Shared Lunch: A novel testing framework for machine learning algorithms

Devilee, Thomas

Universiteit
Leiden
The Netherlands

Shared Lunch

A novel testing framework for machine learning algorithms

Thomas Hendrik Devilee

Thesis advisor: Prof.dr. J.J. Goeman

Defended on November 30th, 2022

MASTER THESIS

STATISTICS AND DATA SCIENCE
UNIVERSITEIT LEIDEN

Abstract

Machine learning algorithms are frequently deployed for predictive classification problems. However, as implied by the No Free Lunch (NFL) theorem, not every algorithm is destined to perform well on a given data set. One is often interested whether predictive capacity of an algorithm is significantly better than chance level (better than chance) for a choice of the hyperparameter(s). Machine learning algorithms generally lack the intrinsic statistical framework of statistical learning algorithms to make statements about better than chance performance. In supervised binary classification problems, multiple methods have been proposed to do so. These have shown to be flawed. Arguably, this can be attributed to the idea that the NFL also applies to these better than chance methods, or in general to any tests, suggesting that performance of the test depends on the type of signal.

Therefore, in this current project, we propose novel global test (GT) based tests that are in accordance with the signal detected by their respective learning algorithm. To do so, we reformulated two popular machine learning algorithms, k-nearest neighbors (kNN) and random forest, as empirical Bayesian linear models. It turned out we can not only construct tests for specific (combinations of) hyperparameters but also for sets of hyperparameters. Properties of these tests have been explored in simulated linearly and nonlinearly separable alternatives as well as in real world data. Results from our simulation studies indicated that our novel tests had competitive power characteristics compared to existing methods. Moreover, we demonstrated their applicability to real world data.

Our finding indicated that our novel tests for kNN and random forest can be readily used to assess better than chance performance. Equally important, the exploited GT framework can be applied to construct tests for other learning algorithms. Ultimately, our tests and possible future GT based tests add to list of existing methods that each serve a niche in the detection of better than chance signal for a learning algorithm.

Contents

Chapter 1

Introduction

Data modelling has gained large popularity in recent years. Where in the early days problems mostly came from scientific experiments, were small in size and uncomplicated, data modelling nowadays happens in virtually all sectors and can be vast and complex. When it comes to data modelling, one often refers to supervised learning in which a certain outcome is predicted based on one or more predictor variables. A vast array of learning algorithms exist to serve this purpose. Initial algorithms were developed from a statistical perspective. This class of algorithms will be referred to as statistical learning algorithms. Advancements in computing and storage technology gave rise to a new class of learning algorithms, machine learning algorithms. In general, statistical learning algorithms are developed to make inference about relationships between variables, whereas machine learning algorithms are more pragmatic, designed to detect patterns and make accurate predictions without the statistical framework.

Not all learning algorithms are destined to perform equally well on every data set. This observation is reflected in the No Free Lunch (NFL) theorem which states that any two learning algorithms perform equally when their performance is averaged across all possible problems [1]. Therefore, common practice in supervised learning problems is to test whether predictive accuracy of a learning algorithm is significantly better than chance level (better than chance), if applicable, for a choice of the hyperparameter(s). To do so, statistical learning algorithms often got access to a framework which allows testing for a relation between individual predictors and the outcome and/or a relation between at least one of the predictors and the outcome. This framework rarely exists in machine learning algorithms, hence requires other approaches.

In the case of supervised machine learning for binary classification problems, there are two widely used groups of tests. *Accuracy-tests* accomplish this by, as their name implies, using prediction accuracy as test statistic. Best known are the tests based on resubstitution accuracy, also known as training accuracy, and k-fold cross-validation accuracy in conjunction with permutation testing. However, Rosenblatt *et al.* recently demonstrated that in high-dimensional problems these approaches can be underpowered compared to a group of tests known as two-group tests [2]. *Two-group tests* are tailored towards detecting either differences in multivariate distribution or a shift in mean vectors

between groups. Indeed, these are computationally less demanding than accuracy-tests as they can be used before fitting a classifier and resampling is not required.

Less known is that the idea of the NFL theorem also applies to hypothesis testing. Thus, despite two-group tests appearing to be an attractive option for evaluation of a classifiers performance, they are not optimal in detecting all types of signals. Ideally, we would like to introduce the testing framework found in statistical learning algorithms to machine learning algorithms in order to be able to test for the signal detected by a learning algorithm. These tests are specific to the learning algorithm and signal, thereby facilitating a Shared Lunch between them.

In this current project we propose methods to assess performance that are in accordance with the signal detected by their respective learning algorithm. These solutions are based on a generalization of the score test, the global test (GT). Similar to classical likelihood based tests (equivalently the F-test in linear regression) in the framework of regression, the GT may be formulated to test whether the coefficients for a set of predictors is equal to zero. In contrast to these classical tests, the GT requires an empirical Bayesian model formulation instead of a classical, purely frequentist, one.

Therefore, we will redevelop two machine learning algorithms, k-nearest neighbors (kNN) and random forest, as empirical Bayesian linear models. Naturally, this alternative view gives rise to a transformation of the predictor matrix that is specific to the algorithm and its parameters. Together these provide the tools to construct GT based tests. Furthermore, we will extend this framework and introduce tests to perform simultaneous inference on multiple valid (combinations of) hyperparameters, thereby opening up possibilities to assess performance over all possible values of the tuning parameters. We will explore these novel tests and compare their performance in terms of power to existing methods in simulated as well as in real data.

Chapter 2

Prerequisites

kNN and random forest are non-parametric supervised machine learning algorithms that can be used for classification tasks as well as regression problems. In this current thesis we develop better than chance methods for these algorithms for binary classification, hence we discuss the algorithms in this context. These learning algorithms are discriminative in the sense that they estimate the conditional probability of some binary outcome y given the predictors X. In principle, the predictors can be continuous, categorical or a mixture. These methods can be applied in low-dimensional problems as well as high-dimensional scenarios.

As discussed in the introduction, (nested) k-fold cross-validation can be deployed to assess whether kNN and random forest, or in general any fitted algorithm, have better than chance predictive performance. These approaches were not developed for this purpose. Instead, cross-validation and its nested variant are best known for assessing predictive capacity and generalized performance, respectively, of a learning algorithm. In principle, this cross-validation framework can be applied to any algorithm as long as some measure of performance (e.g. mean-squared error or accuracy) can be computed. In the context of evaluation of better than chance performance, we have seen that this framework is not optimal in terms of power for the detection of all types of signal fitted by a learning algorithm.

In this current thesis, we propose novel tests based on the GT that correspond with signal detected by the learning algorithm. The GT is a generalization of the score test on a hyperparameter in an empirical Bayesian linear model. Therefore, it is often deployed in high-dimensional problems to test a point null hypothesis against an alternative. It has to be noted that the GT can also be used in low-dimensional scenarios.

In this chapter we provide a brief introduction to kNN, random forest, nested k-fold cross-validation and the GT. We expand on or refer to these topics in this current thesis.

2.1 kNN

kNN is one the best known machine learning algorithms due to its simplicity. Given a training set T of size n and some integer value for k where $1 \leq k \leq n$, kNN makes

predictions for a new point based on the k closest observations T_0 in the training set according to some distance metric. This metric is often taken to be the Euclidean distance for continuous variables and the Hamming distance for discrete variables. For specific applications such as gene expression microarray data, a correlation metric is often used.

In the context of classification problems, a new observations is classified according to the highest estimated probability. The estimated probability of class j for a new observation x_0 is equal to the fraction of observations in T_0 with label j

$$\Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in T_0} I(y_i = j).$$

For binary classification problems this can be further simplified to

$$\begin{aligned} \Pr(Y = 1|X = x_0) &= \frac{1}{K} \sum_{i \in T_0} I(y_i = 1) \\ \Pr(Y = 0|X = x_0) &= 1 - \Pr(Y = 1|X = x_0). \end{aligned}$$

In general kNN is considered to be a low bias, high variance learning algorithm. In other words, it is able to fit nonlinear structures in the data but its predictions become unstable as the noise in the data increases. This trade-off, known as the *bias-variance trade-off*, can be altered by the choice of k. Smaller values for k tend to have lower bias and higher variance compared to larger values for k. Accordingly, larger values for k are known to have more bias but lower variance in comparison to smaller values for k.

The bias-variance trade-off for kNN can be illustrated using a thought experiment. Consider the situation when the true data generating function is reasonably smooth. When $k = n$, the prediction for an observation is equal to the most prominent class in the training set. This prediction is insensitive to small fluctuations in the training set (lower variance) but it does not approximate the true data generating function well (more bias). Now, consider the complete opposite case when $k = 1$. Predictions are now equal to the label of the nearest neighbour. On average, predictions are close to the true label (less bias). However, predictions are unstable as they solely depend on one observation (higher variance). Thus, in practice the optimal k in terms of bias and variance lies in between these extremes and is conditional on the data set.

2.2 Random forest

In contrast to kNN, random forest is an ensemble learning method since it uses multiple learning algorithms, in this case decision trees, to make predictions. An ensemble of

decision trees often has better predictive performance compared to a single decision tree. The bias of a single decision tree is relatively low since they can approximate any arbitrary function reasonably well. However, its variance is large as small fluctuations in the training set affect the fit. This variance problem can be ameliorated by a tree ensemble. The general idea is to grow deep trees that have low correlation. The former implies low bias while the latter decreases variance, thereby improving overall predictive performance compared to a single decision tree. In an approach known as bagging, trees are decorrelated by taking a bootstrap sample of the observations for each tree separately. Trees can be further decorrelated when at each split only a fraction of the original predictors is considered by sampling. This is known as the random forest algorithm.

In tree ensemble methods each tree is grown separately using *recursive binary splitting*. This is a top-down approach. Meaning that it starts with all observations and successively splits the predictors space, giving rise to two new *nodes*. When a node marks the end of a branch this is more commonly known as a *leaf*. It is a greedy algorithm in the sense that it looks for the optimal split at a specific step given some metric. In other words, it does not pick a split that yields a better tree at some splitting point later on. For each split the algorithm selects some cutting point c in predictor j that minimizes a metric L,

$$P_1(j,c) = \{X|X_j < c\} \text{ and } P_2(j,c) = \{X|X_j \geq c\}$$

where j and c are found by minimizing

$$\sum_{i:x_i \in P_1(j,c)} L(y_i, \hat{y}_{i,P1}) + \sum_{i:x_i \in P_2(j,c)} L(y_i, \hat{y}_{i,P2})$$

and $\hat{y}_{i,P1}$ and $\hat{y}_{i,P2}$ depend on the outcomes of training observations in $P_1(j,c)$ and $P_2(j,c)$, respectively.

The process of growing trees is governed by multiple parameters. The metric is one of the most important hyperparameters. In regression it is often taken to be the residual sum of squares, while Gini index or entropy are commonly used in classification tasks. Another important hyperparameter concerns the structure of the tree. These often come in the form of a maximum tree depth, the number of nodes or leaves, minimum number of observations per leaf or node or a combination of these.

2.3 Nested k-fold cross-validation and permutation testing

For a given learning algorithm, conventional k-fold cross-validation provides an estimate of predictive performance for a choice of the hyperparameter(s). This framework does not

suffice when one wants to simultaneously perform hyperparameter selection and estimate predictive performance. In these scenario nested k-fold cross-validation is required. The provided estimate can interpreted as the generalized performance of the algorithm.

There is resemblance between the procedure of nested k-fold cross-validation and its non-nested variant. The main difference is that nested k-fold cross-validation uses two loops, an outer an inner loop. The procedure for the nested variant is as follows. First, data is randomly split into k-folds that are, when possible, of equal size. In the outer loop one fold serves as test set while the remaining folds constitute the training set. Of these training folds, the inner loop sets one fold aside as validation set. A learning algorithm is fitted for a number of (combinations of) parameters on the remaining $k-2$ folds. Some metric is computed for the validation set. In this project we decided on accuracy. This process is repeated such that every of the $k-1$ folds served as validation set. Accuracy is averaged over the validations sets. The parameter corresponding to the highest accuracy is used to fit the algorithm to the training set of the outer loop. Accuracy is computed for the test set of the outer loop. This process is repeated such that all k folds served as outer loop test set once. The average accuracy over all these folds serves as estimate for the generalized performance.

The (nested) k-fold cross-validation is combined with permutation testing to form the basis of a selection of accuracy-tests. When we permute outcomes and apply the cross-validation procedure a (large) number of times, we obtain the empirical null distribution for no association between predictors and the outcome. This can be used to assess performance of a fitted learning algorithm. Identical to the use of the cross-validation framework to provide an estimate performance, the choice of either a nested or non-nested procedure depends on whether the hyperparameter has to be selected or not, respectively.

2.4 Global test

Classical test for linear regression either have low power or break down completely when the number of predictors is close to or exceeds the sample size, respectively. In these high dimensional problems the GT may serve as an alternative. Among all tests, the locally most powerful test has optimal power against alternatives close to the null hypothesis.

Consider an empirical Bayesian linear model where the intercept $\alpha \in \mathbb{R}$ and error variance σ^2 are known and fixed,

$$y_i|\beta \sim \mathcal{N}(\alpha + X_i\boldsymbol{\beta}, \sigma^2) \tag{2.1}$$

where X_i is a row vector for observation i and $\boldsymbol{\beta}$ a random column vector, both of length p. The latter is defined as $\boldsymbol{\beta} = \tau\boldsymbol{b}$, $\mathrm{E}[\boldsymbol{b}] = \mathbf{0}$ and $\mathrm{E}[\boldsymbol{b}\boldsymbol{b}^T] = \Sigma$. In this case $\tau \in \mathbb{R}$ is a fixed but unknown parameter. Also note that no assumptions about the marginal distribution of $\boldsymbol{b}$ are made.

Let $l(\boldsymbol{\beta})$ denote the likelihood of $\boldsymbol{\beta}$ given our model in (2.1). Taking the expectation over an chosen distribution of $\boldsymbol{\beta}$ for fixed τ^2 yields the marginal likelihood of τ^2

$$\hat{l}(\tau^2) = \mathrm{E}_{\boldsymbol{\beta}|\tau^2}[l(\boldsymbol{\beta})].$$

This allows to make inference about τ^2. By means of the locally most powerful test we can formulate the hypothesis

$$H_0 : \tau^2 = 0$$
$$H_A : \tau^2 > 0.$$

When $\tau^2 = 0$ then $\beta = 0$, hence this is equivalent to testing

$$H_0 : \boldsymbol{\beta} = \mathbf{0}$$
$$H_A : \boldsymbol{\beta} \neq \mathbf{0}.$$

This approach has two major drawbacks. First of all, computation of the marginal likelihood $\hat{l}(\tau^2)$ is often intractable as it requires integration over p-dimensions. Second, the marginal likelihood of τ^2 requires specification of the distribution of $\boldsymbol{\beta}$. Power against alternatives depends on this choice, hence it has to be carefully specified. Making an informed decision is typically difficult in high dimensional problems.

However, Goeman *et al.* have shown the score test statistic corresponding to $\hat{l}(\tau^2)$ to be computable from the likelihood of $\boldsymbol{\beta}$, $l(\boldsymbol{\beta})$, and the variance-covariance matrix of $\boldsymbol{b}$, Σ [3]. The score test statistic for our empirical Bayesian linear model is

$$S_\Sigma = \frac{y^T X \Sigma X^T y}{y^T y}.$$

The choice of Σ determines the power against certain alternatives. The case when $\Sigma = I$ deserves special attention as it implies exchangeability of the elements in $\boldsymbol{\beta}$. A sequence of random variables is said to be exchangeable when all possible permutations share the same joint distribution. In other words, when the variance-covariance matrix is chosen to be the identity matrix we do not express a prior belief about the magnitude of elements from $\boldsymbol{\beta}$ and covariance among them. We will not be discussing other choices of Σ as only the exchangeable version of the test is used in this current thesis. The locally most powerful test statistic reduces to

$$S = \frac{y^T X X^T y}{y^T y}. \tag{2.2}$$

The GT is not invariant to parametrization of covariates. To demonstrate this we can write the test statistic as the sum of test statistics over p covariates

$$S = \sum_{i=1}^{p} \frac{y^T x_i x_i^T y}{y^T y}.$$

Write

$$x_i x_i^T = x_i^T x_i \frac{x_i x_i^T}{x_i^T x_i}.$$

Taking $q_i = \frac{x_i}{\sqrt{x_i^T x_i}}$ and $\lambda_i = x_i^T x_i$, we obtain

$$S = \sum_{i=1}^{p} \lambda_i \frac{y^T q_i q_i^T y}{y^T y}.$$

By definition the rank of $x_i x_i^T$ is 1. Here λ_i is the eigenvalue and q_i the corresponding eigenvector of the inner product matrix of covariate i.

We can observe that λ_i is either equal or proportional to the variance of a regressor i depending on whether it is centred or not. Multiplication of x_i with a scalar $\sqrt{\gamma}$ not only increases the variance by a factor γ but also λ_i. In other words, the GT is a weighted sum of the test statistics over p predictors where the weight linearly depends on the variance. This weighted sum of individual GT statistics from (groups of) covariates is referred to as stacking of the GT.

In linear models one often wants to test for any effect of the predictors of the outcome. In this scenario the intercept forms a nuisance covariate which is not tested for. The test in its current form (2.2) does not correct for an intercept, or in general, for nuisance covariates Z. Correcting is done by orthogonalizing X and y with respect to Z such that $Z^T(I - H)X = 0$ and $Z^T(I - H)y = 0$ where I is the identity matrix and H a projection matrix given by $Z(Z^T Z)^{-1} Z^T$ [4]. Let $\tilde{X}$ and $\tilde{y}$ denote the orthogonalized matrices. Note that in case the intercept is the only nuisance parameter $H = \frac{1}{n}\mathbf{1}\mathbf{1}^T$ where n is the number of observations, which is equivalent to centring the covariates. Incorporating this adjustment in the score test statistic yields

$$S = \frac{\tilde{y}^T \tilde{X} \tilde{X}^T \tilde{y}}{\tilde{y}^T \tilde{y}}.$$

Since $I - H$ is idempotent, the locally most powerful test statistic, adjusting for general nuisance covariates Z, can be written as

$$S = \frac{y^T(I - H)XX^T(I - H)y}{y^T(I - H)y}. \tag{2.3}$$

Computation of the p-value corresponding to a test statistic requires the null distribution of the test statistic. Two approaches are available to generate the null distribution [5]. The asymptotic distribution can be computed when the sample size is larger than the number of nuisance parameters. When this is not the case or one or more assumptions of linear regression are possibly violated, the null distribution may be generated via permutations. In this approach null hypotheses are assumed to be exchangeable. This implies that the joint distribution is invariant under permutations given some null hypothesis.

Chapter 3

kNN

In chapter 2 we have seen that the GT requires an empirical Bayesian linear model and that kNN is not of this form as it is a non-parametric algorithm. Therefore, kNN has to be redeveloped as empirical Bayesian linear model in order to construct GT based tests. In this current chapter we introduce a transformation of predictors space that is appurtenant to such a reformulation. Together these allow for the construction of tests that allow to make statements about performance of kNN.

3.1 GT for specific k

The most common scenario is that one wants to evaluate whether kNN predicts better than chance for some value of k on a given data set. In this section we propose multiple GT based tests to do so. All these tests are based on the same transformation of predictor space. We first introduce the transformation and subsequently show that this corresponds to a linear model.

The distance matrix of original data is used in the computation of the kNN predictor matrix. Consider a training set of size n where for each observation i the outcome is binary $\{0, 1\}$ and the regressors reside in p-dimensional space $\mathbb{R}^p$. All pairwise Euclidian distances can be represented in a n x n distance matrix D. An element from this matrix d_{ij} is defined as the Euclidian norm of the difference between x_i and x_j

$$d_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{p}(x_{ik} - x_{jk})^2}.$$

By definition matrix D is symmetric $d_{ij} = d_{ji}$ with $0's$ on the diagonal $d_{ii} = 0$.

Now, for any valid choice of k we can define the kNN predictor matrix X_k based on D. An element x_{ij} from X_k is equal to 1 when neighbor j belongs to the closest k neighbors of i, otherwise it is equal to 0.

Example 1. *Consider a training set of size* 3 *with the predictors in* $\mathbb{R}^2$*:* $n_1 = [4.1, 1], n_2 = [5.3, 2.6], n_3 = [9.5, 8.2]$*. The distance matrix is given by*

$$D = \begin{bmatrix} 0 & 2 & 9 \\ 2 & 0 & 7 \\ 9 & 7 & 0 \end{bmatrix}.$$

For $k = 1, 2, 3$ *we can compute the predictor matrices*

$$X_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad X_2 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad X_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

The predictor matrices for $k = 1$ and $k = n$ are special cases as they are independent of the data. The closest neighbor to some training observation i is always i. Hence the predictor matrix for $k = 1$ is the identity matrix. When $k = n$ all n training observations are closest to observation i. The predictor matrix for $k = n$ is a matrix full with ones $J = \mathbf{1}\mathbf{1}^T$. For all remaining choices of k the predictor matrix is conditional on the data. Note that in common kNN terminology the scenario of 1 nearest neighbor refers to the nearest neighbor that is not the observation itself. In this current thesis this scenario corresponds to $k = 2$. Effectively, when we refer to k neighbors this corresponds to $k-1$ neighbors in widespread kNN nomenclature.

The definition for our new predictor space allows for construction of the kNN estimator. Given our predictor space X_k for some value of k and response vector y, we can define the kNN estimator $\hat{\mu}$ to be

$$\hat{\mu} = k^{-1} X_k y.$$

Indeed, the estimator is tantamount to the mean of responses from observations that belong to the k nearest neighbors. Taking the expectation of this expression yields

$$\mathbb{E}(\hat{\mu}) = \mathbb{E}(k^{-1} X_k y) = k^{-1} X_k \mathbb{E}(y) = k^{-1} X_k \mu. \tag{3.1}$$

When we take $X = X_k$ and $\beta = k^{-1}\mu$, our model resembles linear regression.

We can define the linear model for kNN. The model is effectively a factorial ANOVA with n variables. To aid interpretation and to be able readily test, we orthogonalize X_k and y with respect to the intercept. Making use of the orthogonalized predictor matrix $\tilde{X}_k$ and outcome $\tilde{y}$ we write the linear model as

$$\tilde{y} = \alpha + \tilde{X}_k \beta + \epsilon, \epsilon \sim N(0, \sigma^2 I), \sigma^2 > 0. \tag{3.2}$$

where α and ϵ are the intercept and error vector, respectively.

In this formulation, the intercept represents the proportion of labels and each element of β can be thought of the effect of the corresponding neighbor on the outcome. In other words, when an observation i belongs to the k nearest neighbors and element i from β is positive, the linear predictor increases. When this element i is negative the linear

predictor decreases. The opposite is the case when an observation i does not belong to the k nearest neighbors; the linear predictor increases when β is negative and decreases when β is positive. Note that in actuality this β vector is never estimated.

We can formulate the alternative hypothesis for any effect of kNN for a choice of k

$$\begin{aligned} H_0 &: \tilde{y} = \alpha \\ H_A &: \tilde{y} = \alpha + \tilde{X_k}\beta. \end{aligned} \tag{3.3}$$

The effect of kNN is fully captured in β. Thus, when the null hypothesis is rejected, there is sufficient evidence, according to the specified type I error, to favour an effect of kNN, at least in this reformulation, over simply the proportion of outcomes as predictor.

In practice it is often infeasible to make a sensible choice for k. Therefore, we propose two approaches that determine k based on the data: the *proportion kNN GT* and *p-value kNN GT*. The proportion kNN GT takes the number of nearest neighbors equal to the number of observations in the group of the least prevalent outcome. Let n_0 denote the number of observations with an outcome of 0 and n_1 the number of observations with an outcome of 1 then $k = min(n_0, n_1)$. The p-value kNN GT randomly splits the data in two equally sized groups: a training and test set. In the training set, kNN predictor matrices are generated for a grid of values for k and appurtenant p-values are computed. The value of k corresponding to the smallest p-value is used for the test set.

Properties

The GT statistic is a constant for the predictor matrices of $k = 1$ and $k = n$. When $k = 1$ we can write the test statistic as

$$S = \frac{y^T(I-H)II^T(I-H)y}{y^T(I-H)y} = \frac{y^T(I-H)y}{y^T(I-H)y} = 1.$$

The GT statistic for $k = n$ is equal to

$$S = \frac{y^T(I-H)JJ^T(I-H)y}{y^T(I-H)y} = \frac{y^T 00 y}{y^T(I-H)y} = 0.$$

For illustration purposes, we compute the p-value for these hyperparameters using permutations of y. All possible permutations of y yield the same test statistic, either 1 when $k = 1$ or 0 when $k = n$. Hence we never reject the null hypothesis for these choices of k.

As previously demonstrated, the GT assigns weights proportional to the variance of variables. Every element from the predictor matrix X_k is by definition $\{0, 1\}$. As a consequence, the variance of a column can be interpreted as the variance of a binomial proportion. This is known to be equal to $np(1-p)$. The variance is maximized at $p = 0.5$ and minimized at $p = 0$ and $p = 1$. The variance of the latter corresponds to zero. Thus nearest neighbors may be weighted differently.

Example 2. *Consider the following training set of size* 4 *with predictors in* $\mathbb{R}^2$*:* $n_1 = [-1, 1], n_2 = [-1, 0], n_3 = [2, 1], n_4 = [2, 0], n_5 = [2, -10]$*. The data has been plotted for illustration purposes (**Figure 3.1A**).*

The kNN predictor matrix for $k = 2$ *is equal to*

$$X_2 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

The variance of the first four columns can be shown to be equal to 1.2 *whereas the last column has a variance of* 0.8*. Compared the fifth observation, the remaining observations are weighed* 50% *more heavily. In other words, this framework implicitly assigns smaller weights to isolated neighbors/outliers.*

Example 3. *Consider the following training set of size* 3 *with predictors in* $\mathbb{R}^2$*:* $n_1 = [-1, 0], n_2 = [0, 0], n_3 = [1, 1]$*. The data has been plotted for illustration purposes (**Figure 3.1B**).*

The kNN predictor matrix for $k = 2$ *is equal to*

$$X_2 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

The second column only contains 1*'s resulting in a variance of zero. Therefore, this predictor is not taken into account when computing the test statistic. A neighbor is effectively removed when it is nearest to all observations. In the context of linear regression this can be understood as a constant predictor; it does not contain any information and can be excluded. This phenomenon occurs more frequently when* k *approaches* n*, saturating the matrix with* 1*'s.*

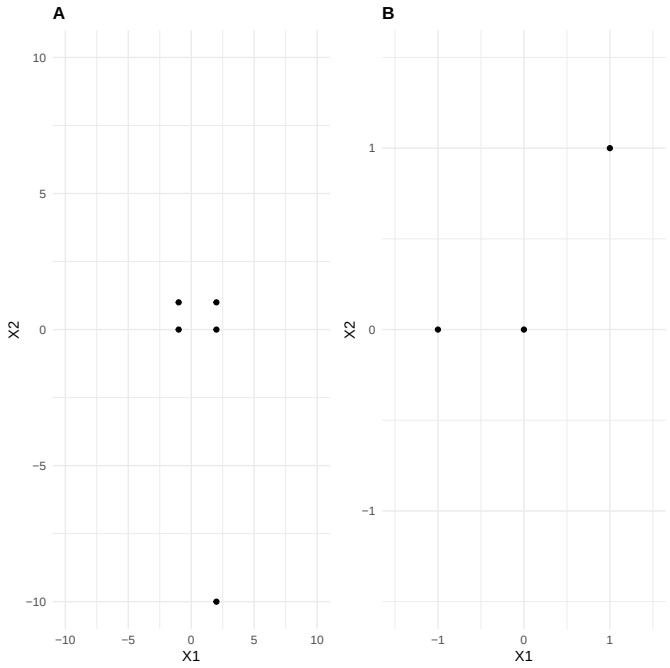

Figure 3.1: **Observations are weighted differently by our novel k-nearest neighbors tests.** Toy data has been plotted to illustrate **(A)** an observation with a relatively small weight and **(B)** an observation with no weight.

3.2 Stacked GT

In the previous section we introduced a novel testing framework for kNN with specific k. However, one might be interested in evaluating whether any valid choice of k has better-than-chance predictive performance. Instead of a n x n predictor matrix, we horizontally stack all predictor matrices from $k = 1, \cdots, n$. The resulting matrix is given by

$$X_{all} = [X_1, \cdots, X_n]$$

and has dimensions n x n^2. This predictor matrix allows for construction of the *overall kNN GT*.

Properties

In this predictor matrix the individual matrices corresponding to a value of k can be considered variables. As a consequence, every value of k has some weight depending on the variance of X_k. Since we cannot compute the variance of X_k directly, we use the variance of the corresponding GT statistic as surrogate. No closed form solution exists to compute the variance of a ratio of quadratic forms. Instead we use a Taylor series approximation [6].

Conditional on a data set, the variance of S only depends on k through X_k. The variance of S is proportional to the eigenvalues of $(I - H)XX^T(I - H)$ and with that

proportional to the variance of the columns of X. We know that the variance is maximized when the number of 0's is (roughly) equal to the number of 1's. Hence we expect the GT to assign largest weights to values of k that are in the neighborhood of $\frac{1}{2}n$. Predictor matrices for values of k close to 1 and n receive smallest weights. Indeed, the weights for $k = 1$ and $k = n$ are always equal to zero as their test statistic is a constant. The overall kNN GT uses this vanilla weighting scheme.

This default weighting scheme might not be appropriate for every data set. The *uniform overall kNN GT* makes use of a uniform weighting scheme in which every k has the same weight. To achieve this, every element of X_{all} is multiplied by the reciprocal of the standard deviation of its corresponding test statistic $\frac{1}{\sqrt{\widehat{\text{Var}}(S_k)}}$. Given this uniform weighting, one can apply an alternative weighting scheme according to their believe. This amounts to multiplication of elements by $\frac{w_k}{\sqrt{\widehat{\text{Var}}(S_k)}}$ where w_k is the weight for elements from X_k. From a Bayesian point of view, weighting schemes could be interpreted as priors, directing power of the test against certain alternatives.

As n grows large, computation of the stacked GT statistic becomes increasingly more demanding, both memory- and time-wise. First of all, storing the stacked predictor matrix X_{all} requires increasingly more memory. Secondly, computation of (reweighted) inner product matrix $X_{\text{all}}X_{\text{all}}^T$ requires increasingly more time. The naive approach has a space and time complexity of $\mathcal{O}(n^3)$ and $\mathcal{O}(n^4)$, respectively. We propose an algorithm that reduces space complexity to $\mathcal{O}(n^2)$ and time complexity to $\mathcal{O}(n^3)$. The following lemma lies at the basis of this algorithm.

Lemma 1. *Given the horizontally stacked predictor matrix X_{all} that includes all possible values of k, $k = 1, \cdots, n$. Each value of k has some weight w_k. Collecting all weights in a matrix W allows for computation of the weighted predictor matrix $\overline{X}_{all} = W \odot X_{all}$. Let Z denote the inner product matrix of $\overline{X}_{all}$. An element z_{ij} from Z is equal to*

$$z_{ij} = \sum_{m=1}^{n} \sum_{k=n-min(|q_{im}|,|q_{jm}|)+1}^{n} w_k^2. \tag{3.4}$$

where q_{im} and q_{jm} denote the column indices of the nonzero elements in $x_{i\cdot}$ and $x_{\cdot j}$, respectively, for the mth neighbor.

Proof. Let w_h be the weight associated with the hth column of X_{all} and N be the set of indices of all columns of X_{all}, $N = \{1, 2, \cdots, n^2\}$. Then,

$$z_{ij} = \sum_{h=1}^{n^2} x_{ih}x_{hj}w_h^2. \tag{3.5}$$

Let p_m contain the indices over all n partitions of the mth neighbor such that

$$N = \bigcup_{m=1}^{n} p_m \text{ and } p_i \cap p_j \neq 0 \quad \forall i \neq j.$$

It follows that

$$z_{ij} = \sum_{h=1}^{n^2} x_{ih} x_{hj} w_h^2 = \sum_{m=1}^{n} \sum_{h \in p_m} x_{ih} x_{hj} w_h^2 \tag{3.6}$$

By definition an element x_{ij} is restricted to $x_{ij} = \{0, 1\}$. We know that $x_{ih} x_{hj} \neq 0$ if $x_{ih} = x_{hj} = 1$. Let q_{im} and q_{jm} denote the indices from p_m corresponding to nonzero elements in $x_{i\cdot}$ and $x_{\cdot j}$, respectively. Then we can write

$$z_{ij} = \sum_{m=1}^{n} \sum_{h \in (q_{im} \cap q_{jm})} x_{ih} x_{hj} w_h^2.$$

As solutions are nested

$$z_{ij} = \sum_{m=1}^{n} \sum_{k=n-|q_{im} \cap q_{jm}|+1}^{n} w_k^2 = \sum_{m=1}^{n} \sum_{k=n-min(|q_{im}|,|q_{jm}|)+1}^{n} w_k^2.$$

□

A shortcut

Our last variant of the stacked GT circumvents computation of this inner product matrix in its entirety. Column-wise rank orders are computed from the distance matrix D where elements are ranked from largest to smallest. This matrix X_{rank} lies at the basis of *ranked distance kNN GT*. The general idea is that this matrix contains roughly the same information as X_{all} since the proposed algorithm uses only this matrix for efficient computation of $X_{all} X_{all}^T$.

3.3 Continuous and categorical variables

In practice data sets do not exclusively consist of continuous variable. Often, one or more categorical variables are included as well. When this is the case, one has to carefully consider their representation in tandem with the distance measure as it fundamentally impacts kNN fitting and with that our novel tests.

The least complicated scenario is that the number of categories is restricted to two as there is no technical distinction between nominal and ordinal variables. When the categories are coded binary $\{0, 1\}$, the Hamming distance and Euclidian distance coincide. Both are equal to 1 when observations differ in their category and equal to 0 when they are the same. Note that this is not the case anymore when the categories are coded such that the difference between them is unequal to 1.

Representation of a binary categorical variables also affects relative weighting compared to continuous variables. Continuous variables are commonly standardized as to give each variable equal weighting. One can show that the expected squared Euclidian distance between two independent standard normal random variables, X and Y,

is equal to $\mathrm{E}[(X-Y)^2] = 2$. A Bernoulli distribution with success probability p is appropriate when these two random variables are binary categorical variables. When its categories are coded as $\{0, \delta\}$, the expected squared Euclidian distance is given by $\mathrm{E}[(X-Y)^2] = \delta^2(2p - 2p^2)$. Using the most common $\{0, 1\}$ representation ($\delta = 1$), the squared distance is maximized at $p = 0.5$ and corresponds to 0.5. In other words, in this formulation the weight of a categorical variables is at least 4 times smaller than a standardized continuous variable. Indeed, one can modulate weight by scaling a variable by a factor δ. A continuous variable with n observations can be discretized in n ways. Arguably, categorical variables should have smaller weight as they contain a fraction of the information of a continuous variable.

In case the number of categories is larger than two, the situation is more complicated. No closed form expression exist to derive their relative weighting instead we focus on the effect of different distance measures. When a variable has c categories, it is common to assign each category an integer, $0, 1 \cdots, c$. In this case the Hamming distance and Euclidian distance do not always give the same solution. The Hamming distance assigns the same value to every pair of differing categories. Hence in case of ordinal data some information is lost. The Euclidian distance preserves this information. However, one has to think about the scaling to accurately reflect the intercategory distances. This preservation of information is a drawback when dealing with nominal data as the Euclidian distance does not assign equal values to all pairs of differing categories. To solve this, one could dummy code $\{0, 1\}$ a nominal variable, effectively, putting each category on a regular simplex in N-dimensional space. This way the distance between any pair of categories is the same. Emphasising the complicated nature of mixture data in kNN, Boriah *et al.* demonstrate that there is no optimal distance measure for categorical data [7]. A measure has to be chosen taking the characteristics of the data into account. For simplicity, we decide to use the Hamming distance for all types of categorical variables.

3.4 Simulations

In the previous sections of this chapter we introduced novel tests for kNN and explored their theoretical characteristics. In this section we deepen our knowledge of their properties via a simulation study. This was conducted in order to obtain insights into performance against certain alternatives.

A performance study that exclusively includes our proposed methods would not provide the full picture due to a lack of reference frame. To provide context, we included existing methods for comparison. Our novel tests use the GT to test for nonzero coefficients of a linear model in the redeveloped kNN predictor space. Naturally, we introduced the *linear GT* which evaluates this in untransformed predictor space. Therefore, it is expected to perform well in linearly separable data. Furthermore, we introduced accuracy-tests based on ridge regression and kNN which are named *nested 5-fold cross-validation with ridge regression* and *nested 5-fold cross-validation with kNN*, respectively. As their names imply, these tests made use of a 5-fold nested cross-validation scheme in

combination with permutation testing with either kNN or ridge regression as learning algorithm (see section 2.3). We expected ridge regression to efficiently detect linearly separable alternatives that are not sparse with regard to the predictors. Moreover, ridge regression has similarities with the GT, both being shrinkage methods. The kNN based accuracy-test serves as a common approach to assess better than chance performance of kNN. Compared to ridge regression, kNN is a more flexible algorithm. Hence we expected it to perform well in linearly separable data with a larger signal-to-noise ratio (SNR) as well as in nonlinearly separable data.

Simulated data

In light of Shared Lunch, we expected some approaches to perform better in terms of power than others depending on the type of signal. Therefore, we evaluated power properties of these tests in two types of simulated data: linearly and nonlinearly separable data. To have a signal distinct from the linear approach, our nonlinear sampling method was chosen such that it could not be approximated well by linearity. Each of these sampling approaches was constructed such that the amount of signal and/or signal type could be modulated.

Linearly separable

To generate linearly separable data, we adhere to the idea of Fisher's linear discriminant analysis (LDA). We sample from two normal distributions that have mean vector μ_1 and μ_2 and share a variance-covariance matrix Σ. The Mahalanobis distance measures the distance between some distribution on $\mathbb{R}^d$ with mean vector m and variance-covariance matrix S and a point x. It is defined as

$$d_M = \|m - x\|_S = \sqrt{(m-x)^T S^{-1}(m-x)}.$$

Note that it is equal to the Euclidian distance when each dimension has unit-variance and the covariances are equal to zero.

This norm can be used as measure of the SNR in our LDA sampling approach. We can simplify our problem by taking one mean vector to be the zero vector and the variance-covariance matrix to be the identity matrix $\Sigma = I$. In doing so, we reduce the problem to choosing one mean/shift vector μ. Assuming balanced data with n observations, we correct for sample size by multiplying the squared Mahalanobis distance by $\frac{n}{2}$

$$SNR := \frac{n}{2}\|\mu\|_\Sigma^2 = \frac{n}{2}\mu^T\Sigma^{-1}\mu = \frac{n}{2}\mu^T\mu.$$

When we take $\mu = c\mathbf{1}$, we can write

$$c = \sqrt{\frac{2 \cdot SNR}{nd}}$$

Typical choices for the SNR are between 1 and 100.

Nonlinearly separable

Our nonlineary separable data follows a checkerboard pattern with square tiles. In other words, directly adjacent tiles have differing labels while indirectly adjacent tiles are from the same class. The board is square and resides in $\mathbb{R}^2$. Arguably, this one of the most complex patterns to separate in 2-dimensional space. The number of observations per tile n_{tile} and the length of the board l_{board} can be varied. By definition, the total number of observations is equal $n_{tile} \cdot l_{board}^2$ and data is always unbalanced as one class always has n_{tile} additional observations over the other class. Observations are uniformly sampled in their respective tile. For illustration purposes, a sample with $n_{tile} = 4$ and $l_{board} = 3$ has been plotted (**Figure 3.2**).

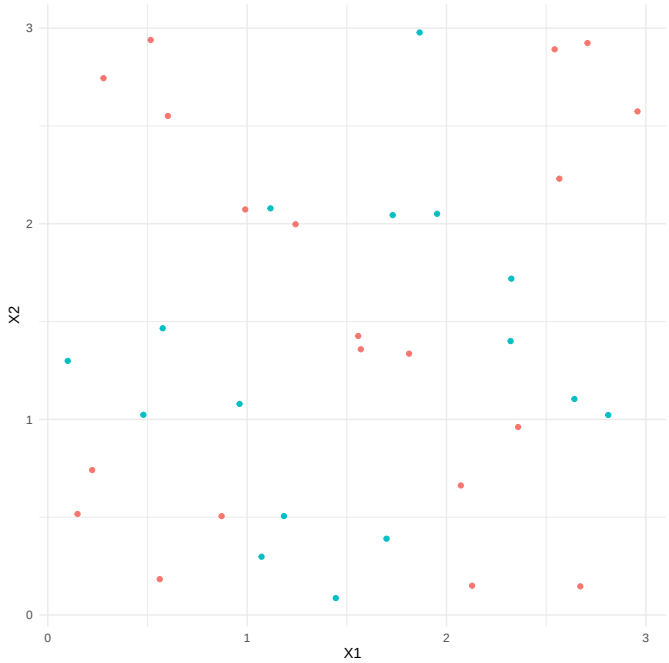

Figure 3.2: **A sample from our nonlinearly seperable data sampling approach.** A sample was taken with a board length of 3, $l_{board} = 3$, with 4 observations per tile, $n_{tile} = 4$. The two classes are represented by different colouring.

Performance

In order to gain insights into power properties of our novel tests, samples were drawn from the previously mentioned simulation approaches under varying parameters. In linearly separable data, power of the tests was analysed as function of the dimensionality, $p = 2, 10, 25, 50, 100$, and effect size, $SNR = 1, 2, \cdots, 5$. The former analysis had a fixed SNR of 5 and the latter analysis kept the dimensionality constant at a value of 2. Both experiments used a balanced sample of 40 observations. In case of nonlinearly separable data, we evaluated the effect of the number of observations per tile $n_{tile} = 2, 5, 10$ with

a constant board length $l_{board} = 3$. Furthermore, we investigated power properties for a fixed number of observation per tile $n_{tile} = 3$ and varying board length $l_{board} = 3, 5$. All code used for this and following chapters is stored in a repository. The link and instructions can be found in the Appendix.

Power, or rejected fraction, was defined as the proportion of replicates where the (empirical) p-value is smaller than 0.05. Per parameter we sampled 100 data sets. In case of the linear GT, proportion and p-value kNN GT we used the asymptotic solution available in the framework of the GT. For the overall and uniform overall kNN GT, nested 5-fold cross-validation with kNN and ridge p-values were computed by means of 1000 permutations under the null. In this scenario standard errors of the estimated rejected fraction under the null and in general are $\leq 2.2\%$ and $\leq 5.0\%$, respectively.

Power profiles of our novel ranked distance, overall and uniform overall kNN GT were competitive in our experiments on linearly separable data (**Figure 3.3A-B**). The linear GT performed best over the investigated parameter space. Despite being a linear predictor by definition, ridge regression appeared to make a compromise in terms of power compared to the former. Moreover, it was outperformed by the ranked distance kNN, overall and uniform overall kNN GT. We hypothesized that in our balanced LDA approach we allow for kNN to have better than chance performance for a large set of values for k. To be more precise, better than chance performance is most likely for values of k in the region of $\frac{1}{2}n$. This becomes less probable as k approaches 1 or n. This is in line with the weighting scheme of the overall GT, which assigns largest weights to values of k in the neighborhood of $\frac{1}{2}n$, explaining the subtle difference in power between the overall and uniform overall GT. The p-value kNN GT had the least desirable properties. Possible benefits of hyperparameter selection in the training set appeared to be overruled by the fact that it only uses half the data to compute a p-value.

In nonlinearly separable data, the p-value and uniform overall kNN GT were viable options when the number of observations per tile increased (**Figure 3.3C**). These tests and nested 5-fold cross-validation with kNN had a rejected fraction either close or equal to 1 for the largest sample size. Remaining tests did not have notable power over the studied parameters space. Having power for all evaluated n_{tile}, nested 5-fold cross-validation with kNN had the most desirable power properties. The uniform overall kNN GT only had substantial power for $n_{tile} = 10$. The p-value kNN took the middle ground being able to detect alternatives $n_{tile} = 5$. In contrast to the previous experiments, here the benefit of splitting data in half for hyperparameter selection outweighs the inherent loss of power. In our experiment on the board length none of our novel tests had considerable power (**Figure 3.3D**).

Taken together, these findings indicated that the ranked distance and overall kNN GT are optimal in terms of power when one is interested in detecting purely linear effects. Arguably, the former test is preferred due to a substantial advantage in terms of computational complexity compared to the latter. The p-value kNN GT appeared to perform best in nonlinear scenarios. Yet, it also had some power in linearly separable data. When one is equally interested in both types of alternatives, the uniform overall kNN GT appeared to be the optimal test. Power of this test was comparable to ranked distance and the overall kNN GT in linear cases while it retained considerable power to reject nonlinear alternatives.

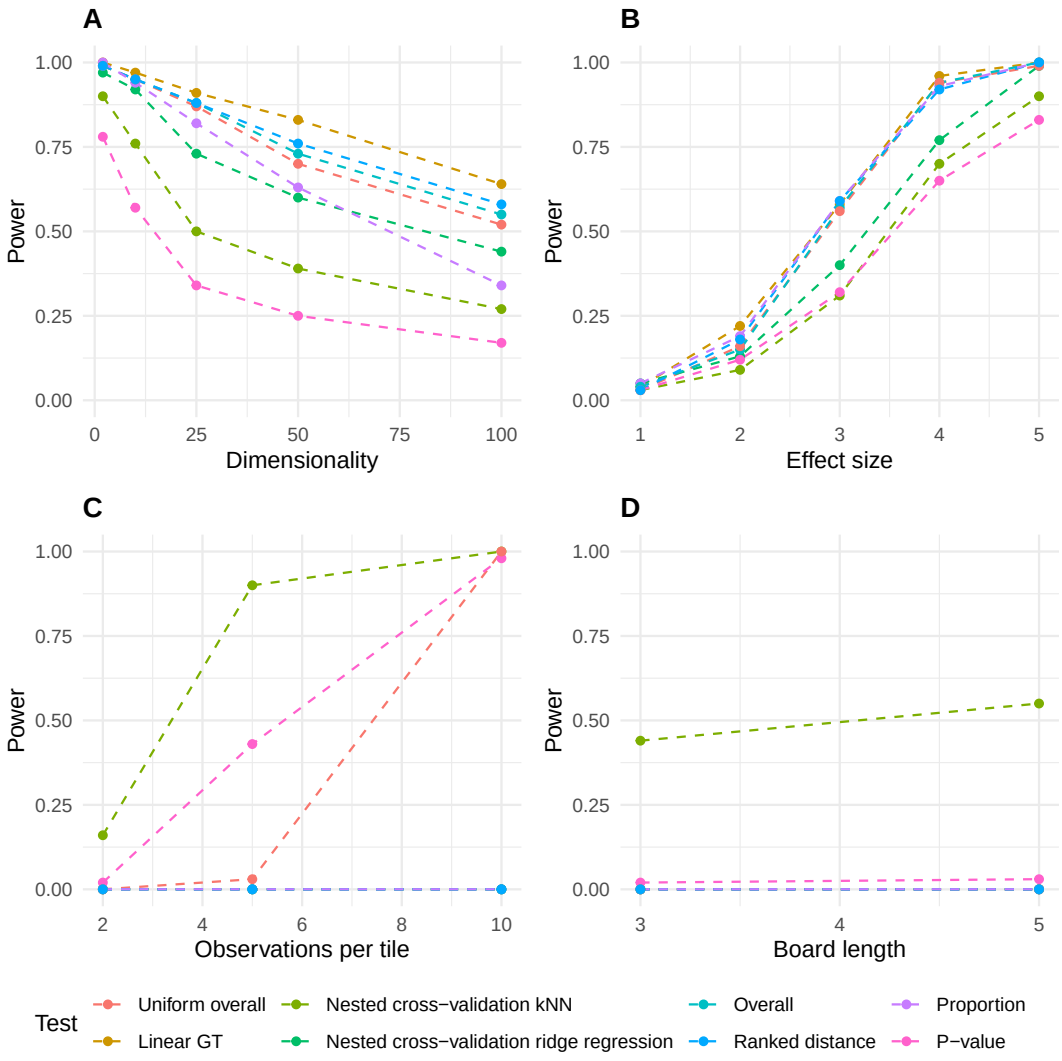

Figure 3.3: **Our novel k-nearest neighbors (kNN) tests have competitive power characteristics.** Power ($\alpha = 0.05$) for our novel kNN and preexisting tests has been assessed over 100 replications using our linearly and nonlinearly separable data sampling approaches. In linearly separable data, power was investigated as function of **(A)** the dimensionality with an effect size of 5 and **(B)** the effect size with a dimensionality of 2. In nonlinearly separable data, power was investigated as function of **(C)** the number of observations per tile with a constant board length of 3 and **(D)** the length of the board with 3 observations per tile.
Abbreviations: GT, global test; Uniform overall, Uniform overall kNN GT; Nested cross-validation kNN, Nested 5-fold cross-validation with kNN; Nested cross-validation ridge regression, Nested 5-fold cross-validation with ridge regression; Overall, Overall kNN GT; Ranked distance, Ranked distance kNN GT; Proportion, Proportion kNN GT; P-value, P-value kNN GT.

Chapter 4

Random forest

Random forest is like kNN a non-parametric algorithm, hence we have to introduce a transformation of predictor space that coincides with an empirical Bayesian linear model in order to construct GT based tests. Due to this apparent similarity, many parallels exists between our approaches for these two learning algorithms. Therefore, this chapter follows the same structure as the chapter on kNN. We introduce tests for random forest and explore their properties from a theoretical point of view as well as in a simulation study.

4.1 GT for specific hyperparameters

Similar to kNN, fitting random forest requires specification of hyperparameters and one is often interested in predictive performance for a choice of the parameters on a given data set. In this section we introduce GT based tests that assess better than chance performance of random forest. However, unlike for kNN, we cannot define the predictor matrix that is required for the GT directly from a fitted random forest. This is due to the supervised aspect of growing a forest. Hence, we require another approach.

The random forest predictor matrix consists of independently grown trees where each tree is constructed according to the hyperparameters and some given data set. Consider a training set of size n where for each observation i the outcome is binary $\{0, 1\}$ and the regressors reside in p-dimensional space $\mathbb{R}^p$. A tree is grown by recursively partitioning the predictor space, in a random fashion, until some tree-complexity criterion is met or when all leaves consist of one observations. Observations are dummy coded following a split. In practice this amounts to sampling a predictor, an observation and an inequality sign. Observations that satisfy this inequality get assigned a one whereas observations that do not a zero. Collecting the result in a vector yields a tree. The predictor matrix is comprised of B such trees where B is a natural number that has to be specified.

Example 4. *Consider the following training set of size* 4 *with predictors in* $\mathbb{R}$*:* $n_1 = [-1], n_2 = [0], n_3 = [1], n_4 = [2]$*. Suppose we partition our predictor space and dummy code according to this split. All possible trees we can grow can be represented in a matrix*

$$X = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Arguably, the first and last column do not represent a partition. Yet, they do not contribute to the GT statistic as their variance is equal to 0*.*

The general idea is that observations that are similar in terms of their outcome lie closer to each other compared to observations with a different outcome. Similar observations are expected to get assigned the same partition more often. This will be reflected in some trees. The actual ratio of trees containing signal to trees not containing any signal depends on the data and choice for the hyperparameters.

When we view each tree as a predictor and we have B of them, the model is a factorial ANOVA with B variables. Indeed, motivation, formulation and interpretation of random forest is identical to kNN except for the fact that kNN has n variables. The newly defined predictors for random forest are often interactions between multiple predictors in the original space, segmenting predictor space in hypercubes. In a similar manner to kNN, we can use this predictor matrix define our estimator (3.1) and, accordingly, formulate a linear model (3.2) with a corresponding hypothesis for any effect of random forest for a choice of the hyperparameters (3.3).

To make statements about the performance of random forest on a given data set, one has to make a choice for the hyperparameters: maximum number of nodes, maximum tree depth (tree depth), minimum number of observations per leaf (leaf size) and node (node size). We propose two tests that take care of this process. Analogous to the p-value kNN GT, we introduce the *p-value RF GT*. An equally sized training and test set are generated by randomly assigning observations to one of these sets. P-values are computed for forests that originate from different choices of the hyperparameters. The parameter combination corresponding to the smallest p-value is used in the test set. The *fixed RF GT* uses, as the name implies, a fixed combination of hyperparameters, grows a forest on the full data set and computes the corresponding p-value. We decided to centre both approaches around one hyperparameter, tree depth, and leave the remaining parameters unbounded.

Properties

Not all trees are weighted equally by the GT. The weight of a tree depends on the partition. Since a tree consists of elements that are by definition $\{0, 1\}$, we can interpret its variance as the variance of a binomial proportion. For instance, in example 4 the third tree has the largest weight, followed by the second and fourth. The tree with the largest weight has roughly as much 0's as 1's. This weighting might not always be

appropriate, for instance when data is unbalanced. In these scenarios one might prefer to apply a different weighting scheme. To do so, one multiplies with $\frac{w_i}{\sqrt{\widehat{\text{Var}}(x_i)}}$ where w_i and $\widehat{\text{Var}}(x_i)$ is the weight and variance, respectively, of the ith tree.

So far we have only considered trees with a maximum depth of 1. When the number of observations is small it is possible to write down all partitions. As we allow for more complex trees and/or increase the number of predictors the number of possible trees grows extremely quickly. Thus, in practice writing down all possible partitions is unfeasible and trees have to sampled.

The process of sampling trees introduces an additional source of variability, *auxiliary randomness*. Both the GT statistic for random forest and kNN tests have variability due to *inherent randomness* of the data. Yet, when conditioning on the data, the GT statistic for the kNN tests is a constant where the GT statistic for our novel random forest tests remains a random variable. To obtain consistent results, the variability of the random forest GT has to be as low as possible. Trees are grown independently and since the GT statistic is the sum of the test statistics over all individual predictors, the law of large numbers (LLN) applies. This implies that, conditional on the data, as the number of trees increases the distribution of the GT statistic converges to its expected value, at least in probability. Depending on the distributional shape of the individual GT statistics, it might take a larger number of trees to converge.

We suspect that when trees are grown deep the test will have suboptimal power. When this is the case, only few contain signal as the majority of trees will be more or less random. On average, a large number of trees have to be sampled to obtain a tree containing signal. The effect is effectively diluted by all trees and power will be low.

In a given data set, a binary tree representation does not exclusively correspond to one partition, and vice versa. It is often the case that different partitions yield identical trees. Furthermore, identical partitions can yield mirrored trees. In the latter scenario both trees detect the same signal. In this case one tree has a coefficient β and the other one a coefficient $-\beta$. In the current formulation of the GT we are interested in detecting the alternative $\sum_{i=1}^{p} \beta_i^2 \neq 0$. In other words, both trees are equally interesting when it comes to detecting alternatives.

Example 5. *Consider the following training set of size* 3 *with predictors in* $\mathbb{R}^2$*:* $n_1 = [-1, 1], n_2 = [0, 0], n_3 = [1, 1]$*. A selection of trees that can be build, given this data, were depicted (**Figure 4.1**).*

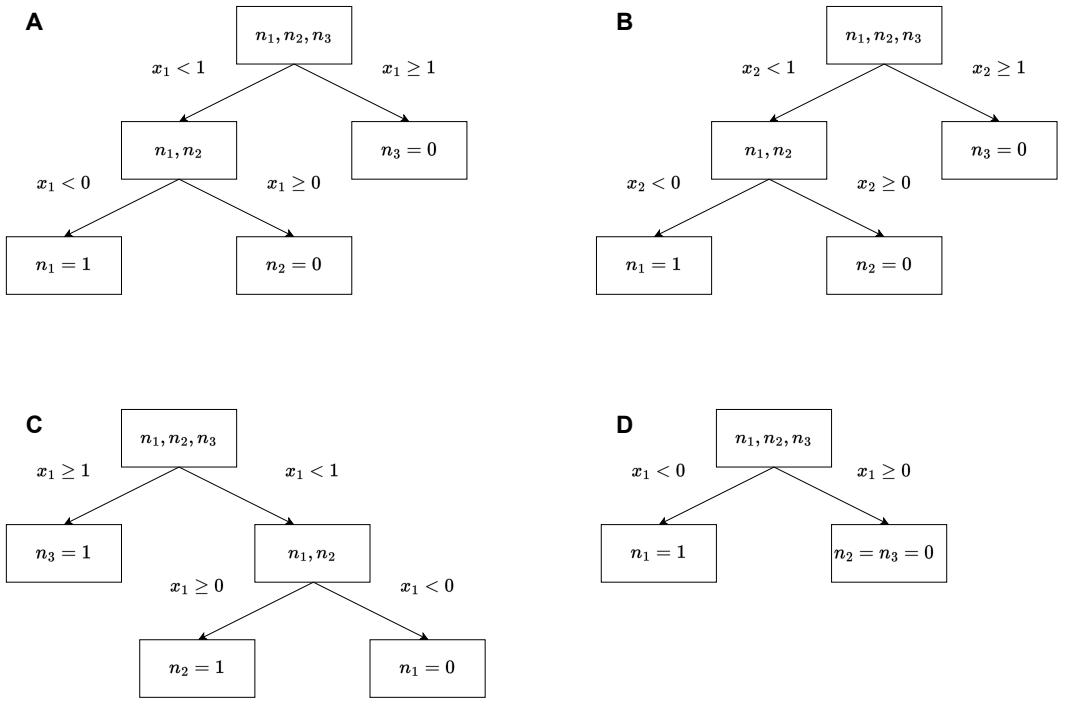

Figure 4.1: **A binary tree representation does not exclusively correspond to one partition, and vice versa.** Trees have been constructed based on the same data set to illustrate that **(A, B)** different partitions of the same tree depth may yield identical trees, **(C)** identical partitions can yield mirrored trees and **(D)** trees of differing depth may have identical trees.

Trees A, B and D are given by $[1, 0, 0]$ *and tree C by* $[0, 1, 1]$. *Tree A and B are identical in their layout except for the fact the former is split on* x_1 *and the latter is split on* x_2. *Tree D has a depth of* 1. *In contrast to A and B, which both have a depth of* 2. *Tree C is comprised of the same splits as tree A but it is mirrored. Tree A and C detect the same signal.*

4.2 Stacked sampling GT

Our previously introduced tests require specific choices for the hyperparameters. Ideally, we would like to a introduce a test that is similar to the stacked GT for kNN in the sense that is able to simultaneously make inference about all valid parameters combinations. This quickly becomes infeasible as the sample size grows. Instead, we propose a sampling approach in which parameters are sampled for each tree. As $B \to \infty$, the GT statistic for all parameters combinations converges to a single point. In our implementation of this sampling approach, the *sampling overall RF GT*, we only sampled the parameter

regarding maximum tree depth. The remaining hyperparameters were unbounded.

4.3 Simulations

Power of our novel random forest tests depend on the tree parameters as well as the number of trees. We have seen that convergence of their GT statistic depends on the number of trees used in its computation. As a consequence, when too few trees are constructed, there is large variability in the p-value which, in turn, leads to reduced power. Therefore, we investigated power as a function of the number of trees. Furthermore, since all our random forest approaches are centred around maximum tree depth, we explored power as a function of this parameter. Together these results allowed for efficient calibration of the random forest approaches. Power properties of these calibrated tests were studied to obtain insight into performance against certain alternatives.

Calibrating the number of trees

As previously discussed, the LLN applies to the GT statistic for random forest. Ideally, forests are grown extremely large to have a large probability to obtain an estimate close to its expected value. However, computational complexity increases as the number of trees grows. To make computation of our novel tests feasible, we aimed to identify the number of trees required to obtain a stable GT statistic estimate.

We suspected that distributional shapes of individual trees' test statistics may vary between data with different types of separation complexity. This in turn could severely affect the number of trees required for a stable estimate. To verify this and to make recommendations accordingly, we studied stability in linearly and nonlinearly separable data.

To investigate stability, we generated forests of predetermined sizes and computed corresponding p-values in 100 data sets. P-values were computed using 1000 permutations of the response. For a given data set, identical permutations were used between different forests to reduce noise. Power ($\alpha = 0.05$) over these data sets is a function of the number of trees. Once GT statistic estimates stabilizes, the p-value also stabilizes. Thus, we expect that after a certain number of trees, power more or less becomes a constant. We define this minimum value to be the number of trees required for a stable estimate.

The experiments relied on the same linear and nonlinear sampling approaches as introduced in chapter 2. Linearly separable data consisted of 20 observations with a dimensionality of 2 and effect size of 2.5. Data sampled from our nonlinear approach had $n_{tile} = 3$ and $l_{board} = 3$. Trees built for the linearly and nonlinearly separable data had a minimum leaf size of 3 and 2, respectively.

Results indicated a substantial difference in number of trees required to obtain a stable GT statistic estimate between our linear and nonlinear approach (**Figure 4.2A-B**). In case of linear separability, power seemed to be stable for forests consisting of more than 10000 trees. This number was markedly increased for nonlinearly separable

data. When using 25000 and 50000 trees power was equal to 0.81 and 0.88, respectively. We argue that the GT statistic stabilized for forest larger than 25000 trees. Equally important, these findings highlight considerable distributional differences between data with different types of separation complexity.

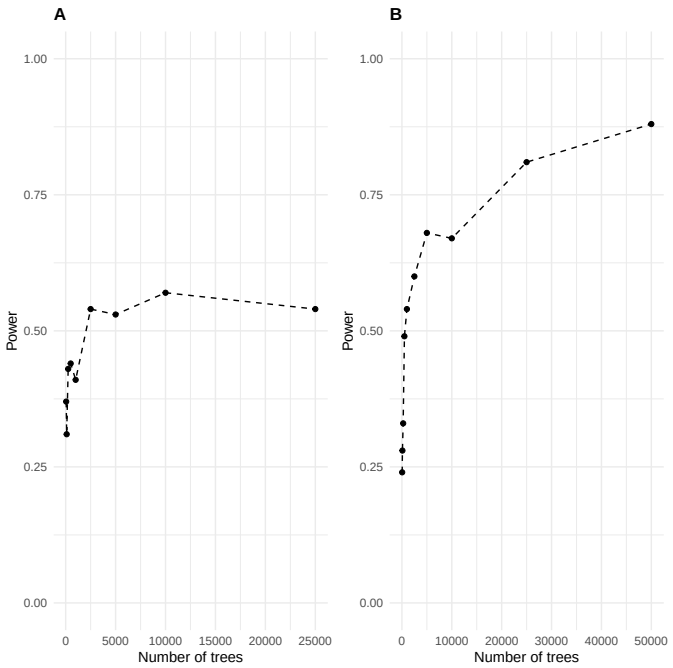

Figure 4.2: **The number of trees required for a stable global test statistic estimate depends on the data.** We investigated power ($\alpha = 0.05$) as function of the number of trees using our **(A)** linearly and **(B)** nonlinearly separable data sampling approaches. Trees for the former and latter approach had a minimum leaf size of 3 and 2, respectively. The remaining tree hyperparameters were unbounded. Linearly separable data consisted of 20 observations with an effect size of 2.5 and dimensionality of 2. Nonlinearly separable data had a board size of 3 with 3 observations per tile. P-values were computed using 1000 permutations. Power was computed over 100 replications.

Calibrating the maximum tree depth

We argued that a forest consisting of deeply grown trees is expected to have less signal compared to its shallow counterpart. Moreover, growing deep trees is computationally complex. Gaining insights into power properties of forest as function of the tree depth deepens our understanding but, equally important, provided the opportunity to calibrate our novel test random forest tests to be more efficient in terms of power and computation

The forests' power characteristics were investigated in both our linear and nonlinear approaches for a range of values for the maximum tree depth. The former sampled 20 observations with an effect size of 5 in 2 dimensions. We set the latter to $n_{tile} = 15$ and

$l_{board} = 3$. P-values were computed using forests consisting of 250 trees in tandem with 1000 permutations. Power was based on 100 replications.

Results indicated that exclusively trees with a depth of at most 10 display considerable power (**Figure 4.3A-B**). In case of our experiment on linearly separable data, extremely shallow trees had largest power. Trees grown deeper than a depth of 4 did not appear to have any power. When it comes to nonlinear separability, trees had to be grown deeper. The power curved displayed a sharp peak centred around a maximum depth of 4. Stumps or trees with maximum depth larger than 9 appeared to be unfavourable. These findings support our hypothesis that when forests are grown taller, trees containing signal do not outweigh trees that are more or less random in terms of weighted effect size.

Naturally, we implemented the p-value, fixed and sampling overall RF GT in such a way that power was directed most efficient. In case of the p-value RF GT this amounted to limiting the grid search for the forest's tree depth to a depth of 10. Trees were grown to a maximum depth of 3 for the fixed RF GT. For the sampling overall RF GT these findings implied uniform sampling of the maximum tree depth over a range of integer values from 1 to 5.

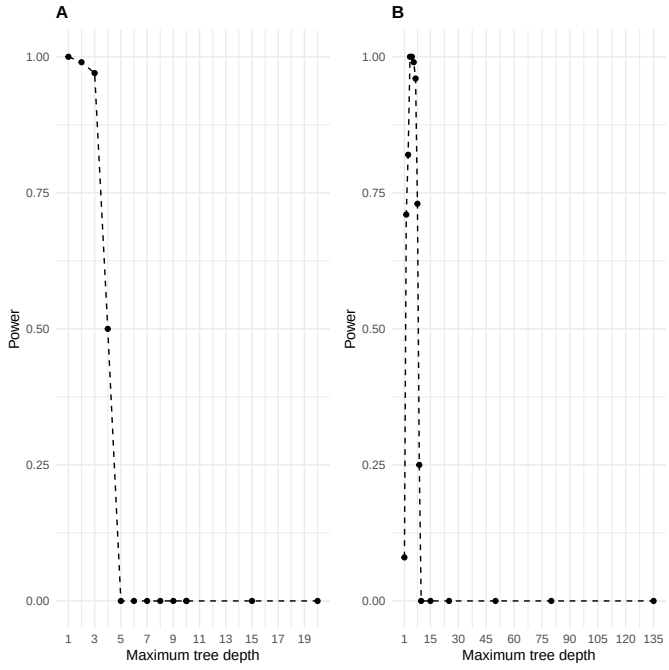

Figure 4.3: **Power profiles as a function of the tree depth differ depending on the data.** We investigated power ($\alpha = 0.05$) as a function of the tree depth using our **(A)** linearly and **(B)** nonlinearly separable data sampling approaches. The remaining tree hyperparameters were unbounded. Linearly separable data consisted of 20 observations with an effect size of 5 and dimensionality of 2. Nonlinearly separable data had a board size of 3 with 15 observations per tile. P-values were computed using 250 trees in combination with 1000 permutations. Power was computed over 100 replications.

Performance

Since we calibrated all hyperparameters of our novel tests for random forest, we decided to explore their power properties. To do so, we used a similar approach as described in the power experiments for kNN tests. This includes the linear and nonlinear sampling method and its parameter choices. P-values of our novel tests were computed using 1000 permutations under the null. In terms of tests, we adopted the linear GT and 5-fold cross-validation with ridge regression for performance comparison. In addition, we introduced *4-fold cross-validation with RF*. This is a commonly deployed accuracy-test for random forest with a fixed combination of hyperparameters. Due to the fixed hyperparameters, the test used an ordinary (4-fold) cross-validation scheme instead of a nested one. The remainder of the procedure is identical to nested cross-validation (see section 2.3). The leaf size and number of trees was fixed at 1 and 250, respectively. At each split $\lfloor \sqrt{p} \rfloor$ were sampled where p is the number of predictors. Due to the flexibility of random forest, we expected it to perform well in linearly separable data with a larger SNR as well as in nonlinearly separable data.

The sampling overall RF GT and fixed RF GT belonged to the best performing tests in experiments on linearly separable data (**Figure 4.4A-B**). The sampling overall RF GT and the fixed RF GT appeared to have comparable power properties, being on par with 5-fold cross-validation with ridge but having less power than the linear GT. The former seemingly performed best out of the two. In both simulations, the p-value RF GT had the least desirable power properties.

In nonlinearly separable data, 4-fold cross-validation with RF had vastly superior characteristics compared to our novel tests (**Figure 4.4C**). Arguably, this test was only outperformed by the p-value RF GT on the smallest sample size $n_{tile} = 2$. Note that the latter test did not demonstrate any power at $n_{tile} = 5$. For the largest sample size $n_{tile} = 10$, the sampling overall and the fixed RF GT displayed notable power, appearing to surpass power of the p-value RF GT. None of the tests had considerable power in the experiment on the board size (**Figure 4.4D**).

Thus, from our novel tests, the sampling overall and fixed RF GT had most desirable properties for detecting linear as well as nonlinear separability. Compared to these test, the p-value RF GT approached their power in scenarios of nonlinear separability while having markedly reduced power in linearly separable data, making it the preferred tests when one is only interested in detecting nonlinear alternatives.

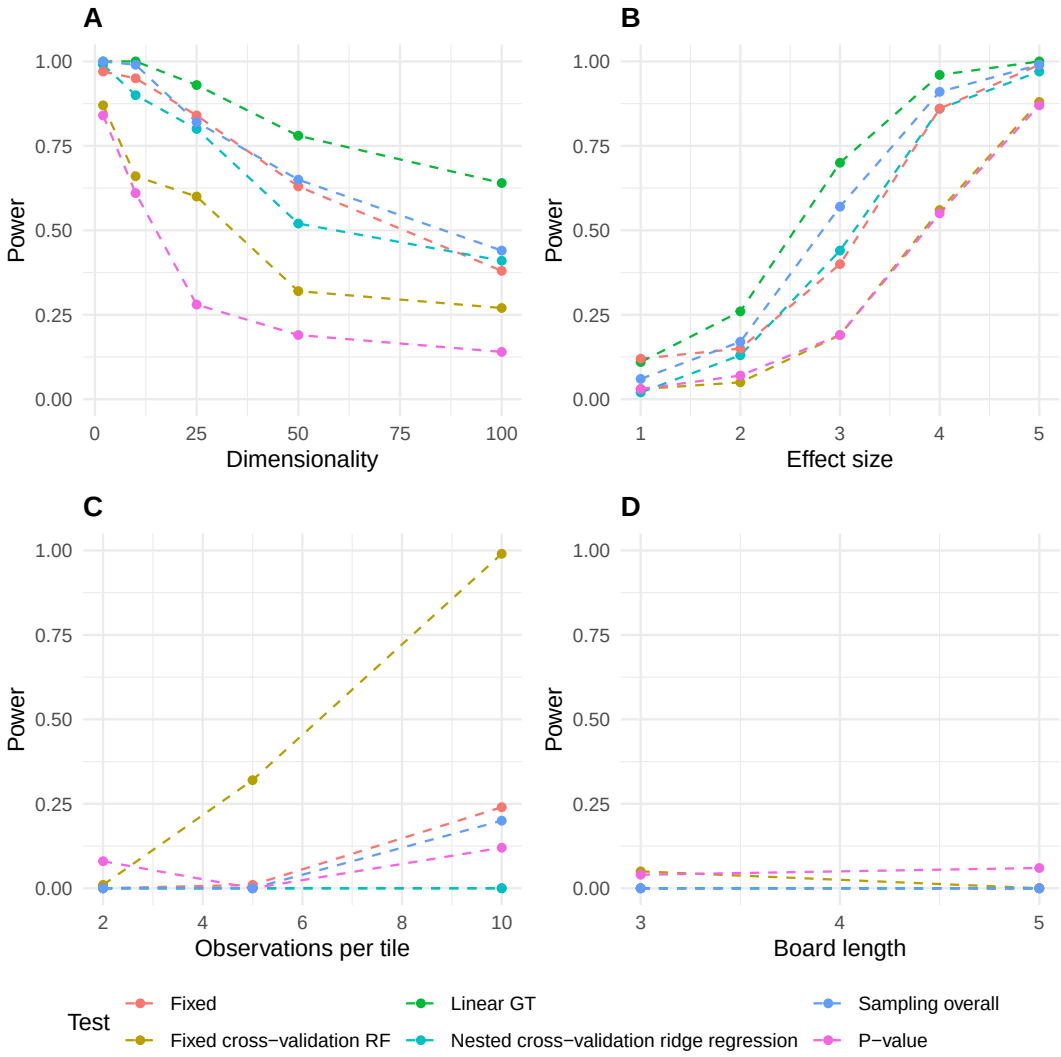

Figure 4.4: **Our novel random forest (RF) tests are viable options in terms of power.** Power ($\alpha = 0.05$) for our novel RF and preexisting tests has been assessed over 100 replications using our linearly and nonlinearly separable data sampling approaches. In linearly separable data, power was investigated as function of **(A)** the dimensionality with an effect size of 5 and **(B)** the effect size with a dimensionality of 2. In nonlinearly separable data, power was investigated as a function of **(C)** the number of observations per tile with a constant board length of 3 and **(D)** the length of the board with 3 observations per tile.
Abbreviations: GT, global test; Fixed, Fixed RF GT; Fixed cross-validation RF, Fixed 4-fold cross-validation with RF; Nested cross-validation ridge regression, Nested 5-fold cross-validation with ridge regression; Sampling overall, Sampling overall RF GT, P-value, P-value RF GT.

Chapter 5

Real data analysis

In the previous chapters we proposed novel methods for kNN and random forest for the detection of better than chance predictive performance and explored their power properties in simulated data. In this current chapter we demonstrate their applicability to real world data and highlight practicalities when doing so. Furthermore, we illustrate the concept of Shared Lunch.

The IMPACT II data as introduced by Steyerberg *et al.* formed the basis for our analysis [8]. This data set belongs to the IMPACT project which aims to advance clinical trial methodology for traumatic brain injury (TBI). IMPACT II is an agglomeration of fifteen studies, eleven randomized controlled trials and four observational studies, on patients with moderate to severe TBI as established by the Glasgow Coma Scale score. All studies recorded 6-month survival alongside ten baseline patient characteristics. Three continuous predictors: age, glucose and hemoglobin. Seven categorical variables: motor score, pupillary reactivity, Marshall Computerized Tomography classification, hypoxia, hypotension, traumatic subarachnoid hemorrhage and epidural hematoma. The first three variables are ordinal in nature with either three or four categories whereas the remaining four are dichotomous. Missing values were present in the data.

Steyerberg *et al.* performed imputation of the missing values in the original data for their analysis. The authors used multiple imputation in which the study also was included as fixed effect. This was done using the *mice* package in R. Subsequently, the authors used this data to illustrate their novel approaches for the detection of heterogeneity in predictor effects and predictions from models that were trained on different sources. The models aimed to predict 6-month mortality based on all introduced predictors.

For our analysis we computed p-values for all previously introduced tests using the full data set. We considered 6-month survival the outcome and did not include the glucose and hemoglobin predictors for practical reasons. The characteristics for both outcomes have been summarized (**Table 5.1**).

Table 5.1: **Summary of baseline characteristics of IMPACT-II by outcome.**
Abbreviations: IQR, interquartile range; CT, Computerized Tomography.

Variable	Alive (N = 8148)	Deceased (N = 2874)
Age (median [IQR])	29 [21, 42]	38 [24, 55]
Hypoxia (%)	1527 (18.7)	848 (29.5)
Hypotension (%)	1042 (12.8)	833 (29.0)
Traumatic subarachnoid hemorrhage (%)	3222 (39.5)	1790 (62.3)
Epidural hematoma (%)	1179 (14.5)	285 (9.9)
Motor score (%)		
1/2	1586 (19.5)	1264 (44.0)
3	1629 (20.0)	656 (22.8)
4	1945 (23.9)	493 (17.2)
5/6	2988 (36.7)	461 (16.0)
Pupillary reactivity (%)		
Both	6127 (75.2)	1198 (41.7)
None	1087 (13.3)	1209 (42.1)
One	934 (11.5)	467 (16.2)
Marshall CT classification (%)		
I/II	3852 (47.3)	627 (21.8)
III	1566 (19.2)	673 (23.4)
IV/V	2730 (33.5)	1574 (54.8)

When evaluating our novel kNN tests, we observed that distance ties can occur. In this scenario it is unclear how to define the predictor matrix for some values of k. In theory it can occur in any data set, yet it most frequently arises in data set lacking truly continuous variables. Namely, in the latter type of data sets it is possible for two or more observations to have identical predictor values resulting in these observations being equidistant to any observation. To illustrate the definition problem, consider the scenario when two observations are closest but equidistant to some observation. We cannot define the predictor matrix for $k = 2$. One solution would to be to add a small amount of noise to one predictor. This ensures that, in practice, distance ties cannot occur and predictor matrices can always be defined for some valid of k. In our current

analysis we added noise that was normally distributed with $\mu = 0$ and $\sigma = 10^{-4}$. Note that absence of at least one truly continuous variable also impacts random forest tests. As observation space cannot always be partitioned, a tree cannot always be grown to their maximum complexity according to the specified hyperparameters. This results in a more parsimonious tree than expected based on the parameters.

When using the full data set, p-values for all introduced tests turned out to be extremely small such that all tests were significant ($\alpha = 0.05$). Instead, we decided to sample subsets of observations and investigate power. Each sample consisted of 30 observations and was approximately representative with respect to the proportion of outcomes of the full data set. Power over 100 replications was computed.

In light of shared lunch, the results indicated nonlinear separability that borders on linearity (**Table 5.2**). From our novel kNN tests the proportion, overall, uniform overall and the ranked distance kNN GT had largest power. The p-value kNN GT performed substantially worse compared to the previous. This power profile corresponds to our experiments on linearly separable data. Results from our random forest tests also point towards this class of alternatives. Power of the fixed and sampling overall RF GT was roughly 3 times larger than the p-value RF GT. Yet, the linear GT had markedly increased power compared to all other tests in our experiments on linear separability. This suggested that the data is not truly linearly separable but it can serve as a reasonable approximation.

These findings regarding approximate linear separability are in line with literature. The full IMPACT II data set has been recently explored in the context of prognostic performance of learning algorithms [9, 10]. Both studies provided a comparison of classical logistic regression to flexible modern learning algorithms when modelling 6-month survival based on all predictors, including the glucose and hemoglobin predictors. The idea is that these modern algorithms should have better predictive performance once the outcome is nonlinearly related to the original predictors. These studies seemed to indicate that both classes of learning algorithms performed equally well. Thus the regressors appear to be roughly linearly related to the outcome.

Table 5.2: **IMPACT II appears to be linearly separable.** Power ($\alpha = 0.05$) for all introduced tests has been computed over 100 replications. Each replicate consisted of a sample of 30 observations from IMPACT II data in which the proportion of outcomes was approximately equal to the full data set.
Abbreviations: GT, global test; kNN, k-nearest neighbors; RF, random forest.

Test	Power
Linear GT	0.55
Nested 5-fold cross-validation with ridge regression	0.25
Nested 5-fold cross-validation with kNN	0.14
Proportion kNN GT	0.31
P-value kNN GT	0.14
Overall kNN GT	0.25
Uniform overall kNN GT	0.28
Ranked distance kNN GT	0.32
Fixed 4-fold cross-validation with RF	0.25
Fixed RF GT	0.54
P-value RF GT	0.18
Sampling overall RF GT	0.55

Chapter 6

Discussion

Current methods for assessing whether predictive performance of a classifier is better than chance are not optimal for every type of signal. In this current thesis, we provided novel approaches to detect better than chance performance of kNN and random forest that are in accordance with the signal detected. Redevelopment of these machine learning algorithms as empirical Bayesian models lied at these basis of these tests. Not only did this reformulation provide the framework to develop tests for specific (combinations of) hyperparameters but also for sets of hyperparameters. Simulation studies revealed that our novel tests had competitive power characteristics under linearly as well as nonlinearly separable alternatives compared to existing approaches. Moreover, we demonstrated their applicability to real word data sets.

Our novel tests distribute power differently compared to linear regression based tests. The explicit separating hyperplane from linear regression and the implicit hyperplane from kNN and random forest can be written as Taylor approximations. Linear regression is restricted to first order Taylor polynomials. Compared to the former, kNN and random forest gain flexibility by also including higher order terms. Where linear regression allocates all power to these first order terms, kNN and random forest have to distribute power over these as well as higher order terms. Often when one is interested in linearly separable alternatives, one is also interested in nonlinear alternatives that slightly deviate from linearity; alternatives that could be reasonably well approximated by linearity. In practice linear regression based tests are able to detect such patterns as long as a general predictive trend exists. More complex patterns that do not follow this trend can be picked up by kNN and random forest. A fitting example is our nonlinear sampling approach. Being supported by our results, a linear approach that yields reasonable separation does not appear to be possible. Thus, whether the power sacrifices made by our novel tests are worth it or not depends on ones interest in detecting certain alternatives.

Current simulation experiments were centred around balanced data. This is often not representative for real world scenarios. Based on theory, we hypothesize that some of our tests will have substantially hampered performance in these settings. The overall kNN GT implicitly assigns largest weights to values of k in the neighborhood of $\frac{1}{2}n$. While this weighting scheme appears to be appropriate in balanced data from a theoretical

point of view as well as in practice, we do not expect it to perform well in unbalanced data. We suggest a scheme where largest weights are shifted towards the number of observations for the least common outcome instead. As the values of k grows larger we expect a lack of signal as predictions will be increasingly more biased towards the most prevalent outcome. Similar weighting issues arise for the random forest based GT. These tests implicitly assign largest weights to trees that partition observations in roughly equally sized groups. By definition these trees cannot perfectly partition predictor space of unbalanced data. This is only accomplished by trees that partition observations in groups sizes from which the ratio is identical to that of the outcome. Thus in case of (highly) imbalanced data we argue that the weight as function of the variance has to resemble a bimodal function; weights are largest at the lower and upper bound of the domain. Future research has to evaluate the effect of imbalanced data on power of our novel test and asses performance of the suggested adaptations.

For kNN and random forest we proposed tests that evaluate performance for specific (combinations of) hyperparameters. In case of kNN, the value of k directly corresponds to a fitted algorithm. This is not the case for our random forest tests. Results indicated that power properties of tests based on random forest were optimal for relatively shallow trees. This observations is at odds with random forest. As previously discussed, random forest stems from the philosophy of ensembling decision trees that individually are unbiased. In other words, trees for random forest are grown to a large depth. Consequently, one can argue that our proposed tests are not valid for detecting better than chance performance of the random forest algorithm as they do not directly assess performance of the fitted algorithm. We believe that our test is valid. Our tests asses whether it is possible to construct decision trees for specific combination of hyperparameter that capture better than chance signal. When it is possible to construct such a tree then, by definition, it is possible to construct a random forest with this characteristic. In some sense a significant test for some combinations of hyperparameters marks the lower bound for tree complexity to have better than chance performance. Albeit redundant, trees can be grown deeper and retain this property. Thus, despite our random forest tests not being directly representative for a combination of hyperparameters, a significant test indirectly implies better than chance performance of less parsimonious forests.

Our proposed tests for both kNN and random forest have their own computational pros and cons. All tests for kNN require computation of the inner product matrix XX^T resulting in a time complexity of at least $\mathcal{O}(n^3)$. While this does not cause any problems for small sample sizes, computation quickly becomes infeasible as the number of observations grows. The tests for random forest do not suffer from increasing sample size. For a given tree-depth, time complexity is independent of the number of observations; time complexity is a constant. Here the main drawback is that the process of growing trees is computationally expensive. We have seen that it is preferable for random forests based tests to use a large number of trees as convergence of the GT statistic, and with that the p-value, to a singularity is generally slow. Thus from a computational point of view, kNN based tests are preferred over tests for random forest when the number of observations is relatively small and the other way round when the sample size is large.

Power of the fixed RF GT, and to a lesser degree the p-value and sampling overall RF GT, may be overestimated. Hyperparamaters of these tests were decided upon by means of the same sampling approaches as used for the power analyses. This can be seen as a (light) variant of "double-dipping". The parameters were calibrated based on their performance in terms of power. Naturally, we expect these tests to overperform in our power experiments. In light of the raised expectations by our current results, their characteristics may be underwhelming against other alternatives.

Computational power was a limiting factor throughout this current thesis. For all experiments the number of replications (per parameter) have been limited to 100. This number of replications is on the low side and an increase benefits stability of estimates. Furthermore, the stability experiment for nonlinearly separable data did not provide a truly decisive number of trees for obtaining stable estimates. Nonetheless, the results from the stability experiments for random forest indicated that for linearly and nonlinearly separable a minimum of 10000 and 25000, respectively, trees should be used for optimal power. Our implementation used forests consisting of 2500 trees. The novel tests are expected to have better performance than our results implied. A similar argument can be made for 4-fold cross-validation with RF. The lack of parameter selection most likely resulted in reduced power. A nested cross-validation scheme with RF is expected to have more power. Thus, upscaling these experiments would make results more robust.

To summarize, we introduced novel approaches that could effectively evaluate better than chance performance of the widely-used machine learning algorithms kNN and random forest. Equally important, the GT framework underlying these tests can be extended to other learning algorithms. Ultimately, approaches stemming from this philosophy further add to list of existing methods, each facilitating a Shared Lunch between learning algorithm and better than chance alternative.

Appendix

This repository contains all source code used for this current project. It is structured in such a way that the content of the chapters on kNN (chapter 3), Random forest (chapter 4) and Real data analysis (chapter 5) each have their separate rmd file. These rmd files require support R scripts to be executed. Running these rmd files yields all the results presented in this thesis. The required files have been listed per chapter. Additionally, we included the workspace images we obtained from the analyses. These are printed *italic*. These instructions can also be found in the repository itself.

kNN (chapter 3)

- KNNSimSynthData.Rmd
- corefuncs_knntests.R
- aux_funcs.R
- sampling_funcs.R
- *wsKNNSynthData.RData*

Random forest (chapter 4)

- RFSimSynthData.Rmd
- corefuncs_rftests.R
- aux_funcs.R
- sampling_funcs.R
- *wsRFSynthData.RData*

Real data analysis (chapter 5)

- RealDataAnalysis.Rmd
- corefuncs_knntests.R

- corefuncs_rftests.R
- aux_funcs.R
- *wsRealDataAnalysis.RData*

Bibliography

[1] Wolpert DH. The lack of a priori distinctions between learning algorithms. Neural computation. 1996;8(7):1341-90.

[2] Rosenblatt JD, Benjamini Y, Gilron R, Mukamel R, Goeman JJ. Better-than-chance classification for signal detection. Biostatistics. 2021;22(2):365-80.

[3] Goeman J, van de Geer S, van Houwelingen JH. Testing against a high dimensional alternative. Journal of the Royal Statistical Society Series B. 2006 06;68:477-93.

[4] Goeman J, van de Geer S, Kort F, van Houwelingen JH. A global test for groups of genes: Testing association with a clinical outcome. Bioinformatics (Oxford, England). 2004 02;20:93-9.

[5] Goeman J, van Houwelingen JH, Finos L. Testing against a high-dimensional alternative in the generalized linear model: Asymptotic type I error control. Biometrika. 2011 05;98:381-90.

[6] Paolella MS. Computing moments of ratios of quadratic forms in normal variables. Computational statistics & data analysis. 2003;42(3):313-31.

[7] Boriah S, Chandola V, Kumar V. Similarity measures for categorical data: A comparative evaluation. In: Proceedings of the 2008 SIAM international conference on data mining. SIAM; 2008. p. 243-54.

[8] Steyerberg EW, Nieboer D, Debray TP, van Houwelingen HC. Assessment of heterogeneity in an individual participant data meta-analysis of prediction models: an overview and illustration. Statistics in medicine. 2019;38(22):4290-309.

[9] van der Ploeg T, Nieboer D, Steyerberg EW. Modern modeling techniques had limited external validity in predicting mortality from traumatic brain injury. Journal of clinical epidemiology. 2016;78:83-9.

[10] Gravesteijn BY, Nieboer D, Ercole A, Lingsma HF, Nelson D, Van Calster B, et al. Machine learning algorithms performed no better than regression models for prognostication in traumatic brain injury. Journal of clinical epidemiology. 2020;122:95-107.