymbol{w} + \epsilon \boldsymbol{e}_j] - \theta(\boldsymbol{w})}{\epsilon} = \frac{\epsilon x_j + w_j x_j + \sum_{i \neq j} w_i x_i - \sum_{i=1}^{n} w_i x_i}{\epsilon} = x_j.$$

Thus, in case of a mean, the influence function reduces to

$$U_j(\boldsymbol{w}) = x_j.$$

Substituting this expression for $U_j(\boldsymbol{w})$ into the weights in Equation (4.5) yields

$$w_j(\tau) = \frac{\mathrm{e}^{\tau x_j}}{\sum_{i=1}^{n} \mathrm{e}^{\tau x_i}}. \tag{4.7}$$

Hence, when θ can be written as a mean, the reweighted distribution $F_{\boldsymbol{w}(\tau)}$ is an exponential tilt of the empirical distribution, which is why the weights are called *exponential tilting weights*.

Since the reweighted distribution $\boldsymbol{w}(\tau)$ is fully determined by the tilting parameter τ, we will simply write F_τ instead of $F_{\boldsymbol{w}(\tau)}$, with the understanding that F_τ represents the reweighted distribution using the exponential tilting weights.

We might initially think that the reduction of the nonparametric problem to a one-parameter family of distributions would make the estimation somehow easier. After all, dealing with a single parameter that adjusts the reweighted distribution might suggest a simplification.

A way to measure this is the Cramér-Rao bound, see, for example, Section 2.6 in Spokoiny & Dickhaus (2015), which is a lower limit on the variance of an unbiased estimator, and provides an idea about its precision. In fact, in case θ is a mean, the Cramér-Rao bound for the unbiased estimation of $\theta(\boldsymbol{w})$ in $\{F_\tau\}_\tau$, evaluated at $\boldsymbol{w} = \boldsymbol{w}^0$ is $n^{-2} \sum_{i=1}^{n} [x_i - \theta(\boldsymbol{w})]^2$, which is the bootstrap estimate of variance for $\hat{\theta}(\boldsymbol{w})$, see Efron (1981) for details. Thus, the reduction does not decrease the estimated variance, and can be understood to be *least-favorable*.

It is, of course, possible to think of different one-parameter families to substitute for the exponential tilting family, and thus different choices for the weights w_i. One notable instance are the so-called *maximum likelihood tilting weights*, which can be obtained by maximizing the likelihood $\Pi_{i=1}^{n} w_i$ under the constraint that $\theta(\boldsymbol{w}) = \xi$, as well as that the weights are non-negative and sum to one. For more options to choose the tilting weights, see DiCiccio & Romano (1990) or Hesterberg (2014).

Recall that, previously, when we assumed the test statistic to be pivotal and monotonically increasing in the test value ξ, we did not need to test each ξ for inclusion in the confidence set individually. Rather, by inversion of the test statis-

tic, we obtained a lower confidence limit. In nonparametric tilting, the confidence set is an interval, as well, as we will argue next.

As mentioned earlier, we can measure the level of evidence against the null hypothesis $\mathcal{H}_\xi^0$: $\theta = \xi$ in terms of the p-value, which is the probability of obtaining at least the observed value of the test statistic in the original sample under the assumption that $\mathcal{H}^0$ is true, that is, we exclude ξ from the confidence set for θ if

$$\mathbb{P}_\tau[T^* \geq T(\tau)] \leq \alpha,$$

where the tilting parameter τ is such that $\theta[w(\tau)] = \xi$. Fortunately, this probability increases monotonically in τ, which follows from the slightly more general Proposition 4.4.1 given below. For that, we will require the concept and properties of the one-parameter canonical exponential family, which we will shortly discuss next. For more details on exponential families, see Lehmann & Romano (2005).

The *one-parameter canonical exponential family* is a class of probability distributions that can be expressed in a general but specific form. Representing a distribution in its canonical structure simplifies many mathematical operations.

In its general form, the probability density function of a distribution from the one-parameter exponential family is proportional to $z \mapsto \mathrm{e}^{\gamma a(z) - b(\gamma)} h(z)$, where γ is called the canonical parameter the distribution is parametrized by; $a(z)$ is the sufficient statistic that captures all necessary information for estimating γ; $b(\gamma)$ is called the *cumulant* function and ensures that the distribution normalizes to one; and $h(z)$ is a measurable function that can vary with z, but not with γ. The important property for our use is that b is at least three-times continuously differentiable and the first derivate of $b(\gamma)$ equals the *first cumulant*, that is, the mean.

We are now set to record and prove the monotonicity property in the upcoming Proposition 4.4.1.

Proposition 4.4.1. *Let Z denote a random variable that follows a distribution represented by its cumulative distribution function Ψ_γ. Let Ψ_γ belong to a one-parameter canonical exponential family with parameter $\gamma \in \mathbb{R}$ and sufficient statistic $a(z) = z$. Then, the probability $\mathbb{P}_{\Psi_\gamma}(Z \geq t)$ is increasing in γ for all $t \in \mathbb{R}$.*

Proof. We begin this proof with an observation that we will come back to later. Let Z' be some arbitrary random variable. If $t' \geq 0$, then $\mathbb{E}(Z' \mathbb{1}\{Z' \geq t'\}) \geq 0$.

If, alternatively, $t' < 0$, then

$$\begin{aligned} Z' &= |Z'| \mathbb{1}\{Z' \geq 0\} - |Z'| \mathbb{1}\{Z' < 0\} \\ &\leq |Z'| \mathbb{1}\{Z' \geq 0\} - |Z'| \mathbb{1}\{t' \leq Z' < 0\} = Z' \mathbb{1}\{Z' \geq t'\}. \end{aligned}$$

Therefore, if $\mathbb{E}(Z') \geq 0$, then $\mathbb{E}(Z' \mathbb{1}\{Z' \geq t'\}) \geq 0$ for all $t' \in \mathbb{R}$.

Now, let Ψ_γ belong to the canonical one-parameter exponential family, that is, its probability density function is proportional to $\mathrm{e}^{\gamma z - b(\gamma)} h(z)$, where b is at least three-times continuously differentiable and $\frac{d}{d\gamma} b(\gamma) = \mathbb{E}_{\Psi_\gamma}(Z)$.

Consider the probability

$$\mathbb{P}_{\Psi_\gamma}(Z \geq t) = \int_t^\infty \mathrm{e}^{\gamma z - b(\gamma)} h(z)\, dz$$

and the derivative

$$\begin{aligned}\frac{d}{d\gamma}\mathbb{P}_\gamma(Z \geq t) &= \int_t^\infty [z - \frac{d}{d\gamma} b(\gamma)]\, \mathrm{e}^{\gamma z - b(\gamma)} h(z)\, dz \\ &= \mathbb{E}_{\Psi_\gamma}[\{Z - \mathbb{E}_{\Psi_\gamma}(Z)\}\, \mathbb{1}\{Z - \mathbb{E}_{\Psi_\gamma}(Z) \geq t - \mathbb{E}_{\Psi_\gamma}(Z)\}].\end{aligned}$$

Recall that we observed at the beginning of this proof that $\mathbb{E}(Z' \mathbb{1}\{Z' \geq t\}) \geq 0$ for all $t \in \mathbb{R}$ if $\mathbb{E}(Z') \geq 0$. Let $Z' = Z - \mathbb{E}_{\Psi_\gamma}(Z)$. Then, since $\mathbb{E}_{\Psi_\gamma}[Z'] = 0$, it follows that

$$\begin{aligned}0 \leq \mathbb{E}(Z' \mathbb{1}\{Z' \geq t'\}) &= \mathbb{E}_{\Psi_\gamma}[\{Z - \mathbb{E}_{\Psi_\gamma}(Z)\}\, \mathbb{1}\{Z - \mathbb{E}_{\Psi_\gamma}(Z) \geq t - \mathbb{E}_{\Psi_\gamma}(Z)\}] \\ &= \frac{d}{d\gamma}\mathbb{P}_\gamma(Z \geq t),\end{aligned}$$

and hence, $\mathbb{P}_{\Psi_\gamma}(Z \geq t)$ is increasing in γ. □

From Proposition 4.4.1, it follows that there exists a maximum value for the tilting parameter τ such that for all $\tau' > \tau$, we include the corresponding value $\theta[\boldsymbol{w}(\tau')]$ in the confidence set for θ. Thus, $\theta[\boldsymbol{w}(\tau)]$ is a lower confidence limit.

This considerably facilitates the calibration of τ for the estimation of the lower confidence limit: Find the largest value of $\tau < 0$ such that

$$\mathbb{P}_\tau[T^* \geq T(\tau)] \leq \alpha; \tag{4.8}$$

that is, $L = \theta[\boldsymbol{w}(\tau)]$ is the largest value of ξ such that, if the sample came from a distribution with parameter L, the probability of observing $T(\tau)$ or an even larger value would be at most α.

When we use a traditional so-called *Monte Carlo simulation* for the estimation of the lower limit L, conceptually, for any candidate value for τ, we would need to compute the exponential tilting weights; draw bootstrap samples from the reweighted distribution; compute the bootstrap test statistics for each bootstrap sample; compute the proportion of bootstrap test statistics that exceeds the observed value of the test statistics under the null hypothesis; and repeat these steps

with different values for τ until the proportion equals α. Obviously, this would be both computationally very expensive and prone to the randomness of repeated sampling.

In order to circumvent this, what we actually do is to employ an *importance sampling reweighting* approach. In the next section, we will explore importance sampling and its use to effectively address the limitations of traditional sampling in the estimation of nonparametric tilting confidence limits, allowing us to directly draw samples from the empirical distribution.

4.4.3 Importance sampling

When we revisit Equation (4.8), it becomes clear that the calibration of the tilting parameter $\tau < 0$ for a lower confidence limit is difficult for two major reasons. The first is that it involves estimating tail probabilities, that is, the likelihood of events that occur with very low probability. This can only hardly be done accurately using traditional Monte Carlo simulation because of the number of samples needed to observe such events directly with sufficiently high precision. The second is that, in addition, there is an excessive amount of distributions that we need to sample from. Therefore, direct sampling becomes computationally prohibitive or even infeasible.

Importance sampling offers a practical solution to such problems. Our presentation will loosely follow Section 23.6 in Efron & Tibshirani (1994). The main idea is, instead of sampling directly from the *target distribution*, to sample from a different, more easily accessible *design distribution* and then weight the samples appropriately to correct for the difference between the two distributions. This way, it is ensured that the estimates remain accurate for the target distribution.

The key advantage of this approach is that it allows for more efficient sampling in difficult regions of the target distribution, such as the tails, by focusing the sampling effort where there is insufficient data. The correction weights are typically computed as the ratio of the probabilities of the sample under the target to the design distribution such that the contribution of each sample reflects its relative likelihood under the target distribution.

In particular, suppose we need to estimate a probability $q_t = \mathbb{P}_\Psi(Z \geq t)$, where Z is distributed according to the target distribution Ψ with probability density function ψ. In addition, let $W = \psi/\psi'$ denote the relative likelihood under Ψ relative to sampling from a design distribution Ψ' with density function ψ'. Then, we can express the probability under Ψ as a mean of a transformed random variable

under Ψ',

$$q_t = \mathbb{E}_\Psi[\mathbb{1}\{Z \geq t\}] = \int_{-\infty}^{\infty} \left(\mathbb{1}\{Z \geq t\} \frac{\psi(z)}{\psi'(z)} \right) \psi'(z)\, dz = \mathbb{E}_{\Psi'}[\mathbb{1}\{Z \geq t\}\, W(Z)],$$

where ψ' needs to dominate ψ; that is, $\psi(z) = 0$ needs to hold whenever $\psi'(z) = 0$, in order to avoid an indefinite integrand.

Hence, we do not need to sample from Ψ in order to compute q_t, but we can instead use samples from a more convenient distribution Ψ', for example, the empirical distribution, and compensate for the misspecification with a multiplicative factor. This way, we can concentrate our effort on the tail regions.

Now, traditional Monte Carlo simulation for the mean $\mathbb{E}_{\Psi'}[\mathbb{1}\{Z \geq t\}\, W(Z)]$ yields the importance sampling estimate of q_t,

$$\hat{q}_t = \frac{1}{B} \sum_{b=1}^{B} \mathbb{1}\{Z_b \geq t\}\, W(Z_b), \tag{4.9}$$

where the Z_b's are i. i. d. samples from Ψ'.

4.4.4 Estimation

In this section, we will integrate the previous findings and give a concise mathematical description of the nonparametric bootstrap tilting confidence interval and its estimation. We will work under a slight extension of a collection of assumptions that are often summarized as *Hall's smooth function model*, see Hall (1988). We will provide it next.

Let $X_1, X_2, \ldots, X_n$ be an i. i. d. random sample from some unknown distribution $\mathcal{F}$ on some arbitrary sample space Ω. Let h denote a smooth function from Ω to $\mathbb{R}$ that maps each random variable X_i to the random variable Y_i. Let $\mu = \mu(\mathcal{F}) = \mathbb{E}_{\mathcal{F}}(Y_i)$ denote the mean of Y_i. Let θ denote a distributional parameter that we assume to be a smooth function $\theta = f(\mu)$ of μ. Additionally, let F_n denote the empirical distribution of the X_i's, and let $\bar{Y}_n = n^{-1} \sum_{i=1}^{n} Y_i$ denote the sample mean of the Y_i's. Lastly, let $\hat{\theta}_n = \theta(F_n) = f(\bar{Y}_n)$ be the observed value of θ in the sample.

Our goal of inference is θ. Specifically, for a given significance level $\alpha \in [0, 1]$, we want to provide a lower $(1 - \alpha) \cdot 100\%$-confidence limit for θ. We will use the duality between testing the null hypothesis $\mathcal{H}_\xi^0 \colon \theta = \xi$ with a confidence interval and with a test statistic

$$T_\xi = \sqrt{n}\, \frac{\hat{\theta}_n - \xi}{\hat{\sigma}_n},$$

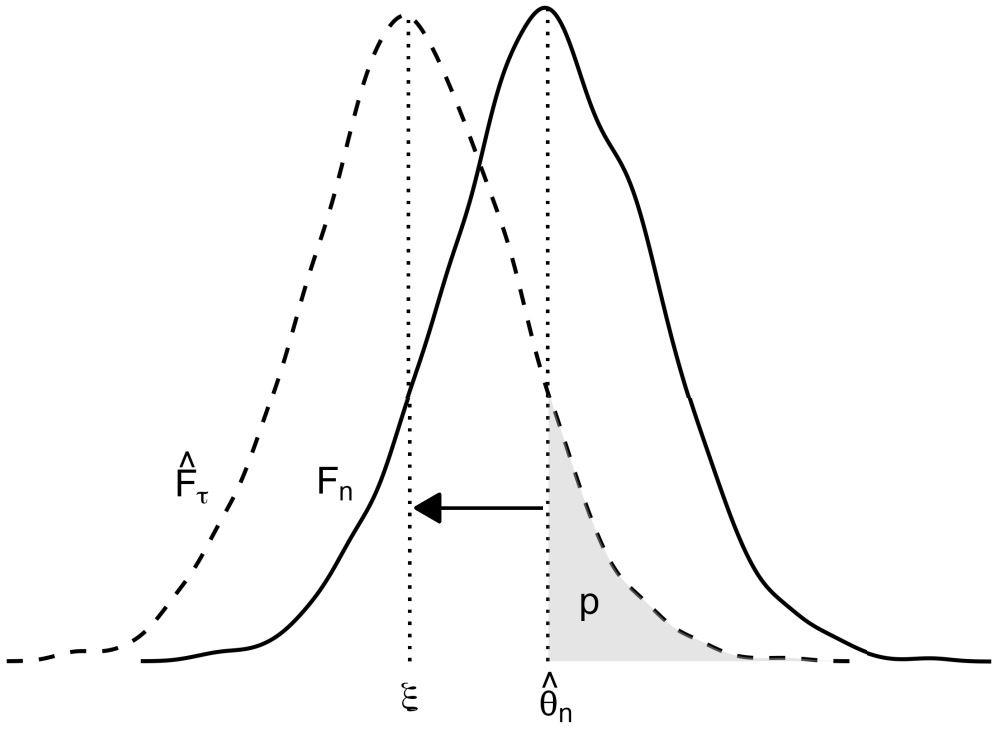

Figure 4.1: Illustration of the estimation of a nonparametric bootstrap tilting lower confidence limit

where $\hat{\theta}_n = \theta(F_n) = f(\bar{Y}_n)$ is the observed value of θ in the sample and

$$\hat{\sigma}_n = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}[f(Y_i) - \hat{\theta}_n]^2}$$

estimates the standard deviation. Then, the bootstrap test statistic is

$$T^* = \sqrt{n}\,\frac{\hat{\theta}_n^* - \hat{\theta}_n}{\hat{\sigma}_n^*},$$

where $\hat{\theta}_n^*$ and $\hat{\sigma}_n^*$ are the mean and standard deviation estimates in the bootstrap sample, respectively. Consequently, by Equation (4.8), the p-value for testing $\mathcal{H}_\xi^0$ is

$$p = \mathbb{P}_\tau\left(\sqrt{n}\,\frac{\hat{\theta}_n^* - \hat{\theta}_n}{\hat{\sigma}_n^*} \geq \sqrt{n}\,\frac{\hat{\theta}_n - \xi}{\hat{\sigma}_n}\right), \tag{4.10}$$

and our objective is to find the largest value for $\tau < 0$ such that $p \leq \alpha$.

We will use importance sampling to estimate the p-value in Equation (4.10), using bootstrap samples from F_n. Specifically, let $M_{i,b}^*$ denote the number of times X_i is drawn into the b-th bootstrap sample $\boldsymbol{X}_b^* = (X_{b1}^*, X_{b2}^*, \ldots, X_{bn}^*)$, $b =$

$1, 2, \ldots, B$, and let $\boldsymbol{w}(\tau)$ denote the vector of exponential tilting weights $w_i(\tau)$, see Equation (4.5). We weight each bootstrap sample with the relative likelihood

$$W_b(\tau) = \frac{\prod_{i=1}^n w_i(\tau)^{M^*_{i,b}}}{\prod_{i=1}^n n^{-1}}$$

of the bootstrap sample under $\boldsymbol{w}(\tau)$-weighted sampling relative to sampling with equal weights n^{-1}. Then, we calibrate the tilting parameter $\tau < 0$ such that the estimated probability of obtaining at least the observed value T of the test statistic in the original sample under $\mathcal{H}^0_\xi$ is equal to α, that is, under the reweighted distribution F_τ. This means, using the importance sampling estimate in Equation (4.9),

$$\hat{p}(\tau) = \frac{1}{B} \sum_{b=1}^{B} \mathbb{1}\left\{ \sqrt{n}\, \frac{\hat{\theta}^*_{n,b} - \hat{\theta}_n}{\hat{\sigma}^*_{n,b}} \geq \sqrt{n}\, \frac{\hat{\theta}_n - \xi}{\hat{\sigma}_n} \right\} W_b(\tau),$$

where

$$\hat{\sigma}^*_{n,b} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} [f(Y^*_{bi}) - \hat{\theta}_n]^2}$$

is the bootstrap estimate of standard deviation in the b-th bootstrap sample, and we calibrate τ such that $\hat{p}(\tau) = \alpha$. Then, $\hat{L} = \hat{\theta}[\boldsymbol{w}(\tau)]$ is the desired lower confidence limit.

To summarize, the basic idea of bootstrap tilting is to adjust the probabilities of samples drawn from the empirical distribution to match the null hypothesis. Utilizing the duality between hypothesis testing and confidence interval estimation, we obtain the lower confidence limit as the parameter of interest estimated from the reweighted distribution. Figure 4.1 illustrates the idea.

It is not immediately clear why the resulting confidence interval should have the nominal coverage probability, which is key for valid statistical inference. The theoretical understanding of the goodness of bootstrap tilting confidence intervals was profoundly advanced by DiCiccio & Romano (1990). We will discuss their findings in the following section.

4.4.5 Asymptotic properties

When we evaluate the theoretical properties of a confidence interval method, one key aspect to consider is the order of correctness. It refers to the rate at which they approach the nominal coverage level as the sample size n tends to infinity. The following two characterizations hold under Hall's smooth function model. Let $\hat{L}$ denote a proposed lower $(1-\alpha)\cdot 100\%$-confidence limit for θ. According to Hall

(1988), $\hat{L}$ is called *first-order correct* if

$$\mathbb{P}(\theta \geq \hat{L}) = 1 - \alpha + \mathcal{O}(n^{-1/2}),$$

and $\hat{L}$ is called *second-order correct* if

$$\mathbb{P}(\theta \geq \hat{L}) = 1 - \alpha + \mathcal{O}(n^{-1}),$$

where $\mathcal{O}(n^{-1/2})$ and $\mathcal{O}(n^{-1})$ represent quantities that have a rate of convergence to zero of $n^{-1/2}$ and n^{-1}, respectively. Therefore, the difference between the coverage probability and the nominal level is proportional to $n^{-1/2}$ and n^{-1}, respectively.

Many confidence interval methods are only first-order correct, such as the Wald and the bootstrap pivotal interval. In practical terms, such methods are often based on relatively simple assumptions, such as the asymptotic normality for the Wald intervals, and are often less reliable regarding coverage probability or less informative, especially in smaller samples. Second-order correct intervals, such as the Wilson interval, in contrast, typically account for more subtle factors, such as skewness of the distribution, and are often more sophisticated. These properties lead to better coverage properties, also in smaller samples. For a detailed review on the asymptotic properties of bootstrap confidence intervals and confidence intervals for binomial proportions, see Hall (1992) and Brown et al. (2001), respectively. Note that the CP interval is an exact method, so we cannot assign an asymptotic order to it.

In fact, DiCiccio & Romano (1990) proved that the nonparametric bootstrap tilting confidence interval is second-order correct. Before we will end this section with some final remarks, we will record this finding for later use.

Proposition 4.4.2. *Under Hall's smooth function model, the lower* $(1-\alpha)\cdot 100\%$*-confidence limit* $\hat{L}(1-\alpha)$ *obtained from nonparametric bootstrap tilting using exponential tilting weights is second-order correct; that is, it holds that*

$$\hat{L}(1-\alpha) = \hat{\theta}_n - n^{-1/2}\hat{\sigma}_n H_n^{-1}(1-\alpha) + \mathcal{O}_{\mathbb{P}}(n^{-3/2}),$$

and

$$\mathbb{P}_\theta[\theta \geq \hat{L}(1-\alpha)] = 1 - \alpha + \mathcal{O}(n^{-1}).$$

4.4.6 Final remarks

We will conclude this chapter with some observations and remarks on the nonparametric bootstrap tilting confidence interval.

Finite samples According to Hall (1988), the asymptotic properties of any confidence interval method is only one part of the information needed to accurately assess it. The other part are simulation studies and applications to real-world data sets. Hesterberg (1999) showed that nonparametric bootstrap tilting confidence intervals reach a comparable level of precision to other bootstrap techniques with far less resamples, and offer a good balance between confidence interval length and accurate coverage in finite samples.

Perfect classifiers A second observation is that, in view of Equation (4.7), we must recognize that nonparametric bootstrap tilting can never work if the data is constant, because then $w_1(\tau) = w_2(\tau) = \cdots = w_n(\tau)$ for any value of the tilting parameter τ, and the empirical distribution cannot be reweighted. This is not only true when the parameter of interest θ is a mean, but in general, see Equations (4.5) and (4.6).

Regarding the application to machine learning prediction models, this issue occurs when the model perfectly predicts the true classes. To deal with this, we could, for example, switch to another, perhaps conservative, interval estimation method, such as the CP interval.

Empirical likelihood Another aspect we would like to mention here are the close links between bootstrap tilting and empirical likelihood methods. Both share foundational similarities in their approach to nonparametric statistical inference, particularly the use of reweighting to approximate distributions. Both estimate a probability distribution that is optimal in the Kullback-Leibler sense.

While bootstrap tilting uses resampling with adjusted weights to reflect the distribution under the null hypothesis, empirical likelihood works directly with the empirical distribution with no resampling required, and is more focused on likelihood theory and nonparametric estimation. While bootstrap tilting minimizes the backward Kullback-Leibler divergence to achieve more accurate statistical inference, empirical likelihood minimizes the forward Kullback-Leibler divergence.

For a more detailed account on the connections between bootstrap tilting and empirical likelihood, or on empirical likelihood, in general, see the considerations around remark 7.12 in Dickhaus (2018) or Schennach (2007) and Owen (2001), respectively.

Part II

Multiplicity-Adjusted Bootstrap Tilting

Chapter 5

Methodology and Theory

In Part I of this work, we explored the development of an effective machine learning prediction model and translated the task of model evaluation into a statistical inference problem. Using the duality to hypothesis testing, we illustrated how to estimate confidence intervals for prediction performance. We also reviewed the bootstrap and discussed nonparametric bootstrap tilting, which is a sophisticated and fairly accurate approach to confidence interval estimation.

Moreover, we presented the default model selection and evaluation pipeline, where we only evaluate a single model, as well as a novel approach by Westphal & Brannath (2020). In this proposed pipeline, multiple models are evaluated and the final model is selected based on its evaluation performance, and we framed this as a multiple testing problem. Additionally, we argued that using the max-T-correction can make simultaneous inference more efficient compared to standard corrections such as the Šidák method, which, in contrast, does not incorporate information about the relationship between the multiple hypotheses.

In this part of the present work, we will integrate these various components to develop an approach that combines the strengths of the different techniques. In particular, we will promote the proposed selection-evaluation pipeline. While Westphal & Brannath (2020) offered a multiple test, we will extend the non-parametric bootstrap tilting confidence interval to correct for multiplicity. The resulting interval will be universally applicable and statistically valid. It will work with any measure of prediction performance from Section 2.2.2; with any combination of prediction models even from different model classes like linear and non-linear candidate models; any model selection strategy, whether formal or informal, or even based on post-hoc considerations; and it will be computationally undemanding as it does not require any additional model training.

The idea to use a resampling-based approach for multiplicity correction is not a new one, though. Westfall & Young (1993) provided a comprehensive overview of various resampling techniques, including the bootstrap, and their use for con-

trolling the FWER. However, the application of the max-T approach in a selection and evaluation problem or, in general, in machine learning applications is not common. It will ultimately enable us to evaluate the conditional performances of multiple candidate models and provide a statistically valid lower confidence limit for the final selected model.

This chapter is organized as follows. In Section 5.1, we will give a complete mathematical description of our proposed multiplicity-adjusted bootstrap tilting, or, for short, MABT confidence limits. In Section 5.2, we will record asymptotic properties of our approach, and in Section 5.3 we will conduct the proofs.

5.1 Mathematical description and estimation

We operate under a multivariate extension of Hall's smooth function model regarding the dimension of the parameter of interest. Let $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ be i. i. d. random vectors from some unknown multivariate distribution $\mathcal{F}$ on some arbitrary sample space Ω. For $m \geq 2$, let h denote a smooth function from Ω to $\mathbb{R}^m$ that maps each $\boldsymbol{X}_i$ to the m-dimensional vector $\boldsymbol{Y}_i = h(\boldsymbol{X}_i)$. Let $\mathcal{F}_j$ and $\mu_j = \mu(\mathcal{F}_j) = \mathbb{E}_{\mathcal{F}_j}(\boldsymbol{Y}_i)$ denote the marginal distributions and marginal means of $\boldsymbol{Y}_i = (Y_{i1}, Y_{i2}, \ldots, Y_{im})$, respectively. Our goal is simultaneous inference for the vector of parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_m)$, using test statistics

$$T_j = \sqrt{n}\,\frac{\hat{\theta}_{j,n} - \theta_j}{\hat{\sigma}_j},$$

where $\theta_j = \theta(\mathcal{F}_j)$ is a smooth function f_j of μ_j, that is, $\theta_j = f_j(\mu_j)$.

In particular, for a prespecified global significance level $\alpha \in [0, 1]$, we want to utilize the duality between estimating confidence intervals and testing the collection of null hypotheses $\{\mathcal{H}_j^0 \colon \theta_j = \xi_j \mid j = 1, 2, \ldots, m\}$ simultaneously, and estimate simultaneous lower confidence limits with asymptotic coverage probability $(1-\alpha)\cdot 100\%$ using the bootstrap tilting approach; that is, for the simultaneous lower limit $\hat{\boldsymbol{L}} = (L_1, L_2, \ldots, L_m)$ it holds that, as $n \to \infty$,

$$\mathbb{P}_{\boldsymbol{\theta}}(\cap_{j=1}^m \{\theta_j > L_j\}) \to 1 - \alpha,$$

where the L_j's are bootstrap tilting lower confidence limits, all calibrated at the same multiplicity-adjusted significance level $\alpha_{\text{adj}} = \alpha_{\text{adj}}(\alpha)$; that is, $L_j = L_j(1 - \alpha_{\text{adj}})$.

Additionally, let $\boldsymbol{X}_1^*, \boldsymbol{X}_2^*, \ldots, \boldsymbol{X}_B^*$ denote bootstrap samples drawn from the original sample $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$, and let $\boldsymbol{T}_b^* = (T_{1b}^*, T_{2b}^*, \ldots, T_{mb}^*)$ be the vector of

test statistics obtained from the b-th bootstrap sample $\boldsymbol{X}_b$, that is,

$$T^*_{jb} = \sqrt{n}\,\frac{\hat{\theta}^*_{jb} - \hat{\theta}_{j,n}}{\hat{\sigma}^*_{jb}}.$$

Note that this way of generating bootstrap samples respects the dependencies between the $\boldsymbol{X}_b$'s.

The presentation will continue as follows: First, we will use the $\boldsymbol{T}^*_b$'s to estimate the marginal reweighted cumulative empirical distribution functions under hypothetical values for the θ_j's, as well as the regular marginal cumulative empirical distribution functions. Then, we will use the joint distribution to account for multiplicity and estimate α_{adj}. Finally, we will calibrate the tilting parameters τ_j accordingly.

We estimate the marginal reweighted distributions the same way we do in regular (univariate) bootstrap tilting. From the bootstrap test statistics $\boldsymbol{T}^*_b$, for each $j = 1, 2, \ldots, m$, we estimate the empirical reweighted distribution function

$$\hat{H}^*_{j,\tau_j}(t) = \frac{1}{B}\sum_{b=1}^{B} \mathbb{1}\{T^*_{jb} \le t\}\, W_{jb}(\tau_j), \tag{5.1}$$

where

$$W_{jb}(\tau_j) = \prod_{i=1}^{n} \frac{w_{ij}(\tau_j)}{n^{-1}} = n^n \prod_{i=1}^{n} \frac{e^{\tau_j x_{ij}}}{\sum_{\ell=1}^{n} e^{\tau_j x_{\ell j}}}$$

are the importance sampling weights.

For each $j = 1, 2, \ldots, m$, we plug the test statistic T_j into $\hat{H}^*_{j,\tau_j}$ to obtain $\hat{H}^*_{j,\tau_j}(T_j)$. We will come back to this later when we use the $\hat{H}^*_{j,\tau_j}(T_j)$'s in the context of another cumulative distribution function which we will derive next. Eventually, this will provide the adjustment of the significance level that we need to address the multiplicity problem.

Next, we need to transform the bootstrap test statistics to a comparable scale by deriving univariate pivots and applying a minimum p-value approach, which is equivalent to a max-T approach and will ultimately yield the multiplicity-adjusted bootstrap tilting, or, for short, MABT lower confidence limits.

For each $j = 1, 2, \ldots, m$, we estimate the marginal empirical cumulative distribution function

$$\hat{H}^*_j(t) = \frac{1}{B}\sum_{b=1}^{B} \mathbb{1}\{T^*_{jb} \le t\}$$

from the bootstrap test statistics $T^*_{j1}, T^*_{j2}, \ldots, T^*_{jB}$. We plug these into $\hat{H}^*_j$, which yields transformed bootstrap test statistics $\hat{H}^*_j(T^*_{j1}), \hat{H}^*_j(T^*_{j2}), \ldots, \hat{H}^*_j(T^*_{jB})$, which are now asymptotically uniformly distributed on the unit interval. Note that these

$\hat{H}_j^*(T_{jb}^*)$'s correspond to one minus p-values.

Next, in each bootstrap sample, we determine the bootstrap-wise maximum transformed test statistic $\max_{j=1}^m \hat{H}_j^*(T_{jb}^*)$ and estimate the maximum empirical cumulative distribution function of the transformed test statistics

$$\hat{G}_{\max}^*(x) = \frac{1}{B}\sum_{b=1}^{B} \mathbb{1}\{\max_{j=1}^{m} \hat{H}_j^*(T_{jb}^*) \leq x\}, \tag{5.2}$$

We use this distribution function to estimate the adjusted significance level α_{adj},

$$\hat{\alpha}_{\text{adj}}^* = 1 - \hat{G}_{\max}^{*^{-1}}(1-\alpha), \tag{5.3}$$

where $\hat{G}_{\max}^{*^{-1}}(q) = \inf\{x \mid \hat{G}_{\max}^*(x) \geq q\}$ denotes the corresponding empirical quantile function.

We note that the estimate $\hat{H}_{j,\tau_j}^*$ in Equation (5.1) only concerns the tilting, while Equation (5.3) corresponds to a maximum one minus p-value approach and yields the multiplicity correction.

Lastly, we combine these cumulative distribution functions and present the calibration task in order to get simultaneous lower confidence limits for the θ_j's: Find the values of the tilting parameters $\hat{\tau}_1^{\hat{L}_1}, \hat{\tau}_2^{\hat{L}_2}, \ldots, \hat{\tau}_m^{\hat{L}_m} < 0$ such that for each $j = 1, 2, \ldots, m$ it holds that

$$\hat{H}_{j,\hat{\tau}_j^{\hat{L}_j}}^*(T_j) = 1 - \hat{\alpha}_{\text{adj}}^*.$$

Once the specific values for the $\hat{\tau}_j^{\hat{L}_j}$'s are identified, we can determine the desired simultaneous lower confidence limits via

$$\hat{L}_j = \hat{\theta}[\boldsymbol{w}(\hat{\tau}_j^{\hat{L}_j})]; \tag{5.4}$$

that is, for each j, the multiplicity-adjusted lower confidence limit for θ_j is given as the parameter of interest obtained from the calibrated reweighted distribution $\hat{H}_{j,\hat{\tau}_j^{\hat{L}_j}}^*$. An `R` implementation of this approach can be accessed via a public GitHub repository at `https://gitlab.informatik.uni-bremen.de/s_opbgf3/clfpp/-/blob/main/5-example/MabtCi-function.R`.

We acknowledge that it is not imperative that we use the empirical distribution function within the multiplicity correction. When we marginally transform the bootstrap test statistics, an alternative approach is to simply use the limit

standard-normal distribution of the test statistics, that is,

$$\hat{G}^*_{\max}(x) = \frac{1}{B}\sum_{b=1}^{B} \mathbb{1}\{\max_{j=1}^{m} \Phi(T^*_{jb}) \leq x\}, \tag{5.5}$$

where Φ denotes the cumulative distribution function of the standard-normal distribution. We will investigate this later in simulation experiments, see Section 6.

We need to mention that the adjusted significance level and, thus, the calibrated tilting parameter and lower confidence limits are obviously functions of the global level α, that is, $\hat{\alpha}^*_{\text{adj}} = \hat{\alpha}^*_{\text{adj}}(\alpha)$, $\hat{\tau}_j^{\hat{L}_j} = \hat{\tau}_j^{\hat{L}_j}[\hat{\alpha}^*_{\text{adj}}(\alpha)]$, $\hat{L}_j = \hat{L}_j[1-\hat{\alpha}^*_{\text{adj}}(\alpha)]$, and, thus,

$$\hat{\boldsymbol{L}} = \hat{\boldsymbol{L}}(1-\alpha) = \big(\hat{L}_1[1-\hat{\alpha}^*_{\text{adj}}(\alpha)], \hat{L}_2[1-\hat{\alpha}^*_{\text{adj}}(\alpha)], \ldots, \hat{L}_m[1-\hat{\alpha}^*_{\text{adj}}(\alpha)]\big).$$

We will, however, mostly omit this dependency for notational simplicity.

On a final note, in the present work, we base our approach on test statistics, while in our publication Rink & Brannath (2025), we directly use the parameter of interest instead. Using the test statistic enables us to establish desirable theoretical properties, as we will see in the following section.

5.2 Theoretical properties

Conceptually, because we base our multiplicity correction on the max-T approach, it is not unlikely that the confidence limits in Equation (5.4) are actually *good*. In the context of confidence interval estimates, a desirable property is that they asymptotically have the correct coverage probability, that is, when the sample size n tends to infinity, the coverage probability converges to $(1-\alpha)\cdot 100\%$.

Specifically, we want to prove that for the proposed MABT simultaneous lower $(1-\alpha)\cdot 100\%$-confidence limits $\hat{\boldsymbol{L}}(1-\alpha) = [\hat{L}_1, \hat{L}_2, \ldots, \hat{L}_m]$ for $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_m)$ it holds that

$$\mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^{m}\{\theta_j > \hat{L}_j\}] \to 1-\alpha, \tag{5.6}$$

or equivalently, using the dual test,

$$\mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^{m}\{T_j \leq \hat{H}^{*^{-1}}_{j,\hat{\tau}_j^{\hat{L}_j}}(1-\hat{\alpha}^*_{\text{adj}})\}] \to 1-\alpha, \tag{5.7}$$

where $\theta(\hat{\tau}_j^{\hat{L}_j}) = \hat{L}_j$.

From the upcoming Theorem 5.2.1, which is the main technical contribution of this work, we will conclude that the convergence in Equation (5.7) and, consequently, the convergence in Equation (5.6) holds true.

In order to conduct the proof, we will need some basic concepts and results from asymptotic statistics, which we provide in Appendix A. Thorough discussions of asymptotic statistics can be found, for example, in DasGupta (2008) and van der Vaart (1998). We denote almost sure convergence by $\xrightarrow{\text{a.s.}}$, convergence in probability by $\xrightarrow{\mathbb{P}}$, and convergence in distribution by $\xrightarrow{\mathcal{L}}$.

Theorem 5.2.1. *The following three statements about the proposed* MABT *confidence limits hold true as* $n \to \infty$.

1. *The estimated adjusted significance level converges to the true adjusted significance level, that is,*

$$1 - \hat{G}^{*^{-1}}_{\max}(1-\alpha) = \hat{\alpha}^*_{adj}(\alpha) \xrightarrow{a.s.} \alpha_{adj}(\alpha) = 1 - G^{-1}_{\max}(1-\alpha),$$

uniformly in $\alpha \in [0,1]$, *where* $G_{\max}(x) = \mathbb{P}[\max_{j=1}^m \Phi(T_j) \leq x]$, *and* Φ *denotes the cumulative distribution function of the standard-normal distribution.*

2. *When using the true adjusted significance level* $\alpha_{adj}(\alpha)$, *for some global significance level* $\alpha \in [0,1]$, *the proposed simultaneous confidence limits asymptotically have coverage probability* $(1-\alpha)\cdot 100\%$, *that is,*

$$\mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^m \{\theta_j > \hat{L}_j[1-\alpha_{adj}(\alpha)]\}] \to 1-\alpha,$$

or equivalently, using the dual test,

$$\mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^m \{T_j \leq \hat{H}^{*^{-1}}_{j,\hat{\tau}_j}[\hat{L}_j][1-\alpha_{adj}(\alpha)]\}] \to \mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^m \{T_j \leq \Phi^{-1}[1-\alpha_{adj}(\alpha)]\}] \quad (=1-\alpha).$$

3. *The difference between the coverage probability obtained when using the true and the estimated adjusted significance level converges to zero, that is, for all* $\alpha \in [0,1]$, *it holds that*

$$|\mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^m \{\theta_j > \hat{L}_j[1-\hat{\alpha}^*_{adj}(\alpha)]\}] - \mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^m \{\theta_j > \hat{L}_j[1-\alpha_{adj}(\alpha)]\}]| \to 0,$$

or equivalently, using the dual test,

$$|\mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^m \{T_j \leq \hat{H}^{*^{-1}}_{j,\hat{\tau}_j}[\hat{L}_j][1-\hat{\alpha}^*_{adj}(\alpha)]\}] - \mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^m \{T_j \leq \Phi^{-1}[1-\alpha_{adj}(\alpha)]\}]| \to 0.$$

In the next section, we will prove the three statements from Theorem 5.2.1. They will hold true under the set of regularity conditions DiCiccio & Romano (1990) assumed for their proof of the second-order correctness of the univariate bootstrap tilting confidence interval, see Proposition 4.4.2.

5.3 Proofs

The proof of Theorem 5.2.1 requires the use of the following three Propositions 5.3.1, 5.3.2 and 5.3.3, which we record and prove first.

Proposition 5.3.1 provides conditions under which we can conclude the almost sure and uniform convergence of quantiles from the convergence of the corresponding cumulative distribution functions, and vice versa.

Proposition 5.3.1. *Let $(\Psi_n)_n$ and Ψ be cumulative distribution functions with quantile functions $(\Psi_n^{-1})_n$ and Ψ^{-1}, respectively, such that $\Psi_n(t) \xrightarrow{a.s.} \Psi(t)$ uniformly in t. If the limit cumulative distribution function Ψ is continuous and strictly increasing, it holds that $\Psi_n^{-1}(q) \xrightarrow{a.s.} \Psi^{-1}(q)$ for all $q \in [0,1]$.*

Proof of Proposition 5.3.1. Choose an arbitrary $q \in [0,1]$ and let $\epsilon > 0$. We observe that from

$$\Psi(t) - \epsilon \leq \Psi_n(t) \leq \Psi(t) + \epsilon,$$

it follows

$$\inf\{t \mid \Psi(t) + \epsilon \geq q\} \leq \inf\{t \mid \Psi_n(t) \geq q\} \leq \inf\{t \mid \Psi(t) - \epsilon \geq q\},$$

which is equivalent to

$$\Psi^{-1}(q - \epsilon) \leq \Psi_n^{-1}(q) \leq \Psi^{-1}(q + \epsilon),$$

where Ψ^{-1} is continuous because Ψ is continuous and strictly increasing.

Since $\Psi_n(t) \xrightarrow{\text{a.s.}} \Psi(t)$ uniformly in t, there exists almost surely a number $N \in \mathbb{N}$ such that for all $n \geq N$ it holds that $|\Psi(t) - \Psi_n(t)| < \epsilon$ for all $t \in \mathbb{R}$, that is,

$$\Psi(t) - \epsilon \leq \Psi_n(t) \leq \Psi(t) + \epsilon,$$

and therefore, according to our observation from above,

$$\Psi^{-1}(q - \epsilon) \leq \Psi_n^{-1}(q) \leq \Psi^{-1}(q + \epsilon).$$

Now, since it holds always true that $\liminf_{n\to\infty} \Psi_n^{-1}(q) \leq \limsup_{n\to\infty} \Psi_n^{-1}(q)$, there exists a number $N' \in \mathbb{N}$ such that for all $n \geq N'$,

$$\Psi^{-1}(q - \epsilon) \leq \liminf_{n\to\infty} \Psi_n^{-1}(q) \leq \limsup_{n\to\infty} \Psi_n^{-1}(q) \leq \Psi^{-1}(q + \epsilon).$$

As this is true for all $\epsilon > 0$, and because Ψ^{-1} is continuous, we obtain

$$\Psi^{-1}(q) \leq \liminf_{n\to\infty} \Psi_n^{-1}(q) \leq \limsup_{n\to\infty} \Psi_n^{-1}(q) \leq \Psi^{-1}(q).$$

Therefore, $\liminf_{n\to\infty} \Psi_n^{-1}(q) = \limsup_{n\to\infty} \Psi_n^{-1}(q)$ needs to hold and, hence, $\Psi_n^{-1}(q) \xrightarrow{\text{a.s.}} \Psi^{-1}(q)$. □

The following Proposition 5.3.2 is a generalization of the widely-known and powerful Glivenko-Cantelli Theorem, see, for example, Theorem 19.1 in van der Vaart (1998). It describes how we can obtain almost sure and uniform convergence from pointwise convergence.

Proposition 5.3.2. *Let $\Psi\colon \mathbb{R} \to [0,1]$ be a monotonically increasing function with $\lim_{t\to-\infty} \Psi(t) = 0$ and $\lim_{t\to\infty} \Psi(t) = 1$, and that is continuous from the right. Additionally, let $(\Psi_n)_n$ be a sequence of functions such that Ψ_n converges pointwise to Ψ, that is, as $n \to \infty$, $\Psi_n(t) \to \Psi(t)$ for all $t \in \mathbb{R}$. Then, it follows that this convergence is even uniform in t.*

The proof of Proposition 5.3.2 is rather technical, so we place it in Appendix B and continue.

Previously, in Proposition 4.4.2, we presented the second-order accuracy of the bootstrap tilting confidence intervals. We use this result in the proof of Theorem 5.2.1 in that we are able to conclude from it the convergence in probability of the quantile function of the reweighted distribution to the quantile function of the standard-normal distribution. We present this convergence in Proposition 5.3.3.

Proposition 5.3.3. *The quantile function of the reweighted distribution converges in probability to the quantile function of the standard-normal distribution, that is, for $q \in [0,1]$,*

$$\hat{H}^{*^{-1}}_{\hat{\tau}^{\hat{L}}}(1-q) \xrightarrow{\mathbb{P}} \Phi^{-1}(1-q).$$

Proof of Proposition 5.3.3. Recall that, in order to a find a bootstrap tilting lower $(1-q)\cdot 100\%$-confidence limit for the mean using the test statistic $T = \sqrt{n}\,(\hat{\theta}_n - \theta)/\hat{\sigma}_n$, we need to find the value $\hat{\tau}^{\hat{L}}$ of the tilting parameter such that the probability under the $\hat{\tau}^{\hat{L}}$-reweighted distribution to observe a value of the test statistic larger than the value in the sample is equal to $q \cdot 100\%$; that is,

$$\mathbb{P}_{\hat{\tau}^{\hat{L}}}\left(\sqrt{n}\,\frac{\hat{\theta}_n^* - \hat{\theta}_n}{\hat{\sigma}_n^*} > \sqrt{n}\,\frac{\hat{\theta}_n - \hat{L}}{\hat{\sigma}_n}\right) = q. \tag{5.8}$$

Then, the parameter of the reweighted distribution $\theta(\hat{\tau}^{\hat{L}}) = \hat{L}$ is the desired lower limit.

Solving Equation (5.8) for $\hat{L}$ yields the following expression for the lower confidence limit,

$$\hat{L} = \hat{\theta}_n - n^{-1/2}\hat{\sigma}_n\,\hat{H}^{*^{-1}}_{\hat{\tau}^{\hat{L}}}(1-q). \tag{5.9}$$

From the second-order accuracy of the bootstrap tilting confidence interval, see Proposition 4.4.2, we know that

$$\hat{L} = \hat{\theta}_n - n^{-1/2}\hat{\sigma}_n \, H_n^{-1}(1-q) + \mathcal{O}_{\mathbb{P}}(n^{-3/2}), \tag{5.10}$$

where $H_n^{-1}(1-q)$ is the $(1-q)\cdot 100\%$-quantile of the true distribution of the test statistic, with cumulative distribution function

$$H_n(t) = \mathbb{P}_F \left\{ \sqrt{n}\, \frac{\hat{\theta}_n - \theta(F)}{\hat{\sigma}_n} \leq t \right\}.$$

Equating the two expressions for $\hat{L}$ in Equations (5.9) and (5.10) yields

$$H_n^{-1}(1-q) - \hat{H}_{\hat{\tau}^{\hat{L}}}^{*^{-1}}(1-q) = \mathcal{O}_{\mathbb{P}}(n^{-3/2})\,\sqrt{n}/\hat{\sigma}_n = \mathcal{O}_{\mathbb{P}}(n^{-3/2})\,\mathcal{O}_{\mathbb{P}}(n) = \mathcal{O}_{\mathbb{P}}(n^{-1/2}),$$

which implies that

$$|H_n^{-1}(1-q) - \hat{H}_{\hat{\tau}^{\hat{L}}}^{*^{-1}}(1-q)| \xrightarrow{\mathbb{P}} 0.$$

We know that the asymptotic distribution of the test statistic is standard-normal, and thus, when Φ denotes the cumulative distribution function of the standard-normal distribution,

$$H_n^{-1}(1-q) \to \Phi^{-1}(1-q),$$

pointwise for all $q \in [0,1]$. Thus, we finally conclude that

$$\hat{H}_{\hat{\tau}^{\hat{L}}}^{*^{-1}}(1-q) \xrightarrow{\mathbb{P}} \Phi^{-1}(1-q),$$

and this holds true for arbitrary $q \in [0,1]$. □

Note that Proposition 5.3.3 addresses the univariate case, and we will apply it marginally for each $j = 1, 2, \ldots, m$ later. Next, we return to our main objective and prove Theorem 5.2.1, which is the main technical contribution of this work.

Proof of Theorem 5.2.1

Proof of part (1). We want to prove that

$$1 - \hat{G}_{\max}^{*^{-1}}(1-\alpha) = \hat{\alpha}_{\text{adj}}^{*}(\alpha) \xrightarrow{\text{a.s.}} \alpha_{\text{adj}}(\alpha) = 1 - G_{\max}^{-1}(1-\alpha),$$

uniformly in $\alpha \in [0,1]$.

From the strong consistency of the bootstrap, see Theorem 4.2.2, we know that

$$\sup_{t\in\mathbb{R}} |\hat{H}_j^*(t) - H_{n,j}(t)| \xrightarrow{\text{a.s.}} 0.$$

Since the asymptotic distribution of the test statistic is standard-normal, we know that

$$\sup_{t\in\mathbb{R}} |H_{n,j}(t) - \Phi(t)| \xrightarrow{\text{a.s.}} 0,$$

Consequently,

$$\sup_{t\in\mathbb{R}} |\hat{H}_j^*(t) - \Phi(t)| \xrightarrow{\text{a.s.}} 0,$$

and this holds true for each $j = 1, 2, \ldots, m$. Thus, it holds that

$$|\max_{j=1}^{m} \hat{H}_j^*(T_j) - \max_{j=1}^{m} \Phi(T_j)| \xrightarrow{\text{a.s.}} 0,$$

and because almost sure convergence implies convergence in distribution, we obtain for all $x \in [0, 1]$,

$$\hat{G}_{\max}^*(x) = \mathbb{P}[\max_{j=1}^{m} \hat{H}_j^*(T_j) \leq x] \to \mathbb{P}[\max_{j=1}^{m} \Phi(T_j) \leq x] = G_{\max}(x).$$

Because $\hat{G}_{\max}^*(x)$ and $G_{\max}(x)$ are cumulative distribution functions, Proposition 5.3.2 implies that $\hat{G}_{\max}^*(x) \xrightarrow{\text{a.s.}} G_{\max}(x)$ uniformly in x. Since $G_{\max}(x)$ is continuous and strictly increasing in x, due to Proposition 5.3.1 we can conclude that $\hat{G}_{\max}^{*^{-1}}(\alpha) \xrightarrow{\text{a.s.}} G_{\max}^{-1}(\alpha)$. Finally, we obtain that

$$\hat{\alpha}_{\text{adj}}^*(\alpha) = 1 - \hat{G}_{\max}^{*^{-1}}(\alpha) \xrightarrow{\text{a.s.}} 1 - G_{\max}^{-1}(\alpha) = \alpha_{\text{adj}}(\alpha).$$

Proof of part (2). The dual test formulation of the second part of the theorem reads

$$\mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^{m}\{T_j \leq \hat{H}_{j,\hat{\tau}_j}^{*^{-1}\hat{L}_j}[1 - \alpha_{\text{adj}}(\alpha)]\}] \to \mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^{m}\{T_j \leq \Phi^{-1}[1 - \alpha_{\text{adj}}(\alpha)]\}].$$

To prove this convergence, we define the random variable

$$Z_j = T_j + \Phi^{-1}[1 - \alpha_{\text{adj}}(\alpha)] - \hat{H}_{j,\hat{\tau}_j}^{*^{-1}\hat{L}_j}[1 - \alpha_{\text{adj}}(\alpha)]\}],$$

and show that

$$\mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^{m}\{Z_j \leq \Phi^{-1}[1 - \alpha_{\text{adj}}(\alpha)]\}] \to \mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^{m}\{T_j \leq \Phi^{-1}[1 - \alpha_{\text{adj}}(\alpha)]\}].$$

Due to Proposition 5.3.3, $|\Phi^{-1}(q) - \hat{H}^{*^{-1}}_{j,\hat{\tau}_j^{\hat{L}_j}}(q)| \xrightarrow{\mathbb{P}} 0$ for all $q \in [0,1]$, and because $|\Phi^{-1}(q) - \hat{H}^{*^{-1}}_{j,\hat{\tau}_j^{\hat{L}_j}}(q)| = |Z_j - T_j|$, we conclude that

$$|Z_j - T_j| \xrightarrow{\mathbb{P}} 0,$$

and this holds true for each $j = 1, 2, \ldots, m$. Let $\boldsymbol{Z}$ and $\boldsymbol{T}$ denote the vectors $(Z_1, Z_2, \ldots, Z_m)$ and $(T_1, T_2, \ldots, T_m)$, respectively, and let $\|\cdot\|_1$ denote the ℓ_1-norm. Then, for any $\epsilon > 0$ it holds that

$$\mathbb{P}(\|\boldsymbol{Z} - \boldsymbol{T}\|_1 \geq \epsilon) \leq \mathbb{P}\Big(\max_{j=1}^{m} |Z_j - T_j| \geq \frac{\epsilon}{m}\Big) \leq \sum_{j=1}^{m} \mathbb{P}\Big(|Z_j - T_j| \geq \frac{\epsilon}{m}\Big) \to 0$$

as $n \to \infty$, for any $j \in \{1, 2, \ldots, m\}$. Hence, $\|\boldsymbol{Z} - \boldsymbol{T}\|_1 \xrightarrow{\mathbb{P}} \mathbf{0}$ and, in particular,

$$\|\boldsymbol{Z} - \boldsymbol{T}\|_1 \xrightarrow{\mathcal{L}} \mathbf{0}.$$

Thus, we obtain that

$$\mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^{m}\{Z_j \leq \Phi^{-1}[1 - \alpha_{\text{adj}}(\alpha)]\}] \to \mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^{m}\{T_j \leq \Phi^{-1}[1 - \alpha_{\text{adj}}(\alpha)]\}],$$

and the limit is equal to $1 - \alpha$ by construction.

Proof of part (3). We use the dual test again to prove the third part of the theorem and show that

$$|\mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^{m}\{T_j \leq \hat{H}^{*^{-1}}_{j,\hat{\tau}_j^{\hat{L}_j}}[1 - \hat{\alpha}^*_{\text{adj}}(\alpha)]\}] - \mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^{m}\{T_j \leq \Phi^{-1}[1 - \alpha_{\text{adj}}(\alpha)]\}]| \to 0.$$

In order to prove this convergence, we show that $|\Delta_1^{(n)}(\alpha) - \Delta_2^{(n)}(\alpha)| \to 0$ as $n \to \infty$, where

$$\Delta_1^{(n)}(\alpha) = \mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^{m}\{T_j \leq \hat{H}^{*^{-1}}_{j,\hat{\tau}_j^{\hat{L}_j}}[1 - \hat{\alpha}^*_{\text{adj}}(\alpha)]\}] - \mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^{m}\{T_j \leq \Phi^{-1}[1 - \hat{\alpha}^*_{\text{adj}}(\alpha)]\}],$$

and

$$\Delta_2^{(n)}(\alpha) = \mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^{m}\{T_j \leq \Phi^{-1}[1 - \hat{\alpha}^*_{\text{adj}}(\alpha)]\}] - \mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^{m}\{T_j \leq \Phi^{-1}[1 - \alpha_{\text{adj}}(\alpha)]\}].$$

From part (2) of Theorem 5.2.1, we obtain the pointwise convergence

$$|\mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^{m}\{T_j \leq \hat{H}^{*^{-1}}_{j,\hat{\tau}_j^{\hat{L}_j}}(q)\}] - \mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^{m}\{T_j \leq \Phi^{-1}(q)]\}]| \to 0,$$

for any $q \in [0,1]$. Due to Proposition 5.3.2, this convergence is even uniform in

q. According to part (1) of Theorem 5.2.1, for any fixed $\alpha \in [0,1]$, $\hat{\alpha}^*_{\text{adj}}(\alpha) \xrightarrow{\text{a.s.}} \alpha_{\text{adj}}(\alpha)$. Therefore, as $n \to \infty$, it follows that

$$|\Delta_1^{(n)}(\alpha)| \leq \sup_{q\in[0,1]} |\mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^m \{T_j \leq \hat{H}^{*^{-1}}_{j,\hat{\tau}_j}(q)\}] - \mathbb{P}_{\boldsymbol{\theta}}[\cap_{j=1}^m \{T_j \leq \Phi^{-1}(q)]\}]| \to 0.$$

Since, by part (1) of Theorem 5.2.1, $1-\hat{\alpha}^*_{\text{adj}}(\alpha) \xrightarrow{\text{a.s.}} 1-\alpha_{\text{adj}}(\alpha)$, it follows from the Continuous Mapping Theorem, see Proposition A.1 in Appendix A, that

$$\Phi^{-1}[1-\hat{\alpha}^*_{\text{adj}}(\alpha)] \xrightarrow{\text{a.s.}} \Phi^{-1}[1-\alpha_{\text{adj}}(\alpha)],$$

uniformly in α. Because the joint asymptotic distribution of $(T_1, T_2, \ldots, T_m)$ is m-dimensional multivariate standard-normal, which has a continuous multivariate distribution function, it follows that $|\Delta_2^{(n)}(\alpha)| \to 0$.

Consequently, in conclusion, $|\Delta_1^{(n)}(\alpha) - \Delta_2^{(n)}(\alpha)| \to 0$ pointwise in α.

This concludes the proof of Theorem 5.2.1. Hence, the simultaneous lower limits

$$\hat{\boldsymbol{L}}(1-\alpha) = \big(\hat{L}_1[1-\hat{\alpha}^*_{\text{adj}}(\alpha)], \hat{L}_2[1-\hat{\alpha}^*_{\text{adj}}(\alpha)], \ldots, \hat{L}_m[1-\hat{\alpha}^*_{\text{adj}}(\alpha)]\big)$$

for $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_m)$ asymptotically have simultaneous coverage probability $(1-\alpha)\cdot 100\%$, that is, Equations (5.6) and (5.7) hold true.

In particular, MABT yields valid lower conditional confidence limits when we use any of the performance measures from Section 2.2.2, independent of the model selection strategy, whether formal or informal, and even when it is based on post-hoc considerations.

Chapter 6

Simulation Experiments

The considerations in Chapter 5 revealed good and desirable theoretical properties of the proposed MABT confidence limits. To complement these asymptotic results, we investigate the goodness of the proposed approach in finite samples using extensive simulation experiments.

We will assess the goodness of the lower confidence limits on the basis of three indicators: conditional coverage probability, the confidence limits themselves, and their informativeness. In addition, we will approximate and compare the true performances of the models selected from the default and the proposed selection-evaluation pipeline, respectively.

Coverage probability This is the proportion of times that a confidence interval, constructed from multiple independent samples, contains the true value for the parameter of interest. Assessing the coverage probability is essential to understand the reliability and accuracy of a confidence interval. Sample size and model or distributional assumptions may affect it. When it is much lower than expected, this indicates that the interval underestimates the true uncertainty in the parameter estimate, and vice versa. In the former case the method is called *anti-conservative* and in the latter *conservative*.

Size of the lower confidence limit In a real-world scenario, the user does not know the truth and assesses the predictive performance by the lower confidence limit. In the proposed selection-evaluation pipeline, we need to account for the present multiplicity, such that it is not immediately clear whether we will actually gain from MABT relative to comparative methods. It might well be the case that the expected gains due to the preselection of multiple candidate models for evaluation are counterbalanced by the multiplicity correction. In other words: There is no real use in finding a better prediction model regarding true predictive perfor-

mance when we are unable to identify it as such via a larger lower confidence limit.

Informativeness of the lower confidence limit To capture the informativeness, we will compute the distance between the true performance and the lower confidence limit, and will refer to this as *tightness* for brevity. We do not have clear expectations regarding the tightness of the MABT limits compares to other methods. This is largely due to observations by Hall (1988) that the coverage probability of a bootstrap confidence interval is not necessarily directly related to its informativeness.

True performance For each simulated data set, we will obtain two selected models, one from the default selection-evaluation pipeline and one from the proposed pipeline. Due to the gainful way of the latter approach, we expect the true performance of the final selected model to be better. Nevertheless, we will not be able to affect model training by, for instance, training on a larger fraction of the sample, so that we do not expect radical improvements in general.

This chapter is organized as follows. In Section 6.1, we will pose and discuss the questions we would like to answer using simulation experiments. In Section 6.2, we will explain in detail how we structure and conduct the experiments. Then, in Section 6.3, we will present the results.

The full `R` implementation of the simulation experiments can be accessed and downloaded from a public GitHub repository at `https://gitlab.informatik.uni-bremen.de/s_opbgf3/clfpp`.

We will base most of the simulation experiments in the present work on the experiments conducted in our publication Rink & Brannath (2025). In line with the theoretical results obtained in the previous chapter, we will use test statistics here, while in the publication we used the parameter of interest directly.

6.1 Objectives

There are three major questions that we will address separately. Broadly speaking, they concern a design choice within the multiplicity adjustment in the proposed confidence limits; alternative methods that use the entire sample to estimate a confidence limit for conditional performance; and how the proposed intervals compare to standard methods, which will be our main concern. We will discuss these issues in the following Sections 6.1.1, 6.1.2, and 6.1.3.

6.1.1 Two variants of multiplicity correction

In Section 5.1 we acknowledge that it is not imperative that we use the empirical distribution function $\hat{H}_j^*$ within the proposed multiplicity correction in Equation (5.2). Another viable option is to use the standard-normal distribution, as the considered test statistic is asymptotically standard-normally distributed. It will be interesting to see whether the additional work of estimating the empirical distributions pays off regarding the coverage probability as well as the size and tightness of the estimated lower confidence limits.

Note that the considerations on the two variants of multiplicity correction are not a part of our publication Rink & Brannath (2025).

6.1.2 Conditional performance

There are proposals in the literature advocating the use of the entire sample for model training, selection, and evaluation. They include *bootstrap bias-corrected cross-validation* (Tsamardinos et al., 2018) and *nested cross-validation* (Bates et al., 2024), or, for short, BBC-CV and NCV, respectively.

The ad-hoc lower $(1-\alpha)\cdot 100\%$-confidence limit estimate from cross-validation is given by $\hat{\theta}_{\text{CV}} - z_{1-\alpha}\,\hat{s}_{\text{CV}}$, where $z_{1-\alpha}$ denotes the $(1-\alpha)\cdot 100\%$-quantile of the standard-normal distribution and

$$\hat{s}_{\text{CV}} = \sqrt{\hat{\sigma}^2_{\text{CV}}/n}. \tag{6.1}$$

This interval relies on standard-normal quantiles, and, additionally, is known to not accurately track conditional performance; see, for example, Hastie et al. (2009) and Bates et al. (2024).

Both BBC-CV and NCV improve on the ad-hoc cross-validation interval and claim that they yield confidence intervals for conditional performance, which makes them direct competitors to our proposed MABT intervals as well as to standard methods that rely on a held-out evaluation set for interval estimation. We will characterize both competitors and compare their results from the default pipeline with MABT later in Section 6.2.3. Note, however, that both BBC-CV and NCV could in principle be extended to the proposed selection-evaluation pipeline by incorporating a multiplicity correction, but we will not consider this in the present work.

Being able to train the model on the entire sample and eliminating the need for a held-out evaluation set (and for a multiplicity correction) are tremendous advantages over MABT and standard methods. While it is clear that methods that use a held-out evaluation set yield intervals for conditional performance, the

situation is more intricate for methods such as BBC-CV and NCV.

It is not straightforward to even design simulation experiments that investigate the ability of a method to produce confidence intervals for conditional performance, especially when no sample splitting should be involved.

Designing a simulation experiment for the unconditional coverage probability is fairly obvious, though. First, we need to decide on a classifier and fix the hyperparameters, for example, the LASSO and the value of the ℓ_1 penalty λ. Then, we need to repeatedly draw samples of the same size from a prespecified distribution. In each sample, we apply the LASSO with hyperparameter λ, estimate a confidence interval, and check whether it contains the true performance value or not. The proportion of such covering intervals among all estimated intervals is an estimate of the unconditional coverage probability. Note that this is a statement about the classifier together with the data-generating process.

In the conditional case, on the other hand, the design is somewhat more intricate, as we need to fix both the classifier with all its hyperparameters as well as the model parameters; that is, it does not suffice to specify the LASSO as the classifier of our choice together with a specific value for the hyperparameter λ, but we also need to fix the coefficients $\hat{\boldsymbol{\beta}}_\lambda$. Because we obtain the latter from the sample at hand, we would need to draw the samples in such a way that we always estimate the exact same vector $\hat{\boldsymbol{\beta}}_\lambda$. We will go into more detail later.

In order to compare BBC-CV and NCV to MABT, we need to investigate how well the respective intervals track the conditional performance. Additionally, we will also investigate the goodness of the three methods in the presence of a distribution shift, that is, the distribution between learning and validation changes. We will present the setups for these experiments in Section 6.2.3.

6.1.3 Comparison to standard methods

The MABT intervals have promising asymptotic properties, which suggest their potential theoretical effectiveness. However, to fully assess their practical effectiveness, we need to test them in finite samples against several popular and widely-used standard methods and assess how they compare in a variety of settings. This will provide a better understanding of whether MABT intervals constitute any improvement over the standard methods, and to which extent.

6.2 Setups

In this section, we will describe the simulation experiments that we set up to address the questions previously raised.

In general, we will simulate under a typical machine learning model selection and evaluation setup: We generate data for a binary classification task, learn several prediction models, select the most-promising one among them, and evaluate its conditional performance using a lower confidence limit. We will provide much more detail on the data generation, model training, performance estimation, model selection, and the estimation of confidence limits in Section 6.2.1.

Note that we will only report a single multiplicity-adjusted confidence limit for the final selected model here instead of simultaneous confidence limits for all the considered prediction models. Of course, it would also be possible to do the latter. However, the former is a more practical and concise scenario within machine learning applications.

For the descriptions of the simulation setups, we will change the order in which we address the issues from the previous section to make the presentation clearer. In particular, for the issues presented in Sections 6.1.1 and 6.1.2, we will rely on ideas and strategies or even reuse parts of the experiments we design for the comparison of MABT to standard procedures. Hence, we will begin with the description of these experiments. Then, we will address the other two objectives.

6.2.1 Comparison to standard procedures

In the following, we will address the various aspects of the experimental setup, that is, how we generate the data, what type of prediction models we fit, which measures we use to assess the performance, how we select a model for evaluation and obtain a lower confidence limit, and how we estimate coverage probability.

Data generation We will split the data into two parts, a so-called *learning set* that combines the training and the validation set and contains 75 percent of the observations, and a held-out *evaluation set*, which contains the remaining 25 percent. We will generate medium-sized samples of 400 observations, so there are 300 observations available for learning and 100 for evaluation.

In addition to the learning and the evaluation set, from the same distribution, we sample another much larger data set of 20 000 observations. We will refer to the latter, slightly misleading, as the *population*. It will serve as a ground truth from which we will later derive the true model performances.

During our experiments, we will generate the features and true class labels in our data sets in two different ways. They represent two different levels of data complexity. Still, both setups will lead to balanced classes.

In the first and simpler case, which we will refer to as the *normal feature* case, we draw uncorrelated random numbers from the standard-normal distribution.

We arrange these numbers into an n-by-k feature matrix $[\boldsymbol{x}_i]_i$ with rows $\boldsymbol{x}_i$, where k is the number of features.

To obtain the true class labels y_i, we specify a sparse true coefficient vector $\boldsymbol{\beta}$ with only one percent non-zero coefficients. Then, using the inverse logit function and a vector of uncorrelated observations r_i that are uniformly distributed on the unit interval, the true class labels are obtained via

$$y_i = \mathbb{1}\Big\{\frac{1}{1+e^{-\boldsymbol{x}_i\boldsymbol{\beta}}} \geq r_i\Big\}.$$

The reasoning behind the use of the r_i's is the following. From the inverse logit function, we obtain probabilities of the i-th observation with features $\boldsymbol{x}_i$ to fall into the positive class. These numbers are deterministic, because they deterministically depend on the features $\boldsymbol{x}_i$. So, until this point, regarding the class labels there is no randomness involved.

In order to introduce some, we generate the r_i's, that is, random sampling from the uniform distribution on the unit interval. We assign the positive class to the i-th observation when the inverse logit is larger than r_i. This means, the chance to fall into the positive class is high when the inverse logit is large. But in some rare cases, when r_i is particularly large, the observation might still fall into the negative class. Other than that, it is likely that the observation falls into the negative class when the inverse logit itself is rather small. In this way, we introduce chance, which we need to proceed.

Specifically, we generate $k = 1000$ features and choose the true coefficient vector to be $\boldsymbol{\beta} = c \cdot (1, 1, \ldots, 1, 0, 0, \ldots, 0)$ to have only ten non-zero entries, that is, only one percent of the features actually contribute to the class label. The factor $c > 0$ represents the signal strength, that is, the clarity and quality of the underlying relationship between the features and the true class labels that the prediction models will try to learn from. When c gets larger, it is easier to detect the signal and separate it from noice. We choose $c = 1$, as this seems to make the classification task sufficiently hard.

In summary, considering the shape of $\boldsymbol{\beta}$, the signal strength, and the underlying standard-normal distribution of the features, we provide a sparse and not particularly strong signal.

The second case, the *`caret` feature case*, we use the `twoClassSim` function from the `caret R` package by Kuhn (2008) to mimic a more complex relationship between the features and the true class labels, which is closer to real-world data than the normal feature case. The function generates a feature matrix that includes linear and non-linear effects as well as noise variables, each both uncorrelated and correlated, where we choose a constant correlation of 0.8. In addition,

one percent of the data is mislabeled.

In particular, the probability of the i-th observation falling into the positive class against the negative is the i-th component of the probability vector $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_n)$ in Equation (C.1), which we place in Appendix C.3. Then, a random number r_i is drawn from the uniform distribution on the unit interval in order to assign the true class label $y_i = \mathbb{1}\{\pi_i > r_i\}$.

Model training For model training, we consider LASSO and random forest classifiers, which we introduced in Section 2.3.1. For brevity, we will use the LASSO classifier only on the *normal feature data*, and the random forests only on the ***caret*** *feature data*. This is an appropriate choice regarding the complexities of the data and the classifiers.

For the LASSO prediction models, we determine the maximum regularization parameter $\lambda_{\max}$ as shown in Equation (2.2), and fit LASSO models with 100 equidistant values for λ between zero and $\lambda_{\max}$. Note that $\lambda_{\max}$ depends on the data at hand. We leave its computation to the `glmnet R` function from the `glmnet R` package by Friedman et al. (2010).

For the random forest prediction models, we use the `tuneRanger R` function from the `tuneRanger R` package. This function automatically tunes the involved hyperparameters of a random forest, that is, number of features to possibly split at in each node, minimal node size, and the fraction of observations to sample, with model-based optimization, see Probst et al. (2018). Again, we fit 100 competing models.

Performance estimation For both, the LASSO and the random forest, we aim for a lower confidence limit for the conditional prediction accuracy of the final selected model.

However, note that in both the default and the proposed selection-evaluation pipeline, we estimate the performance of the prediction models at two different stages. We will go into more detail on model selection in both pipelines in the next section.

First, we need to estimate the prediction performances of all candidate models. In case of the LASSO classifier, we use ten-fold cross-validation on the learning data. For the random forest classifier, we use a resampling-based approach, as implemented in the `tuneRanger R` function that we already use for hyperparameter tuning.

Note that the cross-validation performance estimates cannot be interpreted as

conditional estimates. Rather, as they are averaged over the ten folds, they are unconditional estimates. Similar applies for the resampling-based approach in the random forest case. However, we use the cross-validation performances only to identify promising prediction models to preselect for evaluation.

Selecting a model for evaluation in the learning phase means that we fix the associated hyperparameters; the model parameters, that is, the estimated coefficients, in contrast, might change. This is due to the fact that during learning and model selection for evaluation, we usually obtain performance estimates for models that do not use the entire learning data. For example, when cross-validation is employed, for each fold we obtain a performance estimate, but none of those come from a model that is trained on the entire learning set. During learning, our main goal is to identify the most promising hyperparameter constellations. Once these are identified, the models are refitted on the entire learning set, and the second stage of performance estimation follows. Using the evaluation data, we obtain conditional estimates of their generalization performances.

Model selection Regarding model selection, we need to distinguish between the default and the proposed selection and evaluation pipeline. In the default pipeline, only a single model is selected for evaluation. This will be the one with the best cross-validation performance in the learning set, that is, the best average performance across cross-validation folds.

In contrast, when we employ the proposed pipeline, model selection is conducted at two different stages. First, based on their cross-validation performance, promising models are identified and preselected for evaluation. In the second stage, a final model is selected among them, based on its evaluation performance, that is, the generalization performance on the held-out evaluation set, conditional on the hyperparameters and coefficients estimated from the learning data.

In general, we do not need to use the same selection rule in both stages. It is even possible to use one performance measure for preselection and another one for the final selection. We do not consider this here, though. Instead, in both stages, for preselection and final selection, we will select models based on their prediction accuracy.

There are many ways to preselect multiple prediction models. One apparent option is to rank the candidate models due to their predictive performance on the learning set and preselect a fixed number or a proportion. In particular, we will select the top ten percent of models for evaluation and call this selection rule the *top ten percent* rule, accordingly.

Another idea is to perform a data-driven selection. In particular, based on the

cross-validation performances, we identify the best-performing model and compute the cross-validation standard error $\hat{s}_{\text{CV}}$ from Equation (6.1). Then, we preselect the best-performing model and all models that are within one standard-error of the best-performing one for evaluation. We will refer to this selection rule as the *within one standard-error* selection rule.

Since the standard-error estimate is directly obtained from the cross-validation, it reflects the uncertainty of the average performance estimate itself and, thus, the uncertainty with which a model might be identified to be the most promising one. The standard-error is smaller when the performance estimate of the best-performing model is rather precise, and larger when it fluctuates greatly between cross-validation folds. Thus, less or more models fall into the margin, respectively.

Another dimension to this is that when many candidate models all perform comparably well during cross-validation, more models will be preselected. At the same time, it is possible that only a few candidate models will be preselected in case there are only a few models that perform clearly better than the rest.

In this way, the *within one standard-error* rule is more adaptive than the *top ten percent* rule, but we can only consider it when we employ cross-validation during learning, as in the LASSO experiments. In the random forest experiments, therefore, we will use the *top ten percent* selection rule. The `tuneRanger` implementation does not directly allow for a similar adaptive rule, as it typically does not yield repeated performance estimates per hyperparameter configuration. Rather, it averages the hyperparameters of several promising candidate models; see Section 3.5 in Probst et al. (2018) for details.

Due to the way we address the multiplicity, a larger number of preselected models typically needs a stronger correction, which might yield unnecessarily conservative confidence intervals; if more models are preselected, the probability is higher that any one of them performs better in a bootstrap sample than the final selected model. This shifts the maximum distribution to the right. This is also the reason why we preselect a subset of models for preselection and do not evaluate all candidate models. Note, however, that a high correlation between the predictions from the different candidate models reduces the amount of multiplicity correction needed, as we illustrated in Figure 3.3.

In view of the uncertainty in the cross-validation performance estimates, it seems reasonable to select multiple models for evaluation instead of only a single one; we want to avoid the exclusion of a promising model from evaluation just because it scored slightly worse than the best one. Later, in Section 7.2, we will see that even repeated cross-validation does not eliminate this issue completely.

Moreover, the cross-validation estimates cannot be understood as estimates of conditional performance. Rather, they estimate the unconditional performance,

and it is not immediately clear how both interact.

Once models are preselected, they are refitted using the entire learning set and the final selection is done based on the prediction performance in the evaluation set.

It might well happen that multiple models yield the same performance in the evaluation set. In that case, we select the least complex of those. When dealing with LASSO models, the least complex is the one with the largest value of the regularization hyperparameter λ, which corresponds to the one with the smallest number of features. In case of random forest prediction models, we use the computation time for fitting the model as a substitute for model complexity, and we regard models with shorter computation time as less complex.

Confidence limits For each simulated data set, we will obtain two selected models, one from the default pipeline and one from the proposed pipeline. For each of the models, we will employ different competing interval methods in order to estimate lower limits for the prediction accuracy, using the evaluation set.

In the default pipeline, there is no need for multiplicity adjustment, as only a single model is selected for evaluation. Its performance estimates in the evaluation set are unbiased estimates of their respective conditional performance given the fitted prediction model. Here, we will compute lower confidence limits using the standard approaches presented in Section 3.3.2, that is, the Wald normal approximation interval, the Wilson interval, and the CP interval, each using the full global significance level $\alpha = 5\%$. While the latter is an exact method, both the Wald and the Wilson intervals rely on normal approximations, and it is unclear how well this assumption applies.

In the proposed pipeline, however, in order to control the family-wise error rate, we must adjust for multiplicity due to the multiple models being evaluated. Hence, we compute our proposed MABT lower confidence limits, as they control for multiplicity independent of the selection rule applied, using 10 000 bootstrap samples. Also, they do not assume approximate normality, which we expect to result in better preservation of the nominal coverage probability. We provide a formal algorithmic description of the MABT limits in Appendix C.1.

Another idea that comes to mind is to use the same comparative methods as in the default pipeline, but with a simple multiplicity adjustment such as the Bonferroni or the Šidák correction, to adjust for the multiple models selected for evaluation. As we can safely assume that the predictions from the different candidate prediction models are not negatively dependent, we choose to use the Šidák-adjusted significance level $\alpha_{\text{Šidák}} = 1-(1-\alpha)^{1/m}$ over the Bonferroni level, as

it is less conservative. It is unclear whether there is additional gain in using MABT over the Šidák-corrected limits, as the former incorporates information about the dependency structure of the predictions from the preselected models.

True performance and coverage probability For each simulated data set we will compute the *true* performances of the two selected models obtained from the two pipelines by applying the models to the *population* and comparing the predicted classes to the true labels. This is an obvious imprecision in our considerations, as the population is a finite sample of 20 000 observations. Therefore, we will only obtain an estimate instead of the true prediction accuracy itself. However, these estimates will be fairly precise, as they are expected to differ by no more than $\pm\sqrt{0.5^2/20\,000} = \pm 0.0035$ from the truth.

Then, for each competing interval method, we will compare the lower limit to the respective estimated true performance and decide whether it is covered by the induced confidence interval. To estimate the coverage probability of an interval method, we will report the proportion of covering intervals.

Comparison to Rink & Brannath (2025) The general difference to our publication Rink & Brannath (2025) is, as already mentioned, the use of test statistics instead of the parameter of interest directly. Moreover, we only present a subset of the scenarios from Rink & Brannath (2025) here, as the results are consistent across the scenarios. For a list of all the scenarios considered in the publication, see Appendix C.2. Moreover, the simulation experiments in Rink & Brannath (2025) additionally consider confidence limits from (univariate) bootstrap tilting. In the default pipeline, they turned out similarly competitive as the Wilson limits, while the Šidák-adjusted version was clearly outperformed by the MABT limits in the proposed pipeline. However, to keep the presentation in this work concise, we will not show detailed results here.

6.2.2 Two variants of multiplicity correction

To address the question raised in Section 6.1.1 whether to use the empirical or the standard-normal distribution function within the multiplicity correction of our proposed intervals, we will use the LASSO simulation experiments and compute an additional lower confidence limit that utilizes the standard-normal distribution instead of the empirical distribution, that is, we estimate $G_{\max}$ using Equation (5.5) instead of Equation (5.2). We will present the results in Section 6.3.1.

6.2.3 Conditional performance

In Section 6.1.2 we began to discuss the investigation of conditional coverage probability. In particular, we noted that we would need to generate the data sets in such a way that we consistently arrive at the exact same model, which includes both model and hyperparameters.

In the following, we will argue why it is not inherently possible for BBC-CV and NCV to estimate conditional coverage probability, while it is for MABT. We begin with a short description of how both cross-validation variants work.

Recall that cross-validation does not only yield a selection of a candidate model, but in addition, for each observation, we obtain a single prediction $\hat{y}_{ij}$ of the i-th observation from the j-th candidate model.

BBC-CV The BBC-CV approach by Tsamardinos et al. (2018) repeatedly draws bootstrap samples from the $\hat{y}_{ij}$'s. Let $\mathcal{I}_b$ denote the index set of observations drawn into the b-th bootstrap sample, and let $\mathcal{I}_b^c = \{1, 2, \ldots, n\} \setminus \mathcal{I}_b$ denote the index set of non-drawn observations. For each candidate model, the procedure computes an in-bootstrap-sample performance estimate according to $\mathcal{I}_b$. Typically, in a binary classification setting, this could be the in-bootstrap-sample prediction accuracy $\sum_{i \in \mathcal{I}_b} \mathbb{1}\{\hat{y}_{ij} = y_i\}/|\mathcal{I}_b|$. Next, for each bootstrap sample b, the procedure identifies the best model according to the in-bootstrap-sample performance and estimates the out-of-bootstrap-sample performance of this model according to $\mathcal{I}_b^c$. This yields a series of out-of-bootstrap-sample estimates. BBC-CV reports the lower α-quantile of these estimates as the lower $(1-\alpha) \cdot 100\%$-confidence limit for the candidate prediction model selected from regular cross-validation.

NCV The NCV approach to confidence interval estimation was proposed by Bates et al. (2024). It eventually relies on normal approximations to estimate the confidence interval. However, instead of the regular cross-validation estimate, it uses a nested scheme to estimate the standard-error. In order to stabilize the estimate, the procedure repeatedly performs NCV, including model training, which results in many additional model trainings, and finally aggregates the results.

Both of these variants of confidence interval estimation for cross-validation claim to be adapted to the conditional case. However, they operate on the entire sample and do not leave a held-out evaluation set. Also, as cross-validation is considered to yield unconditional estimates, it is not immediately clear why both variants should lead to valid conditional intervals, as they rely on correction terms.

MABT, in contrast, yields inherently conditional estimates, as it only operates on the predictions from a hold-out evaluation set. In addition, it is easy to design a simulation experiment to assess its conditional coverage properties. The prediction models including model and hyperparameters are fixed after the learning phase. Thus, they are independent of the evaluation set and the interval estimation. Consequently, we can fix the learning set and with it the fitted models, repeatedly sample evaluation sets, and estimate the respective confidence intervals. This way, we obtain an estimate of conditional coverage probability.

Regular scenario In particular, we will slightly modify the LASSO simulation experiments and generate medium-sized random samples of 400 observations, split into 90 percent for learning and ten percent for evaluation. We generate the data analogous to the LASSO experiments in Section 6.2.1; that is, 1000 uncorrelated features randomly drawn from the standard-normal distribution, a sparse true coefficient vector with ten non-zero coefficients and signal strength one, and balanced classes.

A single learning set remains fixed between simulation runs. We use ten-fold cross-validation to compare 100 candidate LASSO models, that is, 100 equidistant values $\lambda_1, \lambda_2, \ldots, \lambda_{100}$ for the ℓ_1 penalty λ between zero and $\lambda_{\max}$, which we obtain from the `cv.glmnet` function from the `glmnet R` package (Friedman et al., 2010).

For each candidate model $j = 1, 2, \ldots, m$, ten-fold cross-validation yields a prediction accuracy estimate $\hat{\theta}_{\text{CV},j}$ and a corresponding estimated coefficient vector $\hat{\boldsymbol{\beta}}_{\lambda_j}$. Based on the $\hat{\theta}_{\text{CV},j}$'s, we use the *within one standard-error* selection rule to preselect m models $\hat{\boldsymbol{\beta}}_{\lambda_{s_1}}, \hat{\boldsymbol{\beta}}_{\lambda_{s_2}}, \ldots, \hat{\boldsymbol{\beta}}_{\lambda_{s_m}}$ for evaluation.

For each of these preselected models, we estimate the true prediction accuracy θ_{s_j} from a large sample of 20 000 observations from the same distribution. We will again refer to this sample as the *population*.

The simulation experiment runs along $D = 100\,000$ generated evaluation samples. On each of these, we predict the true class labels using the preselected models. This way, we obtain evaluation performance estimates $\hat{\theta}_{s_j}^{(d)}$ and a final model selection $s(d) = \text{argmax}\{\hat{\theta}_{s_j} \mid j = 1, 2, \ldots, m\}$. Finally, we use 10 000 bootstrap samples to estimate a MABT lower $(1-\alpha)\cdot 100\%$-confidence limit $\hat{L}_{s(d)}^{(d)}$ and check whether the interval contains the true prediction accuracy $\theta_{s(d)}$, that is, $\theta_{s(d)} > \hat{L}_{s(d)}^{(d)}$.

We proceed in this fashion for each evaluation sample $d = 1, 2, \ldots, D$. Hence, we obtain repeated measurements per preselected model s_j and, thus, estimate the conditional coverage probability $\hat{p}_{s_j}^{\text{cvg}}$ individually for each preselected model

s_j as the proportion of covering intervals among all intervals,

$$\hat{p}_{s_j}^{\text{cvg}} = \frac{\sum_{d:\, s(d)=s_j} \mathbb{1}\{\hat{L}_{s(d)}^{(d)} < \theta_{s(d)}\}}{\#\{d:\, s(d) = s_j\}}, \tag{6.2}$$

for each $j \in \{1, 2, \ldots, m\}$.

Next, we will modify the design to investigate the conditional coverage properties of both BBC-CV and NCV.

As mentioned earlier, we would actually need to first specify the desired model and hyperparameters and then repeatedly draw samples such that model training and cross-validation yield these exact prediction models. While it is easy to specify the candidate values for the hyperparameters λ in a LASSO regression, since both operate on the entire sample, it is practically infeasible to obtain the exact same estimated coefficient vector $\hat{\boldsymbol{\beta}}_\lambda$ from different samples.

As an approximation, we suggest to weaken the conditioning. In particular, for BBC-CV and NCV, we estimate the respective coverage probabilities conditional on the hyperparameters, but not on the model parameters.

Therefore, we modify the simulation design for MABT such that it is applicable to BBC-CV and NCV and present it alongside the MABT design in Figure 6.1.

Specifically, we use the same 360 observations in the single learning set and the same 100 candidate values $\lambda_1, \lambda_2, \ldots, \lambda_{100}$ for the ℓ_1 penalty. Then, we combine the learning sample with each of the D evaluation samples. We will refer to these samples as *learning-evaluation* samples.

On each learning-evaluation sample, ten-fold cross-validation yields prediction accuracy estimates $\hat{\theta}_1^{(d)}, \hat{\theta}_2^{(d)}, \ldots, \hat{\theta}_{100}^{(d)}$. Then, as usual, we select the best-performing model $\tilde{s}(d) = \text{argmax}\{\hat{\theta}_j^{(d)} : j = 1, 2, \ldots, 100\}$. Note that the selection here does not necessarily need to be the same as for the MABT case, that is, in general, $\tilde{s}(d) \neq s(d)$.

For the selected model $\tilde{s}(d)$, we estimate a corresponding lower $(1-\alpha)\cdot 100\%$-confidence limit using BBC-CV and NCV, respectively, and check whether the respective confidence interval contains the true prediction accuracy $\theta_{\tilde{s}(d)}$.

For BBC-CV, we use 10 000 bootstrap samples, and for NCV we use the default value of 50 repetitions of nested cross-validations to combine.

Because the model selections differ between the D learning-evaluation samples, we get repeated measurements per candidate hyperparameter value λ_j on whether the resulting confidence intervals contains the true prediction accuracy or not. Therefore, we estimate the coverage probability conditional on the selected hyperparameter value, analogously to Equation (6.2), but with $\tilde{s}(d)$ instead of $s(d)$ and separately for BBC-CV and NCV.

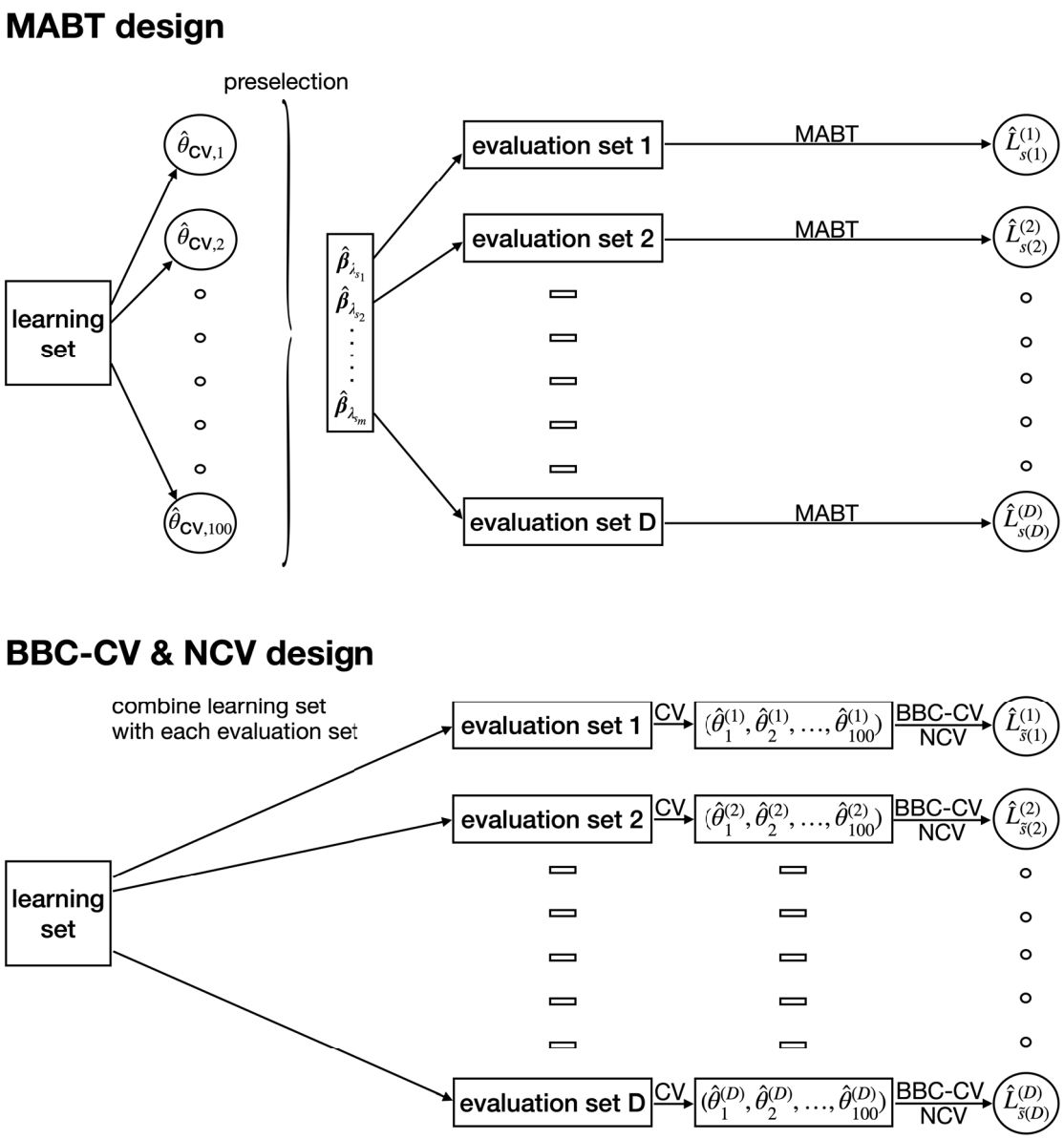

Figure 6.1: Designs for the simulation experiments to investigate the conditional coverage probabilities of MABT as well as BBC-CV and NCV intervals

Neither in the MABT nor BBC-CV and NCV simulation experiments, the candidate prediction models, or rather hyperparameter values in case of the cross-validation variants, are expected to be all selected equally often, and some candidates might even not be selected at all. Because it is our goal to compare coverage probabilities, we want to make sure that there are enough repeated measurements per candidate in order to get an estimate of the coverage probability with acceptable and comparable precision. Consequently, we only consider such candidates that are selected at least 1000 times among the $D = 100\,000$ repetitions. This ensures acceptable precision, and we reach comparable precision by computing the coverage probabilities from exactly 1000 randomly selected results each.

Distribution shift scenario A relevant scenario in real-world applications is that feature effects are amplified in the learning sample compared to the population and the evaluation sample. This distribution shift might happen, for instance, when the measurement of the features is conducted differently between the learning and the evaluation phase, or when the target distribution itself changes.

Hence, to emulate a distribution shift between the learning and the evaluation sets, we halve the signal strength in the evaluation set and in the population. Apart from that, the experiments remain unchanged compared to our previous considerations in this section, which we will refer to as the *regular scenario*; importantly, the (single) learning set remains the exactly the same.

We will present the results to the regular and the distribution shift scenario experiments in Section 6.3.2.

6.3 Results

In this section, we will present the results to the simulation experiments. For the sake of argumentation, we will revert to the original order in which we presented the objectives in Section 6.1.

Before we proceed, we need to establish a few general principles for handling the results, which will apply consistently across all experiments.

Coverage Because we only have finite numbers of simulation experiment replications, it is unlikely to exactly observe the desired confidence level. This is why we define an *acceptable coverage region*, that is, a deviation from the nominal confidence level $1-\alpha$ that we can explain by finite replications. In particular, because coverage probability is a binary proportion, we will allow the observed coverage

probability to vary within one standard error around the nominal level, that is,

$$\{\text{acceptable coverage region}\} = 1 - \alpha \pm \sqrt{\alpha(1-\alpha)/\{\text{no. of replications}\}}.$$

For example, when we repeat the experiment 1000 times, the acceptable coverage region is the interval $(94.31\%, 95.69\%)$, and for 5000 replications, it is $(94.69\%, 95.31\%)$.

Lower limits and tightness In order to avoid that we misleadingly favor confidence limits from anti-conservative methods, we only compare confidence intervals that actually contain the true prediction performance of the respective model. This also applies when we compare the tightness of limits.

In case the MABT interval contains the true value, but the comparator does not, the former is always considered superior than the latter. Similar applies when the interval from the comparative method contains the interval, but the MABT interval does not.

Also, notice that larger lower limits are associated with better prediction models and are, thus, better. In contrast, smaller values for the tightness of a lower limit means that it is more informative, as it gives a better idea about the true performance, and thus, are better than larger values.

True performance As the true performances only differ between the two competing selection-evaluation pipelines and not between the interval methods themselves, we only need to compare them by pipeline. Higher values are, of course, better here.

6.3.1 Two variants of multiplicity correction

In this section, we will investigate whether we should use the empirical or the standard-normal distribution within the proposed multiplicity correction. Both methods need almost the same computation time, so there is no advantage for the normal-variant in that regard.

Overall comparison For the overall comparison, we aggregate the results from all 5000 generated data sets and estimate the coverage probabilities of the two variants as the respective proportion of covering intervals. When we use the empirical distribution within the multiplicity correction, we observe a coverage probability

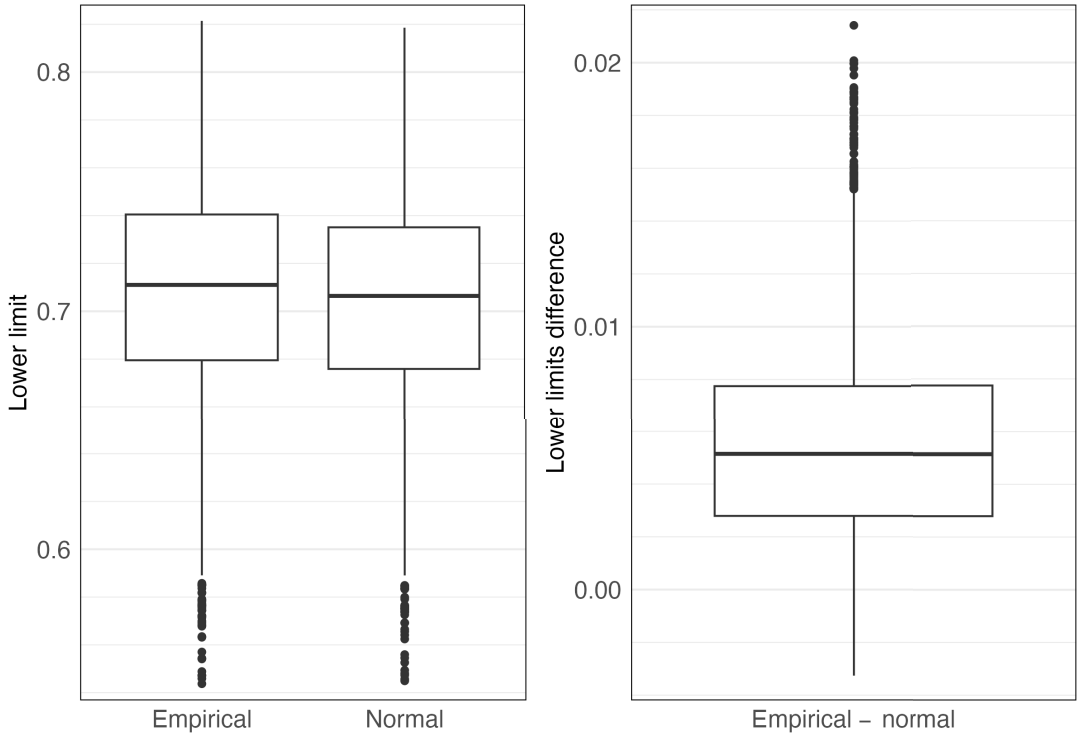

Figure 6.2: Lower confidence limits (left-hand pane) from MABT using the empirical and standard-normal distribution function, respectively, and per-data set differences (right-hand pane)

of 95.24 percent compared to 97.22 percent when we use the standard-normal distribution, respectively. Thus, the former variant yields fairly accurate coverage and is, in particular, less conservative than the standard-normal distribution variant.

However, this does not necessarily mean that the more conservative bootstrap interval method yields less informative confidence intervals, as pointed out by Hall (1988). In the left-hand pane in Figure 6.2, we present the lower limits from the two variants. The boxplots are created over all 5000 simulated data sets. The corresponding summary table is in Table C.4 in Appendix C.4.

There are only small gains of the empirical distribution variant over the standard-normal variant regarding the size of the lower limit. In particular, we do not spot any overly negative effects of the more conservative limits from the standard-normal variant. However, this may be true here, but does not necessarily need to transfer to other settings.

Per-data set comparison We will continue with the per-data set comparison of the lower confidence limits from the two variants. We computed the difference between the lower limit obtained from the empirical distribution-variant and the normal-variant. Hence, a positive difference is in favor of the empirical distribution variant. We only consider those results here where both intervals contain the true performance, which happens in 4762 out of 5000 simulated data sets.

We plot these differences in the right-hand pane in Figure 6.2, and the corresponding summary table is in Table C.4 in Appendix C.4. The empirical distribution-variant limits are at least as large as the normal-variant limits in 98.2 percent of the simulated data sets, and strictly larger in 93.2 percent.

This suggests that using the empirical distribution within the multiplicity correction is superior to the normal distribution transformation as in the vast majority of data sets we obtain larger lower limits, which seems reasonable because the empirical distribution variant does not rely on the approximate standard-normal distribution. Consequently, we will continue to use the empirical distribution variant for all subsequent considerations and experiments.

6.3.2 Conditional performance

In this section, we will compare how well BBC-CV, MABT, and NCV track the conditional performance in both the regular and distribution shift scenarios.

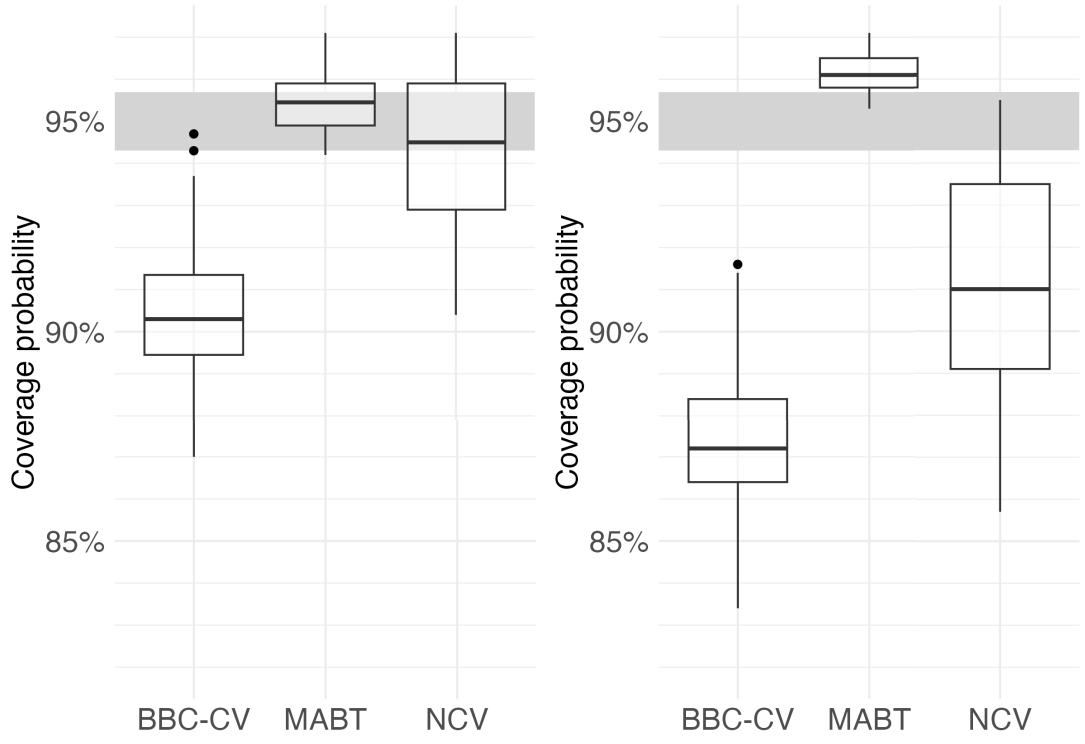

Figure 6.3: Estimated conditional coverage probabilities from the regular and distribution shift simulation experiments on the left-hand and the right-hand side, respectively

Regular scenario The total computation time for the experiments where the samples in the learning and the evaluation sets come from the same distribution is about three days on an AMD Ryzen Threadripper PRO 5975WX CPU using 60 threads. In the MABT experiments, 26 of the 100 candidate LASSO prediction models get preselected, 24 of which in at least 1000 evaluation sets.

Because for BBC-CV and NCV the selection is due to the average performance from a ten-fold cross-validation, both variants end up with the same selected hyperparameter value per evaluation set. Over the course of the $D = 100\,000$ evaluation sets, a total of 83 different values for the hyperparameter get selected, 59 of which in at least 1000 evaluation sets.

In the left-hand pane in Figure 6.3, we present summarized results to the three interval procedures. The corresponding summary table is the upper one in Table C.5 in Appendix C.5. In addition, Tables C.6 and C.9 as well as Figures C.1 and C.2 can also be found there, where we list the estimated conditional coverage probabilities and plot them against the associated hyperparameter values, respectively.

Distribution shift scenario The distribution shift scenario runs about three days, as well, on the same CPU with the same number of threads. Since the learning set is the same as in the regular experiments, for the MABT experiments, the same 26 LASSO prediction models get preselected for evaluation, all of them in at least 1000 evaluation sets.

For the experiments that concern BBC-CV and NCV, 85 candidate hyperparameter values are selected, and 57 of which in at least 1000 evaluation sets.

In the right-hand pane in Figure 6.3 we present summarized results to the three interval procedures. The corresponding summary table is the lower one in Table C.5 in Appendix C.5. In addition, Tables C.8 and C.9 as well as Figures C.1 and C.2 can also be found there, where we list the estimated conditional coverage probabilities and plot them against the associated hyperparameter values, respectively.

Conclusion We conclude that, in our experiments, neither BBC-CV nor NCV yield confidence intervals for the prediction accuracy of a LASSO binary classifier that maintain the desired conditional coverage level; MABT, in contrast, does.

In addition, the first two methods are only applicable when cross-validation itself is employed. This considerably limits the range of potential applications. For

instance, when either is utilized alongside complex neural nets, the total computation time likely gets excessively long. This is not a huge problem with BBC-CV, but with NCV even more, due to the many additional model trainings and the need to run the NCV scheme repeatedly, with different allocations of the observations to the folds, in order to obtain stable results. In our experiments, NCV is repeated 50 times with ten folds each, which results in 5000 additional model fits per confidence interval.

MABT, however, does not require cross-validation at all; it is merely a tool for stabilizing the preselection of promising models. Also, it directly and inherently estimates conditional lower limits, without any additional distributional assumptions or model training.

We note that the confidence intervals from BBC-CV and NCV are not fully conditional, though, that is, on both model parameter and hyperparameter, as this appears infeasible to us. However, they are conditional on the hyperparameter value, and we presume that further conditioning on model parameters would only worsen the situation.

Due to all of these reasons, we rule out both cross-validation variants as suitable comparators to MABT limits. Also, our findings in these experiments highlight the need for a universally valid procedure that yields reliable confidence intervals for conditional performance. MABT is such a procedure.

We acknowledge, however, that BBC-CV and NCV might potentially be useful alternatives when the primary interest is not to obtain a valid interval estimate for conditional performance.

6.3.3 Comparison to standard methods

In this section, we will present the results to the LASSO and random forest simulation experiments to compare MABT intervals to standard methods. As in Section 6.3.1, in addition to the overall comparison, we will compute the differences between the MABT and each of the competing lower limits on each simulated data set in order to conduct per-data set comparisons.

LASSO overall results

We will begin with the LASSO simulation experiments. To estimate the different confidence limits here takes about 70 minutes on an AMD Ryzen Threadripper 3960X CPU using 44 threads. Note, however, that this does not include model training and selection. We will begin with an overall comparison across all 5000 simulated data sets. After that, we will compare the results on a per-data set basis.

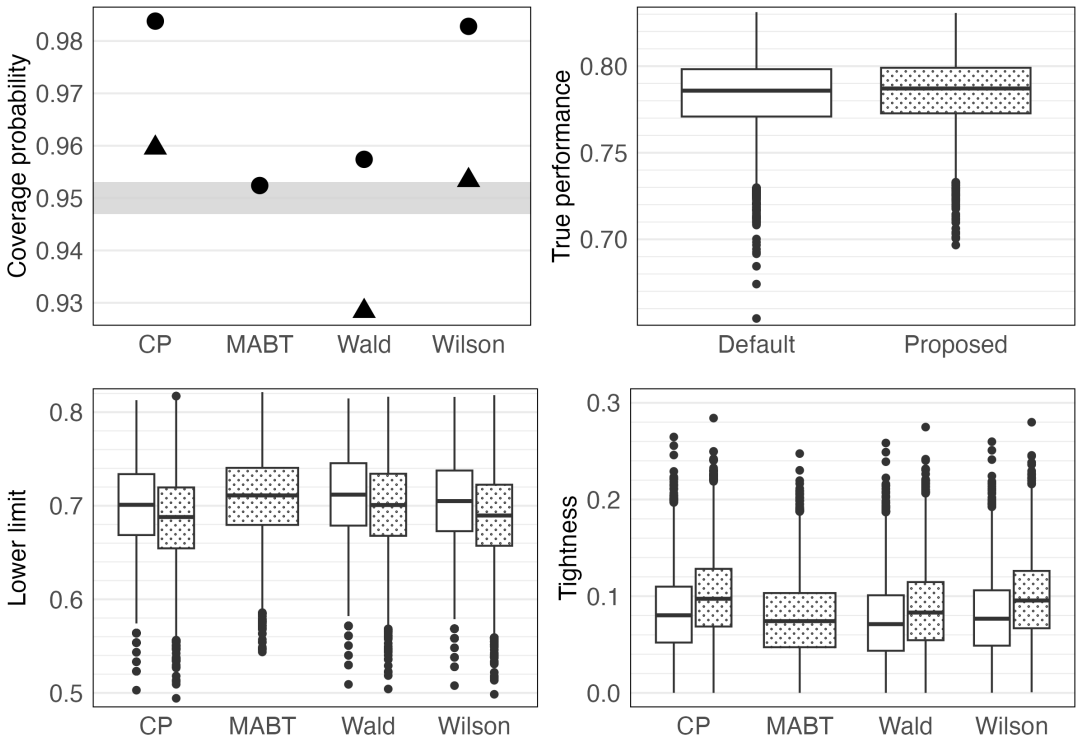

Figure 6.4: Results to the LASSO simulation experiments. The triangles and circles in the upper right plot as well as the plain and dotted patterns in the remaining boxplots represent the default and proposed pipeline, respectively. The shaded area represents the *acceptable coverage region*

Coverage probability The upper left plot in Figure 6.4 shows the observed coverage probabilities of the five competing interval methods, obtained from the default and proposed selection-evaluation pipeline, respectively. As we fix the global significance level at five percent, the desired coverage probability is 95 percent.

In the default pipeline, the CP, Wald, and Wilson limits yield observed coverage probabilities of 95.96, 92.84, and 95.34 percent. Hence, while the Wilson limits are only slightly conservative, the CP limits are conservative, and the Wald limits are anti-conservative.

In the proposed pipeline, the respective Šidák-corrected versions are moderately to highly conservative at 98.38, 95.74, and 98.28 percent. The proposed MABT confidence limits, though, turn out fairly accurate at 95.24 percent coverage probability.

Lower confidence limit The boxplots in the lower left pane in Figure 6.4 illustrate the lower confidence limits of the different interval methods for the different selection-evaluation pipelines and evaluation sample sizes. The associated summary table is in Table C.10 in Appendix C.6.

In the default pipeline, the Wald limits are the largest, but they fall below the desired coverage probability of 95 percent and, hence, make an unfair comparator. Among the remaining three interval methods, the CP and the Wilson limits are somewhat smaller than the Wald limits.

In the proposed pipeline, the Šidák-adjusted limits are all smaller than their unadjusted counterparts. The MABT limits turn out largest among the proposed-pipeline competition, and comparably large as the anti-conservative Wald limits from the default pipeline.

Tightness In the lower right plot in Figure 6.4, we compare the tightness of the different confidence limits, that is, the distance between the true prediction performance of the respective model and the lower limit. The associated summary table is in Table C.10, which can be found in Appendix C.6.

Our findings regarding the sizes of the lower limits can, in principle, be directly transferred. In the default selection-evaluation pipeline, the Wald limits are tightest, but the method is anti-conservative. The limits from the CP and the Wilson method are somewhat less tight.

Among the limits from the proposed pipeline, the ones from MABT are clearly the tightest, and almost as tight as the default-pipeline Wald limits.

	Pipeline	
Method	Default	Proposed
CP	67.0%	100.0%
Wald	34.0%	89.0%
Wilson	54.7%	100.0%

Table 6.1: Proportions of covering confidence intervals from MABT that are larger than the comparator in the LASSO simulation experiments

True performance Lastly, we will compare the true performance of the models selected in the default and the proposed selection-evaluation pipeline, respectively. The boxplots in the upper right pane in Figure 6.4 present the results, and the corresponding summary table is Table C.11 in Appendix C.6. We see that the gains regarding the true predictive performance in the proposed over the default pipeline are small.

LASSO per-data set results

In addition to the overall comparisons, we will compare the limits on a per-data set basis, as the confidence limits come in matched sets.

Lower confidence limit In Table 6.1, we report the proportion of covering intervals in which the limit from MABT is larger than the comparator. In total, 4762 out of 5000 confidence intervals from MABT contained the true performance.

We observe that, overall, in the majority of cases, the limits from the proposed MABT are larger than the comparators. In this regard, MABT loses only to the Wald limits from the default selection-evaluation pipeline, but they are anti-conservative, so this remains an unfair comparison.

True performance When we compare the true predictive performances of the model selected from the proposed selection-evaluation pipeline to the one from the default pipeline, we observe that in 70.48 percent of the 5000 simulated data sets the final model is at least as good, and in 42.66 percent it is strictly better.

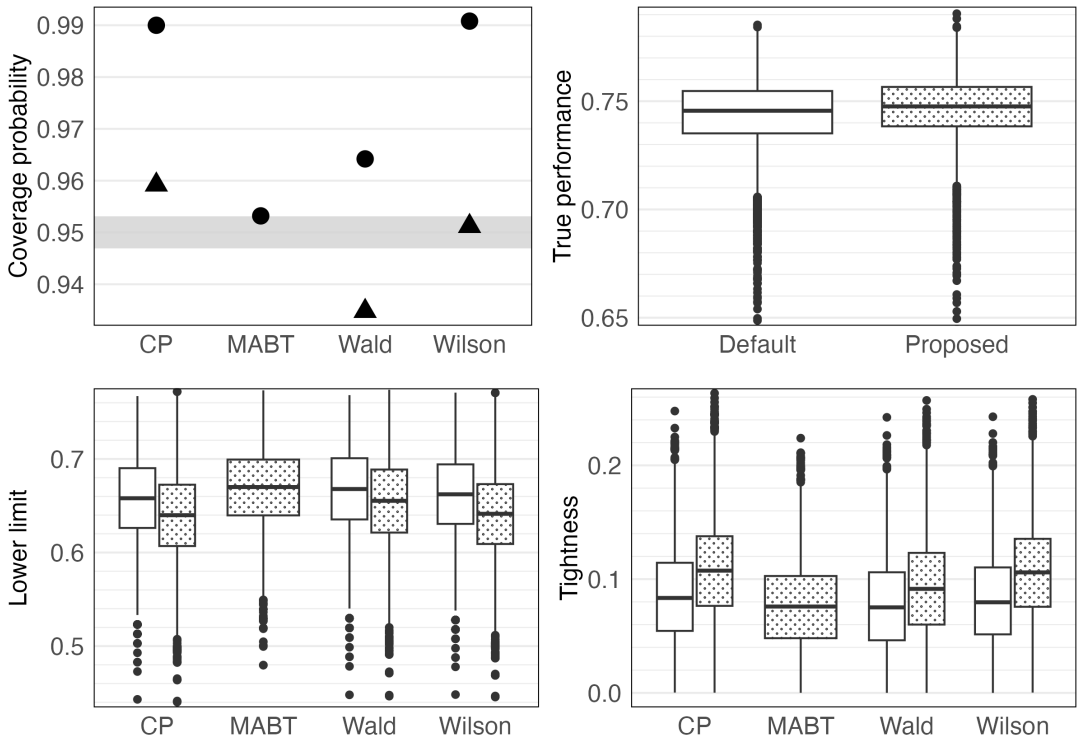

Figure 6.5: Results to the random forest simulation experiments. The triangles and circles in the upper right plot as well as the plain and dotted boxplots represent the default and proposed pipeline, respectively. The shaded area represents the *acceptable coverage region*

Random forest overall results

Next, we turn to the random forest simulation experiments. The estimation of the confidence limits takes about 43 hours on an AMD Ryzen Threadripper PRO 5975WX CPU using 60 threads. Note that this does neither include data generation nor model fitting or selection. Our findings on the random forest experiments are largely consistent with our findings in the LASSO experiments.

Coverage probability The plot in the upper left pane in Figure 6.5 presents the observed coverage probabilities of the competing interval methods for the different selection-evaluation pipelines. Again, we fix the global significance level at five percent, and thus, the desired coverage probability is 95 percent.

In the default pipeline, the CP, Wald, and Wilson limits yield observed coverage probabilities of 95.92, 93.48, and 95.12 percent. Therefore, the CP limits are conservative and the Wald limits are anti-conservative, while the Wilson limits yield very accurate coverage probability.

In the proposed-selection-evaluation pipeline, the respective Šidák-corrected

variants are all conservative at 99.00, 96.42, and 99.08 percent coverage. The proposed MABT limits are very slightly conservative with a coverage probability of 95.32 percent.

Lower confidence limit In Figure 6.5, the boxplots in the lower left pane present the lower confidence limits of the different interval methods for the various selection-evaluation pipelines. The associated summary table is in Table C.12 and placed in Appendix C.7.

In the default pipeline, the Wald limits are the largest, but they are anti-conservative, falling below the desired coverage probability of 95 percent. Thus, again, they make an unfair comparator. Among the two remaining interval methods, the Wilson limits are slightly larger than the CP limits.

In the proposed pipeline, the MABT limits are considerably larger than all of the Šidák-corrected intervals, and they are even larger than all of the limits from the default pipeline, even though they are more conservative than both the Wald and the Wilson limits.

Tightness The observations regarding the size of the lower limits can largely be transferred to the tightness of the limits, which are presented in the lower right pane in Figure 6.5. The corresponding summary table is in Table C.12 in Appendix C.7.

In the default pipeline, the Wald limits are tightest, but the method itself anti-conservative. The Wilson limits are somewhat less tight than the Wald limits, but a bit tighter than the CP limits.

In the proposed pipeline, the MABT limits are evidently the tightest, and comparably as tight as the anti-conservative Wald limits from the default pipeline.

True performance In Figure 6.5, in the upper right pane we present the true predictive performances of the model selected by the default and proposed selection-evaluation pipeline, respectively. The associated summary table is Table C.13 in Appendix C.7. The gains of the proposed over the default pipeline are rather small.

Random forest per-data set results

After we compared the results to the random forest experiments across all 5000 simulated data sets, we will now draw per-data set comparisons.

	Pipeline	
Method	Default	Proposed
CP	69.2%	100.0%
Wald	48.5%	98.1%
Wilson	59.6%	100.0%

Table 6.2: Proportions of covering confidence intervals from MABT that are larger than the comparator in the random forest simulation experiments

Lower confidence limit Table 6.2 shows the proportions of the simulated data sets in which the MABT limit turns out at least as large as each of the comparative limits. In total, 4766 out of 5000 confidence intervals from MABT covered the true performance, respectively.

In the majority of cases, the limits from the proposed MABT are larger than the comparators. Only in comparison to the anti-conservative Wald limits from the default selection-evaluation pipeline, the MABT limits are mostly smaller, but this is an unfair comparison.

True performance Lastly, when we compare the true predictive performances of the models selected from the proposed to the default pipeline, we observe that in 63.8 percent of the 5000 simulated data sets, the final model is at least as good, and in 59.0 percent it is strictly better.

Chapter 7

Applications to Real-World Data

In the previous chapters, we established MABT confidence limits as a viable alternative by both studying their theoretical asymptotic properties as well as their goodness in finite samples via simulation experiments. In this chapter, we will follow two objectives. The first is the application of the MABT confidence limits to classification tasks on real-world data, and the second is to test and compare them to standard methods.

While our first application to real-world data in Section 7.1 can be seen as in-between simulation experiments and real-world application, in Section 7.2 we will present an interesting issue when the default selection-evaluation pipeline is used. In particular, even when repeated cross-validation is used, the default pipeline might lead to the selection of prediction models with only subpar predictive performance.

The considerations in this chapter resemble the real-world applications in our publication Rink & Brannath (2025). The main difference between the publication and the present work is the use of test statistics instead here, in accordance with the theoretical results in Chapter 5.

The `R` code to these applications of MABT can be accessed via a public GitHub repository at `https://gitlab.informatik.uni-bremen.de/s_opbgf3/clfpp`.

7.1 OpenML benchmark

In this first application, we compare the MABT confidence limits to standard methods on a number of different real-world experiments from the OpenML platform (Vanschoren et al., 2013). OpenML is an open-source collaborative hub for machine learning research. It facilitates sharing, exploration, and analysis of machine learning data, tasks, and experiments. All included data sets are uniformly formatted, have detailed descriptions, and can be used with a variety of machine

Quantity	Min	Median	Max
Sample size	522	1563	45 312
Majority class size	278	1280	39 922
Minority class size	46	356	19 237
Class imbalance	1	1.92	16.27
Number of features	5	21	971
Observations per feature	3.6	575.0	5034.7

Table 7.1: Minimum, median, and maximum values among the included OpenML data sets for binary classification. The class imbalance is computed as the quotient of the majority class size by the minority class size. The number of observations per feature is obtained by dividing the sample size by the number of features

learning environments and libraries such as `R`, `Python`, or `Jupyter`. OpenML can be freely accessed via `http://openml.org`.

An OpenML experiment includes a real-world data set and a specific machine learning task such as binary classification. Users can upload their results as well as the detailed evaluation and algorithm pipelines they used.

For our purposes, we resemble Probst et al. (2018), who utilize OpenML to benchmark their `tuneRanger` function against other random forest tuning implementations, using the `OpenML100` benchmarking suite and the `OpenML R` package by Casalicchio et al. (2017).

As a first step, we exclude all OpenML binary classification tasks that either contain missing values in their respective data sets or are expected to take longer than 10 000 seconds to run. Table 7.1 summarizes the basic characteristics of the 33 included data sets such as sample size, number of features, and related quantities. A complete list with detailed information is placed in Table D.15 in Appendix D. In an initial test, we observe that in some of the selected data sets, the random forest classifier reaches near perfect prediction performance. We thus add a small amount of noise to the true class labels in order to make the classification problems more difficult.

We proceed in the same way as in our simulation experiments in Chapter 6. We split the data randomly into 75 percent for learning and 25 percent for evaluation, fit random forests to the learning set using the `tuneRanger R` function, and use the evaluation data to obtain lower confidence limits for the prediction accuracy. Note, however, that it will not be possible to answer the question whether the estimated confidence intervals cover the true performance or not, or questions derived from this information; we do not know the truth, as these are real-world

samples.

In the proposed selection-evaluation pipeline, as we do not employ cross-validation here, we cannot use the *within one standard-error* selection rule. Instead, we will use the *top ten percent* rule, as introduced in Section 6.2.1. In the final selection stage, we will select the best-performing model in the evaluation set and compute our proposed MABT confidence limits using 10 000 bootstrap samples. In case multiple models perform equally good, we prefer less complex models over more complex ones; that is, models that take less computation time during fitting.

For each of the considered OpenML experiments, we will perform learning, selection, and evaluation repeatedly with ten different allocations of the data to learning and evaluation. We do this in order to account for possibly disadvantageous allocation situations. This allows a fairer assessment of the performance of the competing interval approaches.

Throughout, as the results come in matched sets, we will draw per-data set comparisons on the basis of the lower limits averaged over the ten different allocations of the data to learning and evaluation.

As mentioned before, we will consider both binary and multi-class classification experiments. The comparative methods are the CP, Wald, and Wilson limits, all obtained from the default selection-evaluation pipeline. The repeated experiments run about seven hours on an AMD Ryzen Threadripper 3960X CPU using 44 threads.

Figure 7.1 illustrates the results from the binary OpenML classification experiments. For each data set and each interval method, we average the confidence limits over the ten repeated runs.

In the left-hand pane of Figure 7.1, we present the absolute differences between the limit obtained from MABT and the respective comparator, and in the center pane we present the relative gains. In the right-hand pane, we show what percentage of the maximum possible improvement we achieve by using MABT over the comparator, that is,

$$\{\%\text{ of max. possible improvement}\} = \frac{\{\text{MABT limit}\} - \{\text{comparator limit}\}}{1 - \{\text{comparator limit}\}} \cdot 100\%.$$

In all these cases, positive numbers favor MABT. The associated summary tables are placed in Table D.14 in Appendix D.

In addition to the quantified gains, we observe that the average lower limit computed from MABT in the proposed selection-evaluation pipeline is larger than the CP, Wald, and Wilson average lower limit from the default pipeline in 31, 17, and 30 out of the 33 data sets, respectively. Note, however, that the Wald

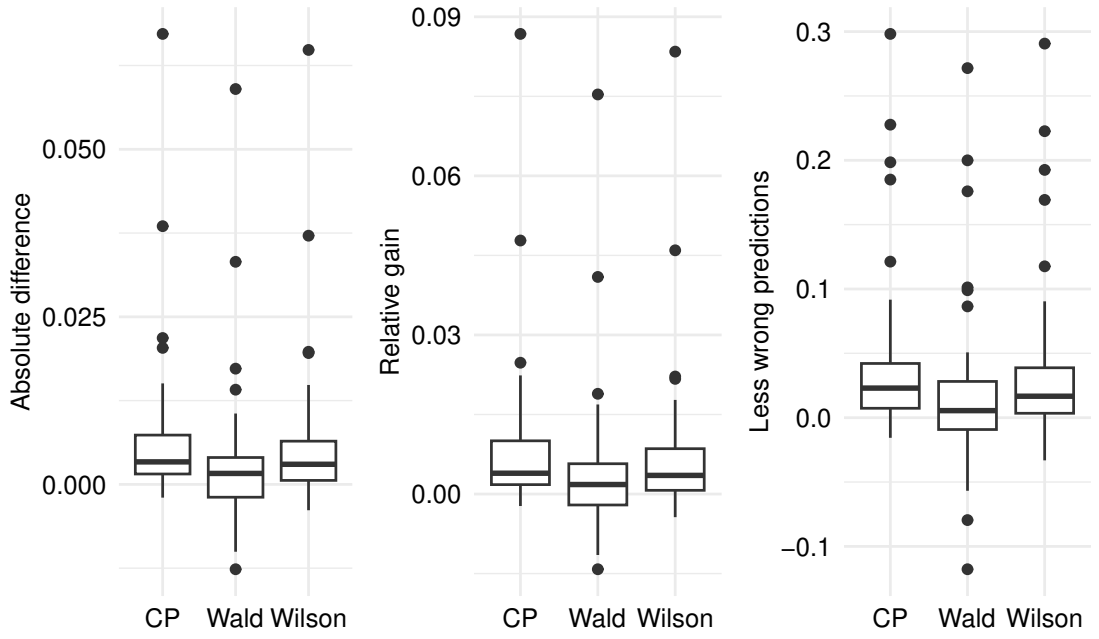

Figure 7.1: Gains of the MABT limits in the OpenML benchmark. From left to right, we present the absolute differences, the relative gains, and the percentage of the maximum possible improvement over the comparators

limits turn out anti-conservative in our simulation experiments in Chapter 6, so this comparison may be unfair.

The gains are mostly small. However, strikingly, there are some instances in which the average lower limit from MABT is considerably larger than the comparators. Overall, the MABT limits appear very competitive.

7.2 Cardiotocography data

In this example we apply the proposed selection-evaluation pipeline and the MABT confidence limits to the Cardiotocography data set provided by Ayres-de Campos & Bernardes (2010), which can be downloaded from the University of California, Irvine Machine Learning Repository Dua & Graff (2017) at `https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(diagnostic)`.

Westphal & Brannath (2020) used this data set to illustrate that the evaluation of multiple methods may be beneficial as to final model performance, and we resemble their setup in order to answer the question whether the potential gains of the proposed method outweigh the losses suffered through the necessary multiplicity adjustment, especially regarding the lower confidence limits.

The Cardiotocography data set contains medical data on unborn children dur-

ing their mothers' pregnancies. Cardiotocography is a common procedure used to assess fetal well-being by measuring heart rate patterns, uterine contractions, and other physiological parameters. The data set comprises 2126 complete cardiotocograms, which were inspected for anomalies by three expert obstetricians. Depending on the degree of the observed anomaly, the experts assigned a consensus label to each cardiotocogram. More details on the data set can be found in Ayres-de Campos et al. (2000).

For our purposes, we dichotomize the ordinal label into two classes, *suspect abnormal state* and *normal state*. There are 471 observations in the former and 1655 observations in the latter class. Our goal is to predict the suspect abnormal state from 23 features of fetal heart rate and report a lower confidence limit for the performance of the prediction model.

For each cardiotocogram the date of measurement is recorded. We will use this information to split the data set into a learning and an evaluation set that contain the first 75 and the last 25 percent of the data according to the date, respectively. Other than that, the date variable will not be used; in particular, it will not be used for model training.

During model learning, we fit 100 models from the elastic net class, see Section 2.3.1. Recall that the elastic net has two hyperparameters γ and λ, which control the balance between the ℓ_1 and ℓ_2 penalty, and the overall strength of regularization, respectively. For γ, we specify five equidistant values between zero and one, and 20 equidistant values between zero and $\lambda_{\max}(\gamma)$ for λ.

In order to obtain the validation performance for each of the candidate models, we will perform ten-fold cross-validation and will compare results from single-run and ten-times repeated cross-validation. To preselect models for evaluation, we will use the *within one standard-error* selection rule in the proposed selection-evaluation pipeline. In the default pipeline, we will select the model with the best performance in the cross-validation.

After refitting the preselected models on the entire learning set, we predict the classes in the evaluation set. Then, in the default pipeline, we estimate a lower confidence limit using the CP, Wald, and Wilson method at the full significance level of five percent.

In the proposed pipeline, we perform the final model selection, select the model with the best performance in the evaluation set among the preselected models, and estimate Šidák-corrected versions of the CP, Wald, and Wilson limits as well as the proposed MABT limit, using 10 000 bootstrap samples.

For each of the following three scenarios, the computation time is about eleven seconds on a single core of an M2 Apple MacBook Air.

Table 7.2 shows the prediction accuracies and ranks of the preselected models in

the validation and evaluation phase. Across all models, there is a substantial performance decrease when going from validation to evaluation. The best-performing model in the validation has a prediction accuracy 94.54 percent and turns out the worst-performing in the evaluation with only 72.93 percent accuracy. At the same time, the second-worst model in the validation has accuracy 93.04 percent and performs best in the evaluation with 83.46 percent accuracy.

Hence, in the default selection-evaluation pipeline, the user discovers a relatively poor model, while in the proposed pipeline, the user is protected from this subpar selection.

Validation		Evaluation	
Prediction accuracy	Rank	Prediction accuracy	Rank
93.10%	7	83.46%	1
93.29%	4	74.06%	5
93.29%	5	79.51%	2
93.16%	6	73.87%	6
93.48%	2	78.57%	3
93.04%	8	73.87%	7
93.35%	3	78.57%	4
93.54%	1	72.93%	9
93.04%	9	73.68%	8

Table 7.2: Prediction accuracies and ranks of the prediction models in the Cardiotocography example when only a single ten-fold cross-validation is performed. When two models have the same prediction accuracy, the less complex one gets ranked higher

Of course, this reflects in the confidence limits. In Table 7.3 we present the estimated lower limits for both, the final selection from the default and the proposed pipeline. The subpar selection from the default pipeline leads to lower limits for the prediction performance of about 70 percent, while the limits associated with the proposed pipeline are at about 79 percent. Amongst those, the proposed MABT limit turns out largest at 79.69 percent.

Admittedly, the presented situation may be a very unfortunate one; a different allocation of the learning data set to the ten cross-validation folds might mitigate the issue.

Indeed, when we use a ten-times repeated cross-validation scheme, that is,

Pipeline	CP	MABT	Wald	Wilson
Default	69.58%	–	69.76%	69.65%
Proposed	79.00%	79.69%	79.38%	78.99%

Table 7.3: Lower confidence limits for the prediction accuracy in the Cardiotocography example when only a single ten-fold cross-validation is performed

repeatedly allocate the learning data to the ten folds and average the performances, the evaluation estimates turn out much closer, as shown in Table 7.4.

In comparison to the single-run cross-validation, the performance drop between validation and evaluation persists, but the selection appears much more stable; the best-performing model in the validation is the second-best in the evaluation, and the margin to the best is relatively small.

Nevertheless, we still observe some rather unexpected differences between the validation and evaluation ranks such as the second-best model in the validation, which turns out worst in the evaluation with a clearly inferior evaluation performance.

Validation		Evaluation	
Prediction accuracy	Rank	Prediction accuracy	Rank
93.18%	5	74.06%	4
93.24%	3	79.51%	1
93.07%	7	73.87%	5
93.46%	1	78.57%	2
93.12%	6	73.87%	6
93.23%	4	78.57%	3
93.39%	2	72.93%	8
93.07%	8	73.68%	7

Table 7.4: Prediction accuracies and ranks of the prediction models in the Cardiotocography example when a ten-times repeated ten-fold cross-validation is performed. When two models have the same prediction accuracy, the less complex one gets ranked higher

Also, we observe that now there are only minor differences in the confidence limits in both, the default and proposed selection-evaluation pipeline, and all the methods yield a similar lower limit for the prediction accuracy of the respective

selected model at about 75 percent.

	CP	MABT	Wald	Wilson
Default pipeline	75.44%	–	75.65%	75.50%
Proposed pipeline	74.82%	75.67%	75.15%	74.83%

Table 7.5: Lower confidence limits for the prediction accuracy in the Cardiotocography example when a ten-times repeated ten-fold cross-validation is performed

Now the question arises as to whether repeated cross-validation is the appropriate tool to prevent situations as presented in Table 7.2 from happening. It turns out that this is not always the case.

Tables 7.6 and 7.7 present the results for another run of ten-times repeated ten-folds cross-validation. The performance drop between validation and evaluation remains. In contrast to the previous repeated cross-validation, here we cannot observe a stabilizing effect on the model selection at all. Rather, we see the same phenomenon as with the initial single-run cross-validation; the best-performing model in the validation has a prediction performance of 93.49 percent and is the second-worst in the evaluation, where it only scores 72.93 percent accuracy, and the margin to the best-performing is rather large with a difference of 7.48 points.

Also, the subpar selection in the default pipeline leads to lower confidence limits for the prediction performance of about 70 percent, whereas the limits from the proposed pipeline are at about 75 percent, and among those, the MABT limit is largest at 75.79 percent.

In such cases, the absolute gain of five points translates to a relative gain of seven percent, and MABT achieves $(0.75 - 0.70)/(1 - 0.70) = 17\%$ of the maximum achievable improvement over the limits from the default pipeline, which is substantial.

Because, still, these are only select cases, we additionally run the selection-evaluation scheme 100 times, which takes about four minutes on an AMD Ryzen Threadripper 3960 CPU using 40 threads. In each instance, we use ten-times repeated cross-validation with different allocations of the learning data to the cross-validation folds, so we can investigate the average behavior of the different interval methods in the two pipelines in this data set.

We observe that in 91 instances the lower confidence limit from MABT is larger than the Wald limit, and in all 100 instances it is larger than both the CP and the Wilson limit, independent from the pipeline. Also, in ten instances the user

Validation		Evaluation	
Prediction accuracy	Rank	Prediction accuracy	Rank
93.12%	6	74.06%	4
93.26%	3	79.51%	1
93.17%	5	73.87%	5
93.46%	2	78.57%	2
93.09%	8	73.87%	6
93.21%	4	78.57%	3
93.49%	1	72.93%	8
93.10%	7	73.68%	7

Table 7.6: Prediction accuracies and ranks of the prediction models in the Cardiotocography example when another instance of ten-times repeated ten-fold cross-validation is performed. When two models have the same prediction accuracy, the less complex one gets ranked higher

	CP	MABT	Wald	Wilson
Default pipeline	69.58%	–	69.77%	69.65%
Proposed pipeline	74.82%	75.79%	75.15%	74.83%

Table 7.7: Lower confidence limits for the prediction accuracy in the Cardiotocography example when another instance of ten-times repeated ten-fold cross-validation is performed

would have reported an unnecessarily small lower confidence limit that is about six points lower than the MABT limit, if the Wald limit was used instead.

Overall, MABT produced the largest lower confidence limits compared to both, the default and the proposed selection-evaluation pipeline, but the margin was rather small. Perhaps more importantly, our proposed approach protected the user from a subpar model selection, and thus, from reporting an unnecessarily small limit. This way, MABT addresses an apparent problem: The information whether the allocation of the learning data to the cross-validation folds will lead to subpar selections remains hidden to the user. In addition, the results show that there is a gain in using MABT for confidence interval estimation beyond the gains of repeated cross-validation and the selection of a better model due to the proposed pipeline.

Chapter 8

Discussion

In this final chapter, we will discuss the key findings on our proposed MABT confidence limits. We will begin the discussion in Section 8.1 by reviewing the results from the simulation experiments and the applications to real-world data. In Section 8.2, we will comment on the rules we used for model selection. Next, we will discuss the consequences of our findings in the presence of distribution shifts in Section 8.3. Ultimately, in Section 8.4, we will summarize the key properties of the proposed MABT confidence limits and conclude the discussion.

8.1 Simulation results

In both the LASSO and random forest simulation experiments, we found that the MABT confidence intervals were highly reliable regarding their coverage probability. In contrast, Wald intervals were anti-conservative, while Clopper-Pearson intervals were conservative. The Wilson intervals in the default pipeline proved to be similarly reliable as MABT intervals in the proposed pipeline. The Šidák-corrected versions of the Clopper-Pearson, Wald, and Wilson intervals were overly conservative.

Overall, across simulated data sets, MABT yielded noticeable but rather small improvements over the standard methods. We observed similarly small gains regarding the tightness of the lower limits and the true performance of the final selected models from the proposed selection-evaluation pipeline compared to the default pipeline.

In summary, when assessed across the respective simulated data sets, MABT intervals were the most reliable. In addition, the MABT lower limits turned out at least as large and similarly as tight as the best comparators, while the proposed pipeline yielded slightly better final models.

The differences between the intervals and pipelines became clearer, though,

when compared on a per-data set basis. Here, MABT yielded larger lower limits in the majority of cases, except when compared to the Wald limits, which were anti-conservative in our experiments, making this comparison unfair. In the random forest simulations, the per-data set gains of MABT over the comparators were even slightly larger than in the LASSO experiments, suggesting that MABT may offer more substantial improvements in more complex situations.

Furthermore, the gains in true model performance were also more apparent in the per-data set comparisons, where the final models from the proposed pipeline turned out strictly better in many, and at least as good in the majority of cases.

The results in the present work were consistent with the results in our publication Rink & Brannath (2025), where we directly employed the parameter of interest to estimate the MABT lower confidence limit. In the present work, though, we aligned the simulation experiments with our theoretical results by utilizing test statistics, and observe that this results in coverage probabilities that are even closer to the nominal level, in both the LASSO and random forest experiments. Additionally, the use of test statistics results in slightly larger lower limits.

In our publication, we also tested MABT confidence limits for AUC within the LASSO experiments, and the results were promising. However, at this stage, we do not have a theoretical justification for this application, which would require the use of so-called U-*statistics*, see, for example, Shao & Tu (1995). Extending the asymptotic theory to U-statistics is a potential line of future research.

8.2 Selection rules

The selection rules that we used in the simulation experiments and applications to real-world data can universally be applied to various validation strategies. In particular, the *top ten percent* selection rule can be employed in any model selection problem, and the *within one standard-error* rule can be used whenever cross-validation is used during learning, although the underlying idea of this rule may be adapted to different validation strategies.

At the same time, however, these rules are relatively generic and, therefore, maybe somewhat simplistic. More advanced selection rules that are tailored to the specific selection problem could allow for a more efficient use of the significance level, reducing the strength of multiplicity correction and, consequently, leading to larger lower limits. Moreover, such problem-specific rules might result in the final selection of models with better prediction performance.

A related issue arises when we need to compute simultaneous lower confidence limits for multiple prediction performance measures. For instance, it is a common task to find a model with both high sensitivity and high specificity. In such cases,

model selection becomes less straightforward because there are two quantities involved.

One possible option is to aggregate both measures into a single metric, such as balanced or weighted prediction accuracy. Typically, though, the sensitivity and specificity of a prediction model often differ considerably, and a model with very high sensitivity but only average specificity could still be selected as the final model. In this case, the max-T-type multiplicity correction in MABT may, on the one hand, yield a relatively informative and acceptable lower limit for the sensitivity, but, on the other hand, the lower limit for the specificity could be overly conservative.

This issue can occur whenever we aim to estimate a lower confidence limit for a specific performance measure, but use another for selection, as the selected model is likely not the optimal choice in terms of both performance measures.

8.3 Distribution shifts and data allocations

We mentioned before that a relevant distribution shift can occur when features are measured differently between the learning and the evaluation stages, or when the target distribution itself changes. Potential reasons include selection bias or overly strong relationships between the features and the labels in the learning data that do not manifest in the evaluation set. Especially in such scenarios model selection becomes highly sensitive to the allocation of the sample data to the learning and evaluation sets.

While cross-validation and repeated cross-validation are often utilized to mitigate this issue, they apparently do not always resolve it. In particular, in the distribution shift simulation experiments that we conducted to assess how well BBC-CV and NCV confidence intervals track the conditional performance, we saw a substantial drop in coverage probability for the two approaches, rendering the respective confidence intervals almost useless. This happened even though only ten percent of the data came from a different distribution than the learning data, where the relationship between the features and the labels was more pronounced.

We encountered another issue when we used the default selection-evaluation pipeline with real-world data, in particular in the Cardiotocography data set. Here, we illustrated the potential problems of selecting only one model for evaluation instead of multiple promising ones, and how the allocation of the learning data to the cross-validation folds can severely harm model selection. Such an unfavorable allocation can result in the selection of a subpar prediction model and, thus, to the reporting of a subpar lower confidence limit. The consequences of this can be severe, potentially leading to the premature abandonment of an otherwise

promising study.

This raises a fundamental question about whether we should select models based on their validation performance at all. In addition, the performance estimates we obtain from cross-validation are estimates of unconditional performance, while we actually aim for conditional performance. This distinction is particularly critical because conditional performance and the estimates obtained from cross-validation can be asymptotically uncorrelated, as Bates et al. (2024) showed; see the discussion around corollary 2. In other words, the cross-validation estimate may not be an indicator of a prediction model's conditional performance at all.

In contrast, by using MABT confidence intervals in the proposed selection-evaluation pipeline, shifting the final model selection to the evaluation phase, these issues are mitigated. This way, the dependence on the specific data allocation is reduced, which makes our proposed approach even more beneficial when data is scarce. Moreover, it ensures the selection of a prediction model based on a true estimate of conditional performance, and ultimately prevents the selection of a subpar prediction model.

8.4 Conclusion

The proposed MABT confidence limits for conditional performance are universally valid across various settings. Their favorable asymptotic properties hold true under Hall's smooth function model, which is sufficiently general to include a wide range of common measures. Additionally, MABT confidence limits remain valid regardless of the selection strategy employed, whether formal, such as stepwise selection or the use of information criteria; informal, such as visual inspection and other diagnostic methods; based on post-hoc considerations, such as retrospective expert judgement; or any combination of those. Furthermore, MABT does not require any additional model training, making it computationally undemanding. It only needs to resample from the models' predictions from the evaluation set. Therefore, the primary computational effort lies in calculating the test statistics across the bootstrap samples. Also, MABT is designed to work with any combination of prediction models within a single selection-evaluation task, whether they are linear or non-linear classifiers.

In conclusion, MABT has practically no drawbacks, as it yields lower confidence limits for conditional prediction performance that are often larger than those from comparators, while reliably maintaining the desired confidence level in finite samples, and demonstrating solid theoretical properties.

Bibliography

Ayres-de Campos, D. & Bernardes, J. (2010). Cardiotocography. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C51S4N.

Ayres-de Campos, D., Bernardes, J., Garrido, A., de Sà, J. M., & Pereira-Leite, L. (2000). Sisporto 2.0: a program for automated analysis of cardiotocograms. *The Journal of Maternal-Fetal Medicine*, 9(5), 311–318.

Bates, S., Hastie, T., & Tibshirani, R. (2024). Cross-validation: What does it estimate and how well does it do it? *Journal of the American Statistical Association*, 119(546), 1434–1445.

Berk, R., Brown, L., Buja, A., Zhang, K., & Zhao, L. (2013). Valid post-selection inference. *The Annals of Statistics*, 41(2), 802–837.

Breiman, L. (2001). Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author). *Statistical Science*, 16(3), 199 – 231.

Brown, L. D., Cai, T. T., & DasGupta, A. (2001). Interval Estimation for a Binomial Proportion. *Statistical Science*, 16(2), 101 – 133.

Carpenter, J. & Bithell, J. (2000). Bootstrap confidence intervals: when, which, what? A practical guide for medical statisticians. *Stat Med*, 19(9), 1141–1164.

Casalicchio, G., Bossek, J., Lang, M., Kirchhoff, D., Kerschke, P., Hofner, B., Seibold, H., Vanschoren, J., & Bischl, B. (2017). OpenML: An R package to connect to the machine learning platform OpenML. *Computational Statistics*, 32(3), 1–15.

Csiszár, I. (1975). I-Divergence Geometry of Probability Distributions and Minimization Problems. *The Annals of Probability*, 3(1), 146 – 158.

DasGupta, A. (2008). *Asymptotic Theory of Statistics and Probability*. Springer.

Davison, A. C. & Hinkley, D. V. (1997). *Bootstrap Methods and their Application*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

DiCiccio, T. J. & Romano, J. P. (1990). Nonparametric confidence limits by resampling methods and least favorable families. *International Statistical Review*, 58(1), 59–76.

Dickhaus, T. (2014). *Simultaneous Statistical Inference*. Springer.

Dickhaus, T. (2018). *Theory of nonparametric tests*. Springer.

Dmitrienko, A. & Hsu, J. C. (2014). *Multiple Testing in Clinical Trials*, chapter 44, (pp. 550–557). John Wiley & Sons, Ltd.

Dua, D. & Graff, C. (2017). UCI machine learning repository. `http://archive.ics.uci.edu/ml`.

Efron, B. (1981). Nonparametric standard errors and confidence intervals. *The Canadian Journal of Statistics*, 9(2), 139–158.

Efron, B. & Tibshirani, R. (1994). *An Introduction to the Bootstrap*. Chapman and Hall/CRC.

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

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Hall, P. (1988). Theoretical comparison of bootstrap confidence intervals. *The Annals of Statistics*, 16(3), 927–953.

Hall, P. (1992). *The Bootstrap and Edgeworth Expansion*. New York: Springer.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, 2nd edition.

Hesterberg, T. (1999). Bootstrap tilting confidence intervals and hypothesis tests. *Computing Science and Statistics*, 31, 389–393.

Hesterberg, T. (2014). Bootstrap. In *Methods and Applications of Statistics in Clinical Trials* chapter 6, (pp. 62–101). John Wiley & Sons, Ltd.

Hothorn, T., Bretz, F., & Westfall, P. (2008). Simultaneous inference in general parametric models. *Biometrical Journal*, 50(3), 346–363.

Japkowicz, N. & Shah, M. (2011). *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press.

Kuhn, M. (2008). Building predictive models in R using the caret package. *Journal of Statistical Software*, 28(5), 1–26.

Lehmann, E. L. & Romano, J. P. (2005). *Testing Statistical Hypotheses*. New York, NY: Springer.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

Owen, A. B. (2001). *Empirical Likelihood*. Chapman and Hall/CRC.

Probst, P., Wright, M., & Boulesteix, A.-L. (2018). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*.

Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. DOI: 10.48550/arxiv.1811.12808.

Rink, P. & Brannath, W. (2025). Post-selection confidence bounds for prediction performance. *Machine Learning*, 114(3), 82.

Schennach, S. M. (2007). Point estimation with exponentially tilted empirical likelihood. *The Annals of Statistics*, 35(2), 634 – 672.

Shalev-Shwartz, S. & Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.

Shao, J. & Tu, D. (1995). *The Jackknife and Bootstrap*. Springer.

Spokoiny, V. G. & Dickhaus, T. (2015). *Basics of Modern Mathematical Statistics*. Springer.

Tsamardinos, I., Greasidou, E., & Borboudakis, G. (2018). Bootstrapping the out-of-sample predictions for efficient and accurate cross-validation. *Machine Learning*, 107, 1895–1922.

van der Vaart, A. W. (1998). *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Vanschoren, J., van Rijn, J. N., Bischl, B., & Torgo, L. (2013). OpenML: Networked science in machine learning. *SIGKDD Explorations*, 15(2), 49–60.

Westfall, P. H. & Young, S. S. (1993). *Resampling-based multiple testing: Examples and methods for p-value adjustment*, volume 279. John Wiley & Sons.

Westphal, M. & Brannath, W. (2020). Evaluation of multiple prediction models: A novel view on model selection and performance assessment. *Statistical Methods in Medical Research*, 29(6), 1728–1745.

Appendices

A Appendix to Chapter 4

In this Appendix, which is based on Sections 2 and 3 in van der Vaart (1998) and DasGupta (2008), respectively, we will provide the main concepts and results on stochastic convergence to complement the theoretical considerations on MABT confidence intervals in Section 5.3.

Let $\boldsymbol{X} = (X_1, X_2, \ldots, X_m)$ denote a vector of random real-valued random variables, and let the map $\boldsymbol{x} \mapsto \mathbb{P}(\boldsymbol{X} \leq \boldsymbol{x})$ denote its distribution function, where $\leq$ is to be understood component-wise. Additionally, let $d(\boldsymbol{x}, \boldsymbol{x}')$ be a distance function on $\mathbb{R}^m$, for example, the Euclidian distance, which is given by

$$d(\boldsymbol{x}, \boldsymbol{x}') = \|\boldsymbol{x} - \boldsymbol{x}'\|_2 = \sqrt{\sum_{i=1}^{m} (x_i - x_i')^2},$$

and let $(\boldsymbol{X}_n)_n$ denote a sequence of random variables.

The sequence $(\boldsymbol{X}_n)_n$ is said to *converge in distribution to* $\boldsymbol{X}$ if $\mathbb{P}(\boldsymbol{X}_n \leq \boldsymbol{x}) \to \mathbb{P}(\boldsymbol{X} \leq \boldsymbol{x})$ as $n \to \infty$ for all points $\boldsymbol{x}$ at which the limit distribution function $\boldsymbol{x} \mapsto \mathbb{P}(\boldsymbol{X} \leq \boldsymbol{x})$ is continuous. An alternative name is *convergence in law*, such that we will denote convergence in distribution by $\boldsymbol{X}_n \xrightarrow{\mathcal{L}} \boldsymbol{X}$.

Another mode of stochastic convergence is convergence in probability. The sequence $(\boldsymbol{X}_n)_n$ is said to *converge in probability to* $\boldsymbol{X}$ if for all $\epsilon > 0$ it holds that $\mathbb{P}[d(\boldsymbol{X}_n, \boldsymbol{X}) > \epsilon] \to 0$ as $n \to \infty$, and we will denote it by $\boldsymbol{X}_n \xrightarrow{\mathbb{P}} \boldsymbol{X}$.

The last type of stochastic convergence that we will mention here is almost sure convergence. The sequence $\boldsymbol{X}_n$ is said to *converge almost surely to* $\boldsymbol{X}$ if $\mathbb{P}[\lim_{n\to\infty} d(\boldsymbol{X}_n, \boldsymbol{X}) = 0] = 1$, and we will denote almost sure convergence by $\boldsymbol{X}_n \xrightarrow{\text{a.s.}} \boldsymbol{X}$.

The next proposition is a simple, but very useful result. It states that the mode of convergence persists under continuous mappings.

Proposition A.1 (Continuous Mapping Theorem)**.** *Let $g\colon \mathbb{R}^m \to \mathbb{R}^{m'}$ be continuous at every point of a set C such that $\mathbb{P}(\boldsymbol{X} \in C) = 1$. Then, the following hold*

true.

1. *If* $\boldsymbol{X}_n \xrightarrow{\mathcal{L}} \boldsymbol{X}$*, then* $g(\boldsymbol{X}_n) \xrightarrow{\mathcal{L}} \boldsymbol{g}(\boldsymbol{X})$*;*
2. *If* $\boldsymbol{X}_n \xrightarrow{\mathbb{P}} \boldsymbol{X}$*, then* $g(\boldsymbol{X}_n) \xrightarrow{\mathbb{P}} \boldsymbol{g}(\boldsymbol{X})$*;*
3. *If* $\boldsymbol{X}_n \xrightarrow{a.s.} \boldsymbol{X}$*, then* $g(\boldsymbol{X}_n) \xrightarrow{a.s.} \boldsymbol{g}(\boldsymbol{X})$*.*

In the next proposition, we will record some of the relationships among the three modes of stochastic convergence.

Proposition A.2. *The relationships among the three modes of stochastic convergence include the following.*

1. *Almost sure convergence implies convergence in probability, that is, from* $\boldsymbol{X}_n \xrightarrow{a.s.} \boldsymbol{X}$ *it follows that* $\boldsymbol{X}_n \xrightarrow{\mathbb{P}} \boldsymbol{X}$*;*
2. *Convergence in probability implies convergence in distribution, that is, from* $\boldsymbol{X}_n \xrightarrow{\mathbb{P}} \boldsymbol{X}$ *it follows that* $\boldsymbol{X}_n \xrightarrow{\mathcal{L}} \boldsymbol{X}$*.*

Next, we will shortly depart from Section 2 in van der Vaart (1998) and, in addition to the previous probabilistic results, record a classical mode of convergence, that is, uniform convergence. Let $(f_n)_n$ denote a sequence of functions. $(f_n)_n$ is said to *converge uniformly* to a limiting function f if for all $\epsilon > 0$ there exists $N \in \mathbb{N}$ such that for all $n \geq N$ it holds that $|f_n(x) - f(x)| < \epsilon$ for every potential value of x.

We return to Section 2 in van der Vaart (1998) and record the Multivariate Central Limit Theorem, which is Theorem 2.18 in van der Vaart (1998). For $m = 1$, the Central Limit Theorem directly follows from it.

Proposition A.3 (Multivariate Central Limit Theorem)**.** *Let* $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ *be i. i. d. m-dimensional random vectors with mean* $\mathbb{E}(\boldsymbol{X}_i) = \boldsymbol{\mu}$ *and covariance matrix* $Cov(\boldsymbol{X}_i) = \boldsymbol{\Sigma}$*. Additionally, let* $\bar{\boldsymbol{X}}_n = n^{-1} \sum_{i=1}^{n} \boldsymbol{X}_i$ *denote the sample mean. Then, the random vector* $\sqrt{n}(\bar{\boldsymbol{X}}_n - \boldsymbol{\mu})$ *converges in distribution to the m-dimensional multivariate normal distribution with mean vector* $\mathbf{0}$ *and covariance matrix* $\boldsymbol{\Sigma}$*; that is,* $\sqrt{n}(\bar{\boldsymbol{X}}_n - \boldsymbol{\mu}) \xrightarrow{\mathcal{L}} N_m(\mathbf{0}, \boldsymbol{\Sigma})$*.*

Finally, we will present a version of the Multivariate Delta Theorem, which is Theorem 3.7 in DasGupta (2008).

Proposition A.4 (Multivariate Delta Theorem)**.** *Under the conditions from Proposition A.3,*

$$\sqrt{n}[f(\bar{\boldsymbol{X}}_n) - f(\boldsymbol{\mu})] \xrightarrow{\mathcal{L}} N_{m'}[\mathbf{0}, \boldsymbol{\Sigma}'],$$

provided $g \colon \mathbb{R}^m \to \mathbb{R}^{m'}$ *is a smooth function and* $\boldsymbol{\Sigma}' = \nabla f(\boldsymbol{\mu})^T \boldsymbol{\Sigma} \nabla f(\boldsymbol{\mu})$ *is positive definite, where* ∇ *denotes the gradient.*

B Appendix to Chapter 5

The following proof resembles an argument that Prof. Dr. Werner Brannath gave in his Statistics I lecture on November 1, 2022, at the University of Bremen. We use it to prove Proposition 5.3.2.

Proof of Proposition 5.3.2. Before we prove the proposition, we observe the following.

1. For all $\epsilon > 0$ we can find a finite grid on $\mathbb{R}$,

$$-\infty \equiv t_0 < t_1 < t_2 < \cdots < t_{\kappa-1} < t_\kappa \equiv \infty,$$

such that

$$\max_{\ell=1}^{\kappa-1}[\Psi(t_\ell-) - \Psi(t_{\ell-1})] \leq \epsilon,$$

where $\Psi(t-) \equiv \lim_{t' \to t^-} \Psi(t')$ is the limit from below. This is true due to the following.

Since $\lim_{t\to-\infty} \Psi(t) = 0$ and $\lim_{t\to\infty} \Psi(t) = 1$, we can always find support points t_1 and $t_{\kappa-1}$ such that both $\Psi(t_1) - \Psi(t_0) \leq \epsilon$ and $\Psi(t_\kappa) - \Psi(t_{\kappa-1}) \leq \epsilon$. This is obviously true if Ψ is continuous, and it also follows in general, as we will argue next.

Let $J(\epsilon) = \{t \in \mathbb{R} \mid \Delta\Psi(t) \equiv \Psi(t) - \Psi(t-) > \frac{\epsilon}{2}\}$ denote the set of jumps of Ψ of at least $\frac{\epsilon}{2}$. Then, it follows that $J(\epsilon)$ needs to be finite, and we can write

$$\Psi(t) = \Psi^{\text{cont}}(t) + \sum_{t' \in J'(\epsilon,t)} \Delta\Psi(t'),$$

where $J'(\epsilon, t) = J(\epsilon) \cap (-\infty, t]$, and $\Psi^{\text{cont}}(t)$ is a continuous version of Ψ by shifting the function downwards at the jumps in order to connect the curve. Then, we can define the grid using $\frac{\epsilon}{2}$, Ψ^{cont}, and $J(\epsilon)$.

2. From the first observation, it follows that

$$\sup_{t\in\mathbb{R}} |\Psi_n(t) - \Psi(t)| \leq \max_{\ell=1}^{\kappa-1} |\Psi_n(t_\ell-) - \Psi(t_\ell-)| + \max_{\ell=1}^{\kappa-1} |\Psi_n(t_\ell) - \Psi(t_\ell)| + \epsilon.$$

This is true, because for $t \in \mathbb{R}$ and $\ell \in \{1, 2, \ldots, \kappa-1\}$ such that $t \in [t_{\ell-1}, t_\ell]$, it holds that

$$\Psi_n(t) - \Psi(t) \leq \Psi_n(t_\ell-) - \Psi(t_\ell-) + \Psi(t_\ell-) - \Psi(t_{\ell-1}) \leq \Psi_n(t_\ell-) - \Psi(t_\ell-) + \epsilon,$$

due the definition of the grid. Similarly, we obtain

$$\Psi_n(t) - \Psi(t) \geq \Psi_n(t_{\ell-1}) - \Psi(t_\ell-) \geq \Psi_n(t_{\ell-1}) - \Psi(t_{\ell-1}) - \epsilon.$$

Consequently,

$$|\Psi_n(t) - \Psi(t)| \leq |\Psi_n(t_\ell-) - \Psi(t_\ell-)| + |\Psi_n(t_{\ell-1}) - \Psi(t_{\ell-1})| + \epsilon.$$

This holds true for every $t \in \mathbb{R}$, it follows that

$$\sup_{t\in\mathbb{R}} |\Psi_n(t)-\Psi(t)| \leq \max_{\ell=1}^{\kappa-1} |\Psi_n(t_\ell-)-\Psi(t_\ell-)| + \max_{\ell=1}^{\kappa-1} |\Psi_n(t_{\ell-1})-\Psi(t_{\ell-1})| + \epsilon.$$

We will now prove Proposition 5.3.2. Let $\epsilon > 0$ and let $t_0 < t_1 < t_2 < \cdots < t_{\kappa-1} < t_\kappa$ denote a grid on $\mathbb{R}$ as in observation (1) from above. Note that Ψ_n and Ψ have the properties of a cumulative distribution function. Let $\mathbb{P}_n$ and $\mathbb{P}$ denote appropriate probability measures.

From the Strong Law of Large Numbers, see, for example, van der Vaart (1998), we obtain

$$|\Psi_n(t_\ell-) - \Psi(t_\ell-)| = |\mathbb{P}_n\{(-\infty, t_\ell)\} - \mathbb{P}\{(-\infty, t_\ell)\}| \xrightarrow{\text{a.s.}} 0,$$

and similarly,

$$|\Psi_n(t_\ell) - \Psi(t_\ell)| = |\mathbb{P}_n\{(-\infty, t_\ell]\} - \mathbb{P}\{(-\infty, t_\ell]\}| \xrightarrow{\text{a.s.}} 0.$$

Consequently, due to observation (2) from above,

$$\max_{\ell=1}^{\kappa-1} |\Psi_n(t_\ell-) - \Psi(t_\ell-)| + \max_{\ell=1}^{\kappa-1} |\Psi_n(t_\ell-) - \Psi(t_\ell)| \xrightarrow{\text{a.s.}} 0,$$

and because this holds true for all $\epsilon > 0$, we conclude that

$$\sup_{t\in\mathbb{R}} |\Psi_n(t) - \Psi(t)| \xrightarrow{\text{a.s.}} 0.$$

□

C Appendix to Chapter 6

C.1 MABT algorithm

Algorithm C.1 MABT lower confidence limit for prediction accuracy

Input: $\hat{y}_{ij}, y_i, \alpha$

Output: $\hat{L}_s$

$\tilde{y}_{ij} \leftarrow \hat{y}_{ij} == y_i$ ▷ check whether prediction equals true label (1) or not (0)
$\hat{\theta}_j \leftarrow mean([\tilde{y}_{ij}]_{i=1,2,\ldots,n})$
$s \leftarrow \text{argmax}\{\hat{\theta}_j \mid j = 1, 2, \ldots, m\}$
$T^*_{jb} \leftarrow bootstrap([\tilde{y}_{ij}]_{i=1,2,\ldots,m})$ ▷ multivariate bootstrap along i
$M_{ib} \leftarrow boot.freq(\tilde{y}_{ij}, b)$
for $j = 1, 2, \ldots, m$ **do**
    $\hat{H}^*_j \leftarrow ecdf(T^*_{j1}, T^*_{j2}, \ldots, T^*_{jB})$
    $\hat{u}^*_{jb} \leftarrow \hat{[H]}^*_j(T^*_{jb})$
end for
$\hat{u}^*_{\max,b} \leftarrow \max\{\hat{u}^*_{jb} \mid j = 1, 2, \ldots, m\}$ for all $b = 1, \ldots, B$
$\hat{G}^*_{\max} \leftarrow ecdf([\hat{u}^*_{\max,b}]_{b=1,2,\ldots,B})$
$U_i \leftarrow infl.func([\hat{y}_{is}]_{i=1,2,\ldots,m})$ for all $i = 1, \ldots, n$
$pval \leftarrow 1$ ▷ initialize p-value
$\tau \leftarrow 0$ ▷ initialize tilting parameter
while $pval \neq \alpha$ **do**
    if $pval > \alpha$ **then** decrease τ
    end if
    if $pval < \alpha$ **then** increase τ
    end if
    $w_i \leftarrow exp.tilt.weights(\tau, U_i)$ for all $i = 1, \ldots, n$
    $W_b \leftarrow rel.likelihood([w_i]_{i=1,2,\ldots,n}, M_{ib})$ for all $b = 1, \ldots, B$
    $\hat{H}^*_{s,\tau} \leftarrow tilt.ecdf(W_b, [T^*_{sb}]_{b=1,\ldots,B})$ ▷ reweighted ecdf for selected model s
    $T^0_s \leftarrow T_s(\tau)$ ▷ test statistic under null hypothesis
    $pval \leftarrow 1 - \hat{G}^*_{\max}[\hat{H}^*_{s,\tau}(T^0_s)]$
end while
$\hat{L}_s \leftarrow weighted.mean([\hat{y}_{is}]_{i=1,2,\ldots,m}, [w_i]_{i=1,2,\ldots,n})$

C.2 Additional simulation experiment scenarios

In this Appendix, we will provide a detailed list of the simulation experiments considered in Rink & Brannath (2025), because, in the present work, we only present a subset, since the general findings are consistent across the scenarios. In particular, in Rink & Brannath (2025) we conducted the LASSO and random forest simulation experiment scenarios presented in Tables C.1 and C.2, respectively.

In addition, in Rink & Brannath (2025) we also considered the area under the receiver operating characteristic curve, or, for short, AUC, as another performance measure in the LASSO simulation experiments. Table C.3 shows the simulations

Features	Sample size	Validation	Selection rule
Normal	200	three-split	within one standard-error
Normal	400	three-split	within one standard-error
Normal	200	cross-validation	within one standard-error
Normal	*400*	*cross-validation*	*within one standard-error*
Normal	200	three-split	top ten percent
Normal	400	three-split	top ten percent
Normal	200	cross-validation	top ten percent
Normal	400	cross-validation	top ten percent
Caret	200	three-split	within one standard-error
Caret	400	three-split	within one standard-error
Caret	200	cross-validation	within one standard-error
Caret	400	cross-validation	within one standard-error
Caret	200	three-split	top ten percent
Caret	400	three-split	top ten percent
Caret	200	cross-validation	top ten percent
Caret	400	cross-validation	top ten percent

Table C.1: LASSO simulation experiment scenarios in Rink & Brannath (2025). The one scenario printed in italic letters was adapted to test statistics and its results are presented in this work

Features	Sample size	Validation	Selection rule
Caret	200	TuneRanger	top ten percent
Caret	*400*	*TuneRanger*	*top ten percent*
Caret	600	TuneRanger	top ten percent

Table C.2: Random forest simulation experiment scenarios in Rink & Brannath (2025). The one scenario printed in italic letters was adapted to test statistics and its results are presented in this work

Features	Sample size	Validation	Selection rule
Normal	400	cross-validation	top ten percent
Normal	400	cross-validation	within one standard-error
Normal	600	cross-validation	top ten percent
Normal	600	cross-validation	within one standard-error
Caret	400	cross-validation	top ten percent
Caret	400	cross-validation	within one standard-error
Caret	600	cross-validation	top ten percent
Caret	600	cross-validation	within one standard-error

Table C.3: LASSO simulation experiment scenarios for AUC in Rink & Brannath (2025)

conducted there. Note that we do not consider AUC in the present work at all, due to the lack of theoretical justification.

C.3 `caret` features

Here, we provide the details how we compute the class probabilities $\boldsymbol{\pi}$ in the `caret` feature case. The multiplication of vectors is meant componentwise.

$$\boldsymbol{\pi} = -5 \cdot \mathbf{1} - 4\boldsymbol{x}_1 + 4\boldsymbol{x}_2 + 2\boldsymbol{x}_1\boldsymbol{x}_2 + \sum_{j=3}^{12} \frac{(-1)^j(13-j)}{4}\boldsymbol{x}_j + \boldsymbol{x}_{13}^3 + 2e^{-6(\boldsymbol{x}_{13}-0.3)^2} + 2\sin(\boldsymbol{x}_{14}\boldsymbol{x}_{15}) + \sum_{j=16}^{515} \boldsymbol{x}_j + \sum_{j=516}^{1000} \boldsymbol{x}_j, \quad \text{(C.1)}$$

where

- $\boldsymbol{x}_1, \boldsymbol{x}_2 \sim \mathrm{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & 0.65\\0.65 & 1\end{bmatrix}\right)$;
- $\boldsymbol{x}_3, \boldsymbol{x}_4, \ldots, \boldsymbol{x}_{12} \sim \mathrm{N}(\mathbf{0}, [\mathbb{1}\{i=j\}]_{i,j})$;
- $\boldsymbol{x}_{13}, \boldsymbol{x}_{14}, \boldsymbol{x}_{15} \sim \mathrm{Uni}[0,1]$ i. i. d.;
- $\boldsymbol{x}_{16}, \boldsymbol{x}_{17}, \ldots, \boldsymbol{x}_{515} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{E})$ i. i. d., where $\boldsymbol{E}$ is the unit matrix;
- and $\boldsymbol{x}_{516}, \boldsymbol{x}_{517}, \ldots, \boldsymbol{x}_{1000} \sim \mathrm{N}(\mathbf{0}, [C_{i,j}]_{i,j})$, where $[C_{i,j}]_{i,j} = \max(\mathbb{1}\{i=j\}, 0.8)$ is a compound symmetry covariance matrix.

C.4 Two variants of multiplicity correction

Overall comparison						
Distribution	Min	Q1	Median	Mean	Q3	Max
ECDF	0.544	0.680	0.711	0.708	0.740	0.821
$N(0, 1)$	0.545	0.676	0.706	0.704	0.735	0.819
Per-data set differences						
Difference	Min	Q1	Median	Mean	Q3	Max
ECDF - $N(0, 1)$	–0.003	0.003	0.005	0.005	0.008	0.02

Table C.4: Summaries on the lower confidence limits from MABT using the empirical and normal distribution transformation, respectively (upper table), and per-data set differences (lower table)

C.5 Conditional and unconditional performance

Regular experiments						
Method	Min	Q1	Median	Mean	Q3	Max
BBC-CV	87.0%	89.5%	90.3%	90.4%	91.4%	94.7%
MABT	94.2%	94.9%	95.5%	95.5%	95.9%	97.1%
NCV	90.4%	92.9%	94.5%	94.4%	95.9%	97.1%
Distribution shift experiments						
Method	Min	Q1	Median	Mean	Q3	Max
BBC-CV	83.4%	86.4%	87.2%	87.4%	88.4%	91.6%
MABT	95.3%	95.8%	96.1%	96.1%	96.5%	96.7%
NCV	85.7%	89.1%	91.0%	91.2%	93.5%	95.5%

Table C.5: Summaries on the estimated conditional coverage probabilities. The upper table refers to the regular experiments, while the lower refers to the distribution shift experiments

Tuning parameter ($\times 10^{-3}$)	Proportion of covering intervals	Tuning parameter ($\times 10^{-3}$)	Proportion of covering intervals
1.04	94.7	2.00	96.6
1.14	95.9	2.09	95.4
1.20	95.1	2.19	94.6
1.26	95.3	2.30	96.7
1.32	95.7	2.41	94.8
1.38	96.0	2.52	95.5
1.44	94.6	2.64	95.7
1.51	94.9	2.77	95.8
1.58	95.2	2.90	96.1
1.74	94.9	3.04	95.0
1.82	94.2	3.18	95.9
1.91	97.1	3.33	95.7

Table C.6: Coverage probabilities in the conditional coverage probability simulation experiments for MABT, aggregated by the value of the hyperparameter λ

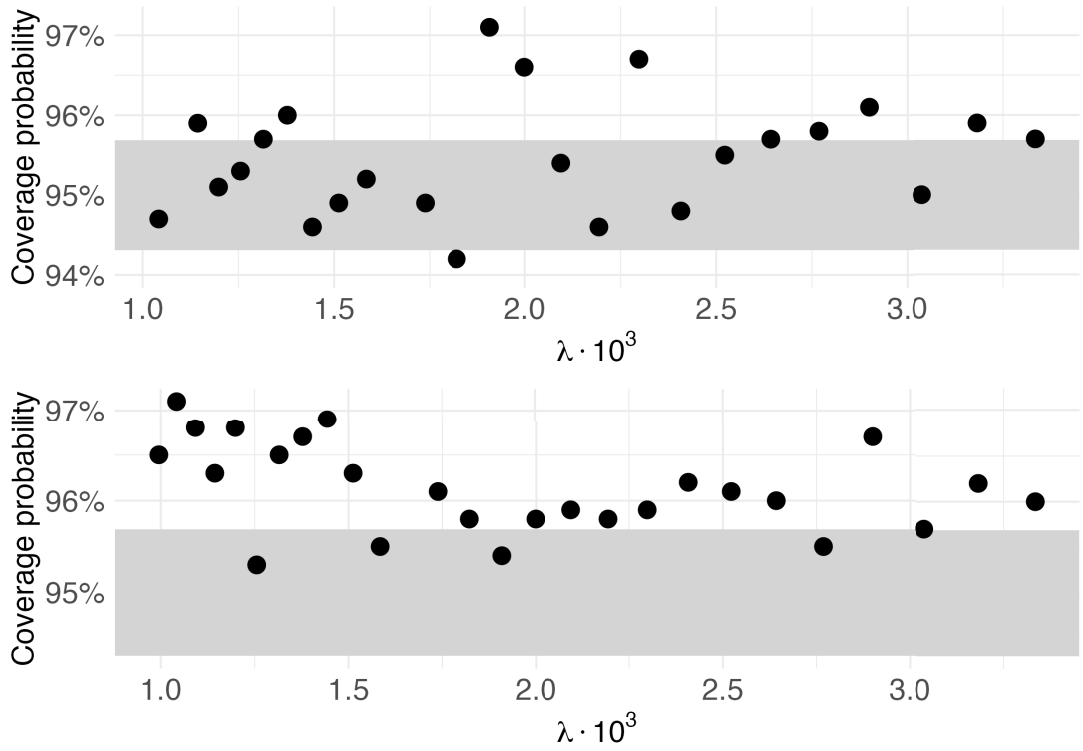

Figure C.1: Estimated conditional coverage probabilities of MABT confidence intervals, aggregated by the value of the hyperparameter λ. The top pane shows the results to the regular simulations experiments, the lower pane shows the results to the distribution shift experiments. The shaded area corresponds to the *acceptable coverage region*

Hyperparameter value ($\times 10^{-3}$)	Proportion of covering intervals (%)		Hyperparameter value ($\times 10^{-3}$)	Proportion of covering intervals (%)	
	BBC-CV	NCV		BBC-CV	NCV
0.99	91.7	96.3	6.40	87.6	91.9
1.04	92.8	96.9	6.70	88.9	93.0
1.09	91.0	96.0	7.02	89.0	93.1
1.14	91.4	96.0	7.35	89.9	94.0
1.20	90.8	96.7	7.70	91.2	94.0
1.26	88.1	96.3	8.07	89.2	92.7
1.32	90.5	96.7	8.45	90.5	92.4
1.38	91.8	96.4	8.86	91.1	93.0
1.44	88.6	95.7	9.28	87.7	92.3
1.51	90.7	97.0	9.72	89.7	90.4
1.58	89.6	97.1	10.18	92.3	93.7
2.41	89.4	95.8	10.67	89.7	92.1
2.64	88.1	95.7	11.18	91.1	94.3
2.90	89.1	96.3	11.71	90.6	92.8
3.04	90.3	96.8	12.27	91.2	91.5
3.18	89.5	94.5	12.85	90.1	91.3
3.33	89.5	95.6	13.46	91.3	92.3
3.49	89.8	94.6	14.10	91.0	91.2
3.66	90.0	95.5	14.77	89.8	92.0
3.83	89.7	95.5	15.48	92.7	91.9
4.02	87.0	95.8	16.21	91.4	92.2
4.21	90.1	96.3	16.99	92.2	93.6
4.41	90.3	96.1	17.79	93.4	93.3
4.62	90.4	95.4	18.64	91.5	93.4
4.84	89.8	95.4	19.53	92.0	94.7
5.07	89.3	94.4	20.46	92.6	93.7
5.31	87.9	94.7	21.43	93.7	94.3
5.56	87.7	95.5	22.45	94.3	96.2
5.83	89.5	93.7	23.52	94.7	95.7
6.10	88.8	92.8			

Table C.7: Estimated coverage probabilities for BBC-CV and NCV, aggregated by the value of the LASSO tuning parameter

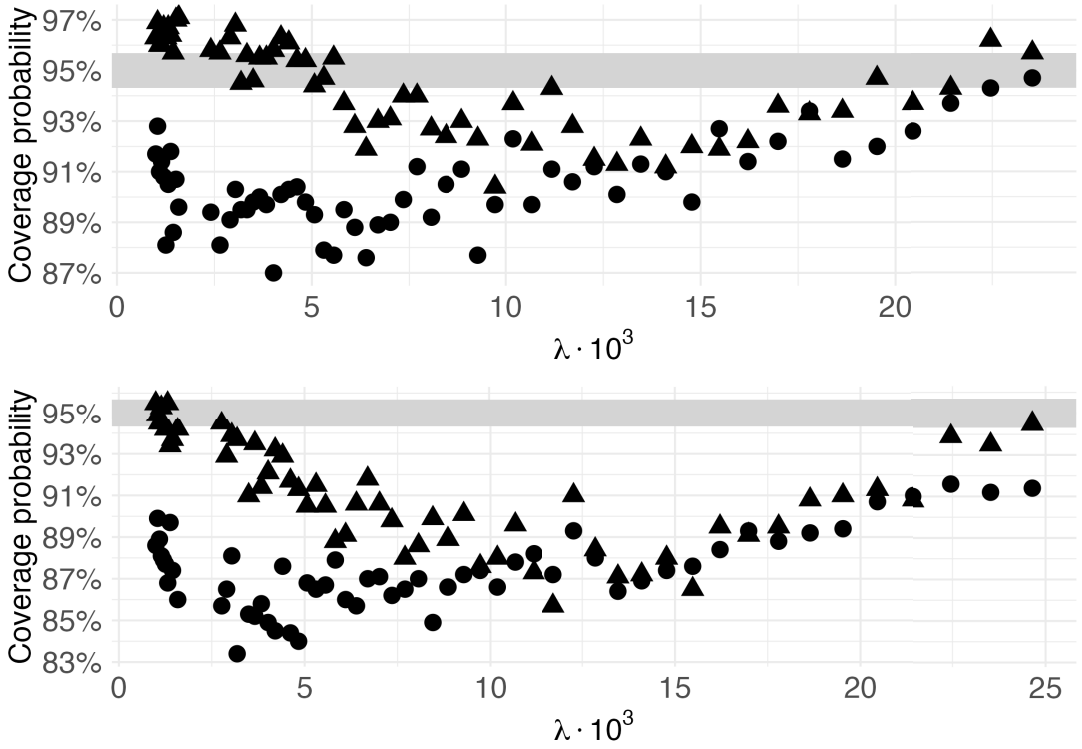

Figure C.2: Estimated conditional coverage probabilities of BBC-CV and NCV confidence intervals against the selected hyperparameter λ. The top pane shows the results to the regular simulations experiments, the lower pane shows the results to the distribution shift experiments. The shaded area corresponds to the *acceptable coverage region*

Tuning parameter ($\times 10^{-3}$)	Proportion of covering intervals	Tuning parameter ($\times 10^{-3}$)	Proportion of covering intervals
0.99	96.5	1.91	95.4
1.04	97.1	2.00	95.8
1.09	96.8	2.09	95.9
1.14	96.3	2.19	95.8
1.20	96.8	2.30	95.9
1.26	95.3	2.41	96.2
1.32	96.5	2.52	96.1
1.38	96.7	2.64	96.0
1.44	96.9	2.77	95.5
1.51	96.3	2.90	96.7
1.58	95.5	3.04	95.7
1.74	96.1	3.18	96.2
1.82	95.8	3.33	96.0

Table C.8: Estimated coverage probabilities for the MABT confidence intervals in the distribution shift experiments, aggregated by the value of the LASSO tuning parameter

Hyperparameter value ($\times 10^{-3}$)	Proportion of covering intervals (%)		Hyperparameter value ($\times 10^{-3}$)	Proportion of covering intervals (%)	
	BBC-CV	NCV		BBC-CV	NCV
0.99	88.6	95.5	7.02	87.1	90.6
1.04	89.9	94.9	7.35	86.2	89.8
1.09	88.9	94.5	7.70	86.5	88.0
1.14	88.1	95.3	8.07	87.0	88.6
1.20	87.9	94.5	8.45	84.9	89.9
1.26	87.7	94.2	8.86	86.6	88.9
1.32	86.8	95.5	9.28	87.2	90.1
1.38	89.7	93.4	9.72	87.4	87.6
1.44	87.4	93.7	10.18	86.6	88.0
1.58	86.0	94.2	10.67	87.8	89.6
2.77	85.7	94.5	11.18	88.2	87.3
2.90	86.5	92.9	11.71	87.2	85.7
3.04	88.1	93.9	12.27	89.3	91.0
3.18	83.4	93.7	12.85	88.0	88.4
3.49	85.3	91.0	13.46	86.4	87.1
3.66	85.2	93.5	14.10	86.9	87.2
3.83	85.8	91.4	14.77	87.4	88.0
4.02	84.9	92.1	15.48	87.6	86.5
4.21	84.5	93.2	16.21	88.4	89.5
4.41	87.6	92.9	16.99	89.3	89.1
4.62	84.4	91.7	17.79	88.8	89.5
4.84	84.0	91.3	18.64	89.2	90.8
5.07	86.8	90.5	19.53	89.4	91.0
5.31	86.5	91.5	20.46	90.7	91.3
5.56	86.7	90.5	21.43	91.0	90.8
5.83	87.9	88.8	22.45	91.6	93.9
6.10	86.0	89.1	23.52	91.2	93.5
6.40	85.7	90.6	24.64	91.4	94.5
6.70	87.0	91.8			

Table C.9: Estimated coverage probabilities for the BBC-CV and NCV confidence limits in the distribution shift experiments, aggregated by the value of the LASSO tuning parameter

C.6 LASSO simulation experiments

Lower confidence limits							
Pipeline	Method	Min	Q1	Median	Mean	Q3	Max
Default	CP	0.503	0.669	0.701	0.701	0.734	0.813
Default	Wald	0.509	0.679	0.712	0.709	0.745	0.815
Default	Wilson	0.508	0.673	0.705	0.704	0.738	0.816
Proposed	CP	0.494	0.654	0.688	0.686	0.720	0.817
Proposed	MABT	0.544	0.680	0.711	0.708	0.740	0.821
Proposed	Wald	0.504	0.668	0.701	0.699	0.734	0.816
Proposed	Wilson	0.499	0.657	0.690	0.688	0.722	0.818
Tightness							
Pipeline	Method	Min	Q1	Median	Mean	Q3	Max
Default	CP	0.000	0.052	0.080	0.083	0.110	0.265
Default	Wald	0.000	0.044	0.071	0.074	0.101	0.258
Default	Wilson	0.000	0.049	0.077	0.079	0.106	0.260
Proposed	CP	0.000	0.068	0.097	0.099	0.128	0.284
Proposed	MABT	0.000	0.047	0.074	0.077	0.103	0.247
Proposed	Wald	0.000	0.054	0.083	0.086	0.115	0.275
Proposed	Wilson	0.000	0.067	0.096	0.097	0.126	0.280

Table C.10: Summaries on the lower confidence limits (upper table) and tightness (lower table) in the LASSO simulation experiments

Pipeline	Min	Q1	Median	Mean	Q3	Max
Default	0.654	0.771	0.786	0.783	0.798	0.831
Proposed	0.697	0.773	0.787	0.785	0.799	0.831

Table C.11: Summaries on the true prediction performances of the final selected models from the default and the proposed selection-evaluation pipelines in the LASSO simulation experiments

C.7 Random forest simulation experiments

Lower confidence limits							
Pipeline	Method	Min	Q1	Median	Mean	Q3	Max
Default	CP	0.443	0.626	0.658	0.658	0.690	0.767
Default	Wald	0.448	0.635	0.668	0.666	0.701	0.768
Default	Wilson	0.448	0.631	0.662	0.661	0.694	0.771
Proposed	CP	0.440	0.607	0.640	0.639	0.672	0.772
Proposed	MABT	0.480	0.640	0.670	0.669	0.699	0.773
Proposed	Wald	0.446	0.621	0.655	0.653	0.689	0.774
Proposed	Wilson	0.446	0.609	0.641	0.640	0.673	0.771

Tightness							
Pipeline	Method	Min	Q1	Median	Mean	Q3	Max
Default	CP	0.000	0.055	0.083	0.085	0.114	0.248
Default	Wald	0.000	0.046	0.075	0.078	0.106	0.242
Default	Wilson	0.000	0.052	0.080	0.082	0.110	0.243
Proposed	CP	0.000	0.077	0.107	0.107	0.138	0.263
Proposed	MABT	0.000	0.048	0.076	0.078	0.103	0.224
Proposed	Wald	0.000	0.060	0.091	0.093	0.123	0.257
Proposed	Wilson	0.000	0.076	0.106	0.106	0.135	0.258

Table C.12: Summaries on the lower confidence limits (upper table) and tightness (lower table) in the random forest simulation experiments

Pipeline	Min	Q1	Median	Mean	Q3	Max
Default	0.632	0.735	0.746	0.743	0.755	0.785
Proposed	0.628	0.738	0.748	0.746	0.757	0.790

Table C.13: Summaries on the true prediction performances of the final selected models from the default and the proposed selection-evaluation pipelines in the random forest simulation experiments

D Appendix to Chapter 7

Absolute difference						
Method	Min	Q1	Median	Mean	Q3	Max
CP	–0.002	0.002	0.003	0.008	0.007	0.067
Wald	–0.013	–0.002	0.002	0.004	0.004	0.059
Wilson	–0.004	0.001	0.003	0.007	0.006	0.065
Relative difference						
Method	Min	Q1	Median	Mean	Q3	Max
CP	–0.002	0.002	0.004	0.010	0.010	0.087
Wald	–0.014	–0.002	0.002	0.005	0.006	0.08
Wilson	–0.004	0.001	0.004	0.008	0.009	0.083
Maximum achievable improvement						
Method	Min	Q1	Median	Mean	Q3	Max
CP	–0.016	0.007	0.023	0.049	0.042	0.298
Wald	–0.118	–0.009	0.005	0.022	0.028	0.272
Wilson	–0.033	0.003	0.017	0.043	0.039	0.291

Table D.14: Summaries on the absolute and relative differences (upper and central table) between the MABT confidence limits and the comparators as well as on the maximum achievable improvement (lower table) in the OpenML benchmark

Task ID	Data ID	Name	Target var	Maj class size	Min class size	No of features	Sample size
3	3	kr-vs-kp	class	1669	1527	37	3196
31	31	credit-g	class	700	300	21	1000
37	37	diabetes	class	500	268	9	768
43	44	spambase	class	2788	1813	58	4601
49	50	tic-tac-toe	Class	626	332	10	958
219	151	electricity	class	26075	19237	9	45312
3485	312	scene	Urban	1976	431	300	2407
3492	333	monks-problems-1	class	278	278	7	556
3493	334	monks-problems-2	class	395	206	7	601
3494	335	monks-problems-3	class	288	266	7	554
3891	1038	gina_agnostic	label	1763	1705	971	3468
3899	1046	mozilla4	state	10437	5108	6	15545
3902	1049	pc4	c	1280	178	38	1458
3903	1050	pc3	c	1403	160	38	1563
3913	1063	kc2	problems	415	107	22	522
3917	1067	kc1	defects	1783	326	22	2109
3918	1068	pc1	defects	1032	77	22	1109
3954	1120	MagicTelescope	class:	12332	6688	12	19020
9946	1510	wdbc	Class	357	212	31	569
9952	1489	phoneme	Class	3818	1586	6	5404
9957	1494	qsar-biodeg	Class	699	356	42	1055
9967	1504	steel-plates-fault	Class	1268	673	34	1941
9970	1479	hill-valley	Class	606	606	101	1212
9971	1480	ilpd	Class	416	167	11	583
9976	1485	madelon	Class	1300	1300	501	2600
9978	1487	ozone-level-8hr	Class	2374	160	73	2534
9980	1467	climate-model-simulation-crashes	Class	494	46	21	540
9983	1471	eeg-eye-state	Class	8257	6723	15	14980
10093	1462	banknote-authentication	Class	762	610	5	1372
10101	1464	blood-transfusion-service-center	Class	570	178	5	748
14965	1461	bank-marketing	Class	39922	5289	17	45211
34537	4534	PhishingWebsites	Result	6157	4898	31	11055
34539	4135	Amazon_employee_access	target	30872	1897	10	32769

Table D.15: Overview of the included data sets for binary classification from the OpenML platform